aws

使用者指南

AWS Secrets Manager

AWS Secrets Manager: 使用者指南

Table of Contents

什麼是 AWS Secrets Manager？

AWS Secrets Manager 協助您在整個生命週期中管理、擷取和輪換資料庫認證、應用程式憑證、OAuth 權杖、API 金鑰和其他機密。許多 AWS 服務會在秘密管理員中儲存和使用機密。

Secrets Manager 可協助您改善安全狀態，因為應用程式原始程式碼中不再需要硬式編碼憑證。將憑證儲存在 Secrets Manager 中有助於避免任何可以檢查您的應用程式或元件的人可能造成的危害。您可以將硬式編碼憑證取代為對 Secrets Manager 服務的執行期呼叫，以在需要時動態擷取憑證。

使用 Secrets Manager，您可以為機密設定自動輪換排程。這可讓您以短期秘密取代長期秘密，進而大幅降低洩漏風險。由於憑證不再與應用程式一起存放，因此輪換憑證不再需要更新您的應用程式並將變更部署至應用程式用戶端。

對於您的組織中可能具有的其他類型機密：

- AWS 憑證 — 我們建議您使用AWS Identity and Access Management。
- 加密金鑰 – 建議使用 AWS Key Management Service。
- SSH 金鑰 – 建議使用 Amazon EC2 Instance Connect。
- 私有金鑰和憑證 – 建議使用 AWS Certificate Manager。

Secrets Manager 入門

如果您是 Secrets Manager 新手，則請從 概念 或下列其中一個教學課程開始：

- the section called “替換已寫死機密 ”
- the section called “替換寫死資料庫憑證 ”
- the section called “交替使用者輪換”
- the section called “單一使用者輪換”

您可以使用機密執行的其他任務：

- 建立和管理秘密
- 控制對機密的存取
- 擷取秘密
- 輪換 秘密

- 監控秘密
- 稽核秘密以合規
- 建立秘密 AWS CloudFormation

符合標準

AWS Secrets Manager 已接受多項標準的稽核，當您需要取得合規性認證時，可成為您解決方案的一部分。如需詳細資訊，請參閱 合規驗證。

定價

當您使用 Secrets Manager 時，將按使用量付費，而沒有最低收費或設定費用。不會針對已標示為刪除的機密收取任何費用。如需目前完整定價清單，請參閱 AWS Secrets Manager 定價。

您可以使用 AWS 受管金鑰 `aws/secretsmanager` Secrets Manager 創建的免費加密密碼。如果您建立自己的 KMS 金鑰來加密密碼，請按照目前的 AWS 費 AWS KMS 率向您收費。如需詳細資訊，請參閱 AWS Key Management Service 定價。

當您開啟自動輪換 (受管循環除外) 時，Secrets Manager 會使用 AWS Lambda 函數來旋轉密碼，並以目前的 Lambda 速率向您收取旋轉函數的費用。如需詳細資訊，請參閱 AWS Lambda 定價。

如果您在帳戶 AWS CloudTrail 上啟用，您可以取得 Secrets Manager 所傳送之 API 呼叫的記錄。Secrets Manager 會將所有事件記錄為管理事件。 AWS CloudTrail 免費儲存所有管理事件的第一個副本。不過，對於用來儲存日誌的 Amazon S3 及 Amazon SNS (如果您啟用通知)，您可能需要付費。此外，如果您設定額外的線索，則管理事件的額外副本可能會產生成本。如需詳細資訊，請參閱 AWS CloudTrail 定價。

AWS 使用 AWS Secrets Manager 機密的服務

- AWS App Runner – 請參閱《AWS App Runner 開發人員指南》中的參考環境變數和管理環境變數。
- AWS 應用程式容器 — 請參閱《應用程式容器使用指南》中的管理 AWS App2Container 的AWS 密碼。
- AWS AppConfig – 請參閱《AWS AppConfig 使用者指南》中的建立自由格式組態設定檔。
- Amazon AppFlow — 見由其他服務管理的秘密.
- AWS AppSync – 請參閱《AWS AppSync 開發人員指南》中的教學課程：Aurora Serverless。

- Amazon Athena – 請參閱《Amazon Athena 使用者指南》中的使用 Amazon Athena 聯合查詢。
- Amazon Aurora-見由其他服務管理的秘密。
- AWS CodeBuild— 請參閱AWS CodeBuild 用戶指南中的 AWS Secrets Manager 示例 CodeBuild的私人註冊表。
- AWS DataSync – 請參閱由其他服務管理的秘密。
- Amazon DataZone — 請參閱 Amazon 用 DataZone 戶指南中的使用新 AWS Glue 連接為亞馬遜 Redshift 數據庫創建數據源。
- AWS Direct Connect – 請參閱由其他服務管理的秘密。
- AWS Directory Service— 請參閱將 Linux EC2 執行個體無縫加入您的 AWS 受管 Microsoft AD 目錄、將 Linux EC2 執行個體無縫加入 AD Connector 目錄，以及將 Linux EC2 執行個體無縫加入AWS Direct Connect 使用者指南中的 Simple AD 目錄。
- Amazon DocumentDB (with MongoDB compatibility) – 請參閱 the section called “建立資料庫秘密” 和《Amazon DocumentDB 開發人員指南》中的管理 Amazon DocumentDB 使用者。
- AWS Elastic Beanstalk – 請參閱《AWS Elastic Beanstalk 開發人員指南》中的 Docker 組態。
- Amazon 彈性容器登錄 — 請參閱 Amazon ECR 使用者指南中的建立直通快取規則。
- Amazon 彈性容器服務 – 請參閱《Amazon 彈性容器服務開發人員指南》中的教學課程：使用 Secrets Manager 秘密指定敏感資料、透過應用程式以程式設計方式擷取秘密、透過環境變數擷取秘密、擷取日誌記錄組態的秘密、教學課程：搭配 Amazon ECS 使用 FSx for Windows File Server 檔案系統、FSx for Windows File Server 磁碟區，以及任務的私有登錄身分驗證。
- Amazon 彈性容器服務服務 Connect — 見由其他服務管理的秘密。
- Amazon ElastiCache — 請參閱 Amazon 使用 ElastiCache 者指南中的自動輪替使用者密碼。
- AWS Elemental Live— 在 Elemental Live 使用者指南中，請參閱從執行階段交付 AWS Elemental Live 到運 MediaConnect 作的運作方式。
- AWS Elemental MediaConnect – 參閱《AWS Elemental MediaConnect 使用者指南》中的AWS Elemental MediaConnect中的靜態金鑰加密。
- AWS Elemental MediaConvert— 請參閱使用AWS Elemental MediaConvert 者指南中的使用 Kantar 以取得 AWS Elemental MediaConvert 輸出中的音訊浮水印。
- AWS Elemental MediaLive— 請參閱《MediaLive 使用指南》中的「設定 MediaLive 為受信任的實體」。
- AWS Elemental MediaPackage – 參閱《AWS Elemental MediaPackage 使用者指南》中的 Secrets Manager 存取 CDN 授權。
- AWS Elemental MediaTailor— 請參閱《AWS Elemental MediaTailor 用戶指南》中的配置 AWS Secrets Manager 訪問令牌身份驗證。

- 在 Amazon EC2 上執行 Amazon EMR – 請參閱《Amazon EMR 管理指南》中的儲存 Secrets Manager 中的敏感組態資料和將 Git 型儲存庫新增至 Amazon EMR。
- EMR Serverless – 請參閱《Amazon EMR Serverless 使用指南》中的透過 EMR Serverless 使用 Secrets Manager 進行資料保護。
- Amazon EventBridge — 見由其他服務管理的秘密.
- Amazon FSx – 請參閱《Amazon FSx for Windows File Server 使用者指南》中的檔案共享和將檔案共享組態遷移至 Amazon FSx。
- AWS Glue DataBrew – 請參閱由其他服務管理的秘密。
- AWS Glue Studio — 請參閱AWS Glue 開發人員指南中的教學課程：使用適用於彈性搜尋的 AWS Glue 連接器。
- AWS IoT SiteWise – 請參閱《AWS IoT SiteWise 使用者指南》中的設定資料來源身分驗證。
- Amazon Kendra – 請參閱《Amazon Kendra 使用者指南》中的使用資料庫的資料來源。
- Amazon Kinesis Video Streams – 請參閱《Amazon Kinesis Video Streams 開發人員指南》中的將 Amazon Kinesis Video Streams 邊緣代理程式部署至 AWS IoT Greengrass。
- AWS Launch Wizard— 請參閱《使用者指南》中 AWS Launch Wizard 的〈設定AWS Launch Wizard 使用中的目錄〉。
- Amazon Lookout for Metrics – 請參閱《Amazon Lookout for Metrics 開發人員指南》中的搭配 Lookout for Metrics 使用 Amazon RDS 和搭配 Lookout for Metrics 使用 Amazon Redshift。
- Amazon Managed Grafana – 請參閱《Amazon Managed Grafana 使用者指南》中的設定 Amazon Redshift。
- AWS Managed Services – 請參閱《AMS 進階使用者指南》中的 AWS Secrets Manager (AMS 自助式佈建)。
- Amazon Managed Streaming for Apache Kafka – 請參閱《Amazon Managed Streaming for Apache Kafka 開發人員指南》中的使用 AWS Secrets Manager進行使用者名稱與密碼身分驗證。
- 適用於 Apache 氣流的 Amazon 受管工作流程 — 請參閱 Amazon Apache 氣流使用者指南中 AWS Secrets Manager 的使用秘密管理員機密設定 Apache 氣流連線和針對 Apache 氣流變數使用秘密金鑰。
- AWS Marketplace – 請參閱由其他服務管理的秘密。
- AWS Migration Hub— 請參閱AWS Migration Hub 協調器使用者指南中的將 SAP NetWeaver 應用程式遷移到 Amazon EC2 上的應用程式 AWS和重新託管應用程式。
- AWS OpsWorks for Chef Automate – 請參閱由其他服務管理的秘密。
- AWS Panorama – 請參閱《AWS Panorama 開發人員指南》中的管理 AWS Panorama中的攝影機串流。

- AWS ParallelCluster – 請參閱《AWS ParallelCluster 使用者指南》中的整合 Active Directory。
- Amazon 問答 — 請參閱 Amazon Q 開發人員指南中的概念-身份驗證。
- Amazon QuickSight — 請參閱 Amazon 用 QuickSight 戶指南中的使用 AWS Secrets Manager 秘密代替 Amazon QuickSight 中的數據庫憑據。
- Amazon RDS – 請參閱 由其他服務管理的秘密。
- 亞馬遜 Redshift — 請參閱由其他服務管理的秘密the section called “建立資料庫秘密”、在中存放資料庫登入資料 AWS Secrets Manager、使用 Amazon Redshift 資料 API 和使用亞馬 Amazon Redshift 管理指南中的查詢編輯器查詢資料庫。
- Amazon Redshift 查詢編輯器 v2 – 請參閱 由其他服務管理的秘密。
- Amazon SageMaker — 請參閱 Amazon 開發人員指南中的關聯 Git 儲存庫與 Amazon SageMaker 筆記本執行個體、從資料庫匯入資料 (JDBC) 以及從雪花匯入資料。 SageMaker
- AWS Schema Conversion Tool— 請參閱《使用指南》 AWS Secrets Manager 中的使用 AWS SCT AWS Schema Conversion Tool 者介面中的使用。
- AWS Toolkit for JetBrains – 請參閱《AWS Toolkit for JetBrains 使用者指南》中的存取 Amazon Redshift 叢集。
- AWS Transfer Family – 請參閱《AWS Transfer Family 使用者指南》中的 AS2 連接器的基本身分驗證、使用自訂身分識別提供者，以及產生和管理 PGP 金鑰。
- AWS Wickr — 請參閱《W AWS ickr 管理指南》中的啟動資料保留機器人。

存取 AWS Secrets Manager

您可透過以下方式來使用 Secrets Manager：

- Secrets Manager 主控台
- 命令列工具
- AWS 開發套件
- HTTPS 查詢 API
- AWS Secrets Manager 端點

Secrets Manager 主控台

您可以使用瀏覽器型 Secrets Manager 主控台管理您的秘密，並使用主控台執行幾乎任何與您秘密相關的任務。

命令列工具

命 AWS 令列工具可讓您在系統命令列中發出指令，以執行 Secrets Manager 和其他工 AWS 作。與使用主控台相較，此方法更快速也更便利。如果您想要建置指令碼來執行工 AWS 作，指令列工具會很有用。

在命令 shell 中輸入命令時，存在命令歷史記錄被存取或公用程式存取命令參數的風險。請參閱the section called “降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險”。

命令列工具會自動使用 AWS 區域中服務的預設端點。您可以為 API 請求指定其他端點。請參閱the section called “Secrets Manager 端點”。

AWS 提供兩組指令行工具：

- AWS Command Line Interface (AWS CLI)
- AWS Tools for Windows PowerShell

AWS 開發套件

AWS SDK 包含各種程式設計語言和平台的程式庫和範例程式碼。開發套件的工作諸如以密碼演算法簽署請求、管理錯誤以及自動重試請求。若要下載並安裝任何 SDK，請參閱適用於 Amazon Web Services 的工具。

AWS SDK 會自動使用 AWS 區域中服務的預設端點。您可以為 API 請求指定其他端點。請參閱the section called “Secrets Manager 端點”。

有關 SDK 文件，請參閱：

- C++
- Go
- Java
- JavaScript
- 科特林
- .NET
- PHP
- Python（肉毒桿菌 3）
- Ruby
- Rust
- 阿巴
- 迅速

HTTPS 查詢 API

HTTPS 查詢 API 可讓您以程式設計方式存取機 Secrets Manager 和 AWS. HTTPS 查詢 API 可讓您直接將 HTTPS 請求發給該服務。

雖然您可以直接呼叫 Secrets Manager HTTPS 查詢 API，但建議您改為使用其中一個 SDK。SDK 可執行您原本必須手動執行的許多實用任務。例如，SDK 會自動簽署您的請求，並將回應轉換為語法上適合您語言的結構。

若要對 Secrets Manager 進行 HTTPS 呼叫，請連接至 ???。

AWS Secrets Manager 端點

若要以程式設計方式連接至 Secrets Manager，請使用端點，也就是服務的進入點 URL。Secrets Manager 端點是雙堆疊端點，這表示它們同時支援 IPv4 和 IPv6。

Secrets Manager 在部分區域提供支援聯邦資訊處理標準 (FIPS) 140-2 的端點。

Secrets Manager 支援 TLS 1.2 和 1.3。Secrets Manager 支援所有區域中的 PQTLS，唯中國地區除外。

> **Note**
>
> Python AWS SDK 和 AWS CLI 嘗試按順序調用 IPv6，然後 IPv4，所以如果你沒有啟用 IPv6，它可能需要一些時間之前通話超時，並重試 IPv4。若要解決這個問題，您可以完全停用 IPv6 或移轉至 IPv6。

以下是 Secrets Manager 的服務端點。請注意，命名與典型的雙堆疊命名慣例不同。

區域名稱	區域	端點	通訊協定
美國東部 (俄亥俄)	us-east-2	secretsmanager.us-east-2.amazonaws.com	HTTPS
		secretsmanager-fips.us-east-2.amazonaws.com	HTTPS
美國東部 (維吉尼亞北部)	us-east-1	secretsmanager.us-east-1.amazonaws.com	HTTPS
		secretsmanager-fips.us-east-1.amazonaws.com	HTTPS
美國西部 (加利佛尼亞北部)	us-west-1	secretsmanager.us-west-1.amazonaws.com	HTTPS
		secretsmanager-fips.us-west-1.amazonaws.com	HTTPS
美國西部 (奧勒岡)	us-west-2	secretsmanager.us-west-2.amazonaws.com	HTTPS
		secretsmanager-fips.us-west-2.amazonaws.com	HTTPS
非洲 (開普敦)	af-south-1	secretsmanager.af-south-1.amazonaws.com	HTTPS
亞太區域 (香港)	ap-east-1	secretsmanager.ap-east-1.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
亞太區域 (海德拉巴)	ap-south-2	secretsmanager.ap-south-2.amazonaws.com	HTTPS
亞太區域 (雅加達)	ap-southeast-3	secretsmanager.ap-southeast-3.amazonaws.com	HTTPS
亞太區域 (墨爾本)	ap-southeast-4	secretsmanager.ap-southeast-4.amazonaws.com	HTTPS
亞太區域 (孟買)	ap-south-1	secretsmanager.ap-south-1.amazonaws.com	HTTPS
亞太區域 (大阪)	ap-northeast-3	secretsmanager.ap-northeast-3.amazonaws.com	HTTPS
亞太區域 (首爾)	ap-northeast-2	secretsmanager.ap-northeast-2.amazonaws.com	HTTPS
亞太區域 (新加坡)	ap-southeast-1	secretsmanager.ap-southeast-1.amazonaws.com	HTTPS
亞太區域 (悉尼)	ap-southeast-2	secretsmanager.ap-southeast-2.amazonaws.com	HTTPS
亞太區域 (東京)	ap-northeast-1	secretsmanager.ap-northeast-1.amazonaws.com	HTTPS
加拿大 (中部)	ca-central-1	secretsmanager.ca-central-1.amazonaws.com secretsmanager-fips.ca-central-1.amazonaws.com	HTTPS HTTPS

區域名稱	區域	端點	通訊協定
加拿大西部 (卡加利)	ca-west-1	secretsmanager.ca-west-1.amazonaws.com	HTTPS
		secretsmanager-fips.ca-west-1.amazonaws.com	HTTPS
歐洲 (法蘭克福)	eu-central-1	secretsmanager.eu-central-1.amazonaws.com	HTTPS
歐洲 (愛爾蘭)	eu-west-1	secretsmanager.eu-west-1.amazonaws.com	HTTPS
歐洲 (倫敦)	eu-west-2	secretsmanager.eu-west-2.amazonaws.com	HTTPS
歐洲 (米蘭)	eu-south-1	secretsmanager.eu-south-1.amazonaws.com	HTTPS
歐洲 (巴黎)	eu-west-3	secretsmanager.eu-west-3.amazonaws.com	HTTPS
歐洲 (西班牙)	eu-south-2	secretsmanager.eu-south-2.amazonaws.com	HTTPS
歐洲 (斯德哥爾摩)	eu-north-1	secretsmanager.eu-north-1.amazonaws.com	HTTPS
歐洲 (蘇黎世)	eu-central-2	secretsmanager.eu-central-2.amazonaws.com	HTTPS
以色列 (特拉維夫)	il-central-1	secretsmanager.il-central-1.amazonaws.com	HTTPS
中東 (巴林)	me-south-1	secretsmanager.me-south-1.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
中東 (阿拉伯聯合大公國)	me-central-1	secretsmanager.me-central-1.amazonaws.com	HTTPS
南美洲 (聖保羅)	sa-east-1	secretsmanager.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (美國東部)	us-gov-east-1	secretsmanager.us-gov-east-1.amazonaws.com secretsmanager-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS
AWS GovCloud (美國西部)	us-gov-west-1	secretsmanager.us-gov-west-1.amazonaws.com secretsmanager-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS

AWS Secrets Manager 概念

下列概念對於瞭解 Secrets Manager 的運作方式相當重要。

- 秘密
- 版本
- 輪換
- 輪換策略

秘密

在 Secrets Manager 中，秘密由秘密資訊、秘密值，以及關於秘密的中繼資料組成。秘密值可以為字串或二進位。若要將多個字串值存放在秘密中，建議您搭配鍵/值對使用 JSON 文字字串，例如：

```
{
  "host"       : "ProdServer-01.databases.example.com",
  "port"       : "8888",
  "username"   : "administrator",
  "password"   : "EXAMPLE-PASSWORD",
  "dbname"     : "MyDatabase",
  "engine"     : "mysql"
}
```

秘密的中繼資料包括：

- Amazon Resource Name (ARN) 具有以下格式：

  ```
  arn:aws:secretsmanager:<Region>:<AccountId>:secret:SecretName-6RandomCharacters
  ```

  Secrets Manager 會在秘密名稱末尾包含六個隨機字元，協助確保秘密 ARN 是唯一。如果刪除原始秘密，然後使用相同的名稱建立新秘密，則這兩個秘密會因為這些字元而具有不同的 ARN。具有舊秘密存取權的使用者不會自動取得新秘密的存取權，因為 ARN 不同。
- 秘密的名稱、描述、資源政策和標籤。
- 加密金鑰的 ARN，這是 Secrets Manager 用來加密和解密秘密值的 AWS KMS key。Secrets Manager 一律會以加密形式存放秘密文字，並且一律加密傳輸中的秘密。請參閱the section called “秘密加密和解密”。

- 關於如何輪換秘密的資訊 (如果設定輪換)。請參閱the section called “輪換”。

Secrets Manager 會使用 IAM 許可政策，藉此確保唯有授權的使用者才能存取或修改秘密。請參閱AWS Secrets Manager 的身分驗證與存取控制。

秘密具有會保存加密秘密值副本的版本。在變更秘密值或輪換秘密後，Secrets Manager 會建立新的版本。請參閱the section called “版本”。

您可以透過複寫秘密在多個 AWS 區域 中使用秘密。在複寫秘密時，您會建立原始或主要秘密 (稱為複本秘密) 的副本。複本秘密會保持與主要秘密的連結。請參閱the section called “將機密複寫到其他區域”。

請參閱建立和管理秘密。

版本

秘密具有會保存加密秘密值副本的版本。在變更秘密值或輪換秘密後，Secrets Manager 會建立新的版本。

Secrets Manager 不會儲存帶有版本的秘密線性歷史記錄，而會為三個特定版本加上標記來進行追蹤：

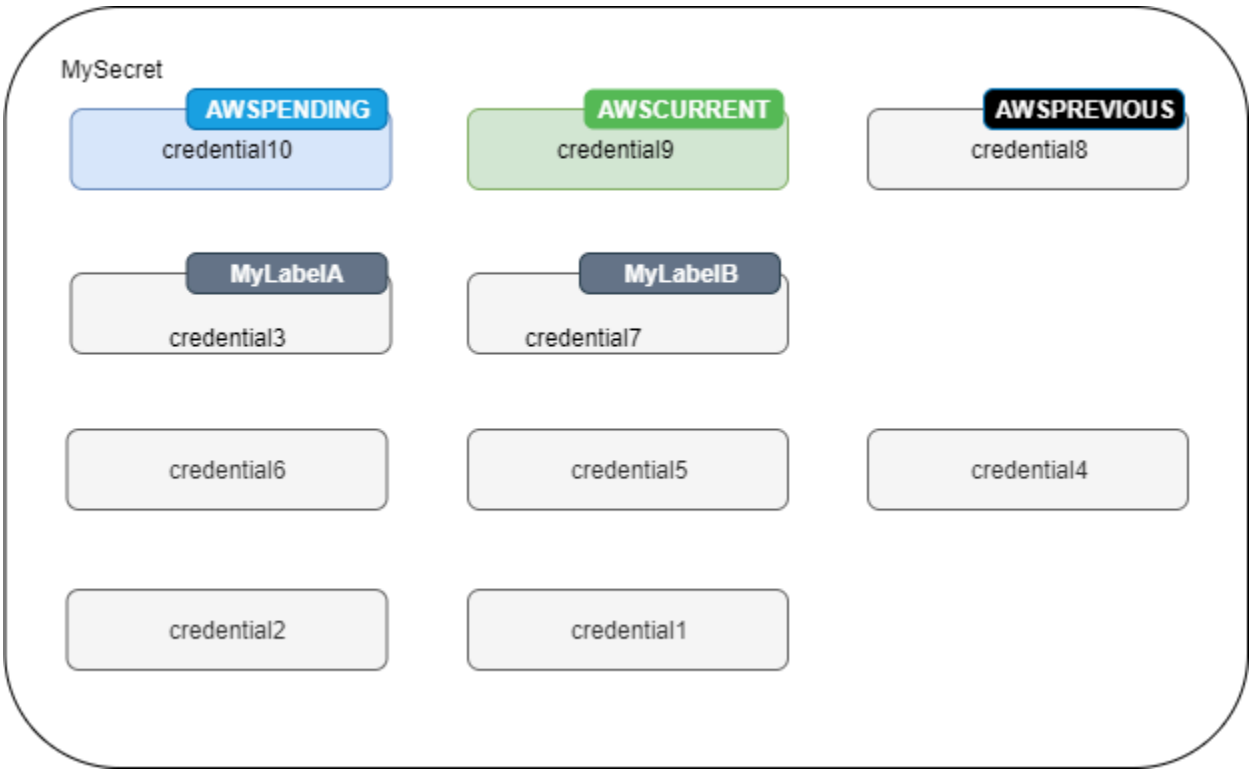
- 目前版本：AWSCURRENT
- 舊版本：AWSPREVIOUS
- 未激活版本 (輪換期間)：AWSPENDING

秘密始終有一個標記為 AWSCURRENT 的版本，當您擷取秘密值時，Secrets Manager 會依預設傳回該版本。

您也可以在 AWS CLI 中呼叫 update-secret-version-stage，從而用自己的標籤為版本加上標記。您最多可以將 20 個標籤連接至秘密中的版本。秘密的兩個版本不能擁有相同的預備標籤。一個版本可以有多個標籤。

Secrets Manager 永遠不會移除已標記版本，但會將未標記版本視為已遭取代。如果版本超過 100 個，Secrets Manager 會移除已棄用版本。Secrets Manager 不會刪除建立時間不到 24 小時的版本。

下圖顯示了帶有 AWS 標記版本和客戶標記版本的秘密。未標記版本將被視為已遭取代，並將在未來某個時間點被 Secrets Manager 移除。



輪換

輪換是定期更新秘密的程序，用意是讓攻擊者更難以存取憑證。在 Secrets Manager 中，您可以為秘密設定自動輪換。當 Secrets Manager 輪換秘密時，其會更新秘密和資料庫或服務中的憑證。請參閱輪換 秘密。

> **Tip**
>
> 對於部分 由其他服務管理的秘密，您可以使用受管輪換。若要使用 受管輪換，您可以先透過管理服務建立機密。

輪換策略

Secrets Manager 提供兩種輪換策略：

- 輪換策略：單一使用者
- 輪換策略：交替使用者

輪換策略：單一使用者

此策略會在一個秘密中更新一個使用者的憑證。對於 Amazon RDS Db2 執行個體，因為使用者無法變更自己的密碼，因此必須使用單獨的密碼來提供管理員登入資料。這是最簡單的輪換策略，適用於大多數使用案例。特別是，建議您將此策略用於一次性 (臨機操作) 或互動式使用者的憑證。

秘密輪換時，不會中斷開啟的資料庫連線。輪換正在進行時，資料庫中的密碼變更與更新秘密之間，有一小段時間落差。在此期間，資料庫有可能拒絕使用輪換後憑證的呼叫。您可以透過適當的重試策略降低這種風險。輪換之後，新的連線會使用新的憑證。

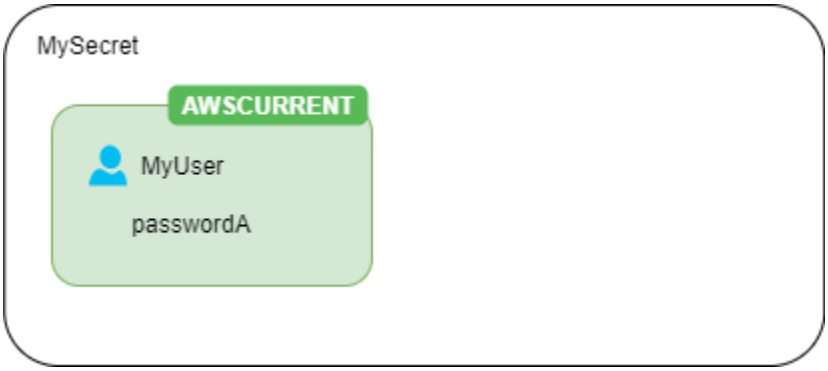
輪換策略：交替使用者

此策略會在一個秘密中更新兩個使用者的憑證。您可以建立第一個使用者，然後在第一次輪換期間，輪換函數會複製該使用者，以建立第二個使用者。每當秘密輪換時，輪換函數都會交替要更新的使用者密碼。由於大多數使用者沒有複製自身的許可，所以您必須提供其他秘密中 `superuser` 的憑證。在資料庫中複製的使用者沒有與原始使用者相同的許可時，建議使用單一使用者輪換策略，以及將其用於一次性 (臨機操作) 或互動式使用者的憑證。

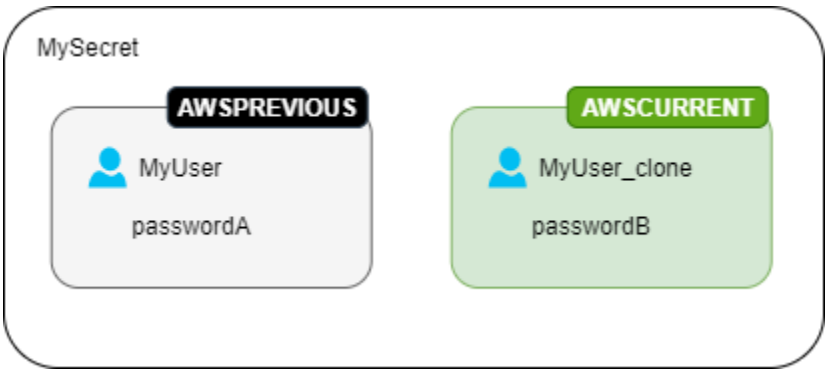
此策略適合具有許可模型的資料庫，其中一個角色擁有資料庫資料表，而第二個角色具有存取資料庫資料表的許可。這也適用於需要高可用性的應用程式。如果應用程式在輪換期間擷取秘密，應用程式仍會取得一組有效的憑證。輪換之後，`user` 和 `user_clone` 憑證都有效。在這種類型的輪換期間，應用程式遭到拒絕的機率比單一使用者輪換更低。如果資料庫託管於伺服器陣列，將密碼變更傳播到所有伺服器需要一段時間，則資料庫有可能拒絕使用新憑證的呼叫。您可以透過適當的重試策略降低這種風險。

Secrets Manager 會建立複製使用者，該使用者擁有與原始使用者相同的許可。如果在建立複製使用者後變更原始使用者的許可，您也必須變更複製使用者的許可。

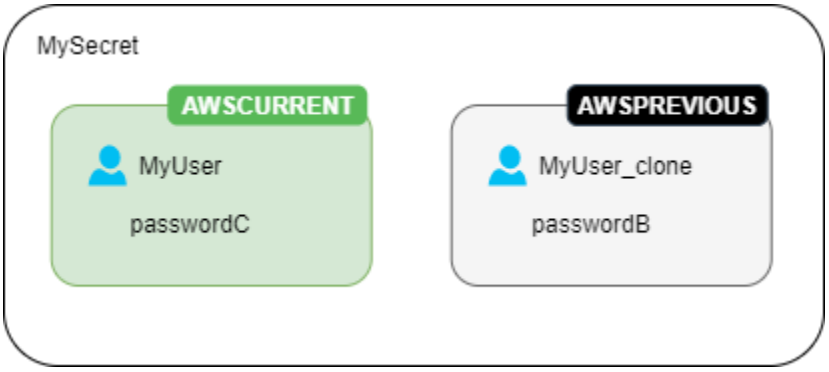
例如，如果您使用資料庫使用者的憑證建立秘密，則該秘密會包含一個具有這些憑證的版本。



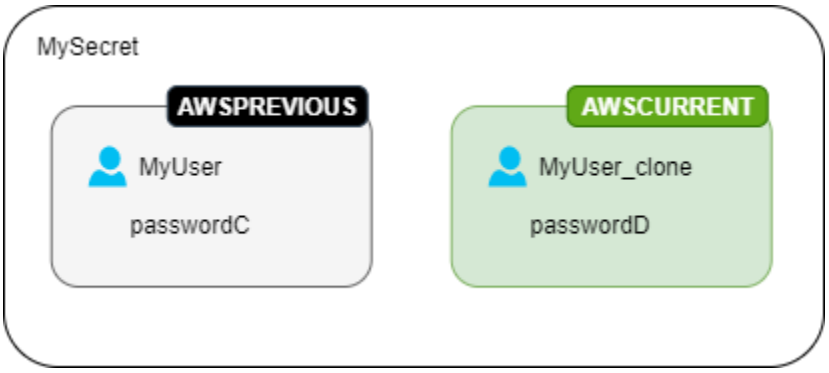
第一次輪換：輪換函數會使用產生的密碼建立用戶的複製，並且這些憑證會成為目前的秘密版本。



第二次輪換：輪換函數會更新原始使用者的密碼。



第三次輪換：輪換函數會更新複製使用者的密碼。



AWS Secrets Manager 教學課程

主題

透過 Amazon CodeGuru Reviewer 在您的程式碼中尋找未受保護的機密

Amazon CodeGuru Reviewer 是一項服務，可使用程式分析和機器學習來偵測開發人員難以找到的潛在缺陷，並提供改善 Java 和 Python 程式碼的建議。CodeGuru Reviewer 與機密管理員整合，在您的程式碼中找到未受保護的機密。如需該服務可以找到的機密類型，請參閱《Amazon CodeGuru Reviewer 使用者指南》中的 CodeGuru Reviewer 偵測的機密類型。

一旦您找到已寫死機密，請採取措施來替換它們：

- the section called “替換寫死資料庫憑證 ”
- the section called “替換已寫死機密 ”

將已寫死機密移動到 AWS Secrets Manager

如果您的程式碼中有純文字機密，我們建議您輪換它們並將它們儲存在機密管理員中。將機密移動到機密管理員可解決機密顯示問題，讓任何看到程式碼的人無法看見，因為再下一步，您的程式碼就能直接從機密管理員擷取機密。輪換機密將撤消目前已寫死機密，使其不再有效。

如需資料庫憑證機密，請參閱 將硬式編碼的資料庫認證移至 AWS Secrets Manager。

在開始之前，您需要決定需要存取該機密的人。我們建議使用兩個 IAM 角色來管理您的機密權限：

- 管理組織中機密的角色。如需相關資訊，請參閱 the section called “Secrets Manager 管理員許可”。您將使用此角色建立並輪換機密。

- 可在執行階段使用機密的角色，例如在本教學中使用 *RoleToRetrieveSecretAtRuntime*。您的程式碼擔任此角色以擷取機密。在本教學中，您僅授予該角色擷取一個機密值的權限，並透過使用機密的資源策略授予權限。如需其他替代方案，請參閱 the section called “後續步驟”。

步驟：

步驟 1：建立機密

第一步是將現有的已寫死機密複製到機密管理員中。如果機密與 AWS 資源相關，請將其儲存在與資源相同的區域。否則，請將其儲存在您使用案例中延遲最低的區域中。

若要建立機密 (控制台)

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：
   a. 針對 Secret type (機密類型)，選擇 Other type of secret (其他機密類型)。
   b. 將您的機密輸入為 Key/value pairs (金鑰/值對) 或 Plaintext (純文字)。一些範例：

API 金鑰/值對：

ClientID：*my_client_id*

ClientSecret ：*wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

憑證金鑰/值對：

Username：*saanvis*

Password：*EXAMPLE-PASSWORD*

API 金鑰/值對：

ClientID：*my_client_id*

ClientSecret ：*wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

OAuth 字符純文字：

AKIAI44QH8DHBEXAMPLE

數位憑證純文字：

```
-----BEGIN CERTIFICATE-----
EXAMPLE
-----END CERTIFICATE-----
```

私有金鑰純文字:

```
----BEGIN PRIVATE KEY ---
EXAMPLE
---- END PRIVATE KEY ----
```

c. 如需 Encryption key (加密金鑰)，請選擇 aws/secretsmanager 使用用於機密管理員的 AWS 受管金鑰。使用此金鑰無需任何成本。您也可以使用自己的顧客託管金鑰，例如從另一個 AWS 帳戶 存取機密。如需有關使用客戶受管金鑰的成本的資訊，請參閱 定價。

d. 選擇 Next (下一步)。

4. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：

a. 輸入描述性的 Secret name (機密名稱) 和 Description (描述)。

b. 在 Resource permissions (資源許可) 中，選擇 Edit permissions (編輯許可)。貼上以下政策，此政策允許 *RoleToRetrieveSecretAtRuntime* 擷取機密，然後選擇 Save (儲存)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:role/RoleToRetrieveSecretAtRuntime"
      },
```

```
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

c. 請選擇頁面最下方的 Next (下一頁)。

5. 在 Configure rotation (設定輪換) 頁面上，保持輪換關閉。選擇 Next (下一步)。
6. 在 Review (檢閱) 頁面上，檢閱機密詳細資訊，然後選擇 Store (存放)。

步驟 2：更新您的程式碼

您的程式碼必須擔任 IAM 角色 *RoleToRetrieveSecretAtRuntime* 以便能夠擷取到機密。如需相關資訊，請參閱《切換到 IAM 角色 (AWS API)》。

接下來，您可以使用機密管理員提供的範本程式碼更新您的程式碼，以便從機密管理員中擷取機密。

若要尋找範本程式碼

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 在 Secrets (機密) 頁面中，選擇機密。
3. 向下捲動至 Sample code (範本程式碼)。選擇您的程式設計語言，然後複製程式碼片段。

在您的應用程式中，刪除已寫死機密並貼上程式碼片段。根據您的程式碼語言，您可能需要向程式碼片段中的函數或方法新增調用。

測試您的應用程式是否按預期運作，並使用機密代替已寫死機密。

步驟 3：更新機密

最後一個步驟是撤消並更新已寫死機密。請參閱機密的來源以尋找撤消和更新機密的指示。例如，您可能需要停用目前機密並產生新機密。

使用新值更新機密

1. 開啟位於的機密管理員控制台 https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Secrets (機密)，然後選擇該機密。
3. 在 Secret details (機密詳細資訊) 頁面，向下捲動並選擇 Retrieve secret value (擷取機密值)，然後選擇 Edit (編輯)。

4. 更新機密，然後選擇 Save (儲存)。

接下來，測試您的應用程式在新機密下是否能按預期運作。

後續步驟

從程式碼中刪除已寫死機密後，接下來需要思考一些構想：

- 若要在 Java 和 Python 應用程式中尋找已寫死機密，我們推薦《Amazon CodeGuru Reviewer》。
- 您可以透過快取機密來提高效能並降低成本。如需更多詳細資訊，請參閱 擷取秘密。
- 如須從多個區域存取的機密，請考慮複製您的機密以改善延遲。如需更多詳細資訊，請參閱 the section called “將機密複寫到其他區域”。
- 在此教學中，您授予 *RoleToRetrieveSecretAtRuntime* 僅擷取機密值的權限。若要向角色授予更多許可 (例如獲取有關機密的中繼資料或檢視機密清單)，請參閱 the section called “許可政策範例”。
- 在此教學中，您透過使用機密的資源策略授予 *RoleToRetrieveSecretAtRuntime* 權限。如需授予權限的其他方式，請參閱 the section called “將許可政策連接至身分”。

將硬式編碼的資料庫認證移至 AWS Secrets Manager

如果您的程式碼中有純文字資料庫憑證，我們建議您將憑證移動到機密管理員，然後立即輪換它們。將憑證移動到機密管理員可解決憑證顯示問題，讓任何看到程式碼的人無法看見，因為再下一步，您的程式碼就能直接從機密管理員擷取憑證。輪換機密會更新密碼，然後撤銷目前的已寫死密碼，使其不再有效。

如需 Amazon RDS、Amazon Redshift 和 Amazon DocumentDB 資料庫，請使用此頁面中的步驟將寫死憑證移動到機密管理員。如需其他類型的憑證和其他機密，請參閱 the section called “替換已寫死機密 ”。

在開始之前，您需要決定需要存取該機密的人。我們建議使用兩個 IAM 角色來管理您的機密權限：

- 管理組織中機密的角色。如需相關資訊，請參閱 the section called “Secrets Manager 管理員許可”。您將使用此角色建立並輪換機密。
- *RoleToRetrieveSecretAtRuntime*在本教學課程中，可以在執行時間使用認證的角色。您的程式碼擔任此角色以擷取機密。

步驟：

步驟 1：建立機密

第一步是將現有的寫死憑證複製到機密管理員中的機密中。如需實現最低延遲，請將機密儲存在與資料庫相同的區域中。

若要建立機密

1. 請開啟位於 https://console.aws.amazon.com/secretsmanager/ 的機密管理員控制台。
2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：
   a. 針對 Secret type (機密類型)，選擇要存放的資料庫憑證的類型：
      - Amazon RDS 資料庫
      - Amazon DocumentDB 資料庫
      - Amazon Redshift 數據倉庫。
      - 如需其他類型的機密，請參閱《替換已寫死機密》。
   b. 如需憑證，請輸入資料庫的現有寫死憑證。
   c. 如需 Encryption key (加密金鑰)，請選擇 aws/secretsmanager 使用用於機密管理員的 AWS 受管金鑰 。使用此金鑰無需任何成本。您也可以使用自己的顧客託管金鑰，例如從另一個 AWS 帳戶存取機密。如需有關使用客戶受管金鑰的成本的資訊，請參閱 定價。
   d. 如需資料庫，請選擇您的資料庫。
   e. 選擇 Next (下一步)。
4. 在 Configure secret (設定秘密) 頁面上，執行下列動作：
   a. 輸入描述性的 Secret name (機密名稱) 和 Description (描述)。
   b. 在 Resource permissions (資源許可) 中，選擇 Edit permissions (編輯許可)。貼上下列可擷取密碼*RoleToRetrieveSecretAtRuntime*的原則，然後選擇 [儲存]。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:role/RoleToRetrieveSecretAtRuntime"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

c. 請選擇頁面最下方的 Next (下一頁)。

5. 在 Configure rotation (設定輪換) 頁面上，暫時將輪換關閉。您稍後再打開。選擇 Next (下一步)。
6. 在 Review (檢閱) 頁面上，檢閱機密詳細資訊，然後選擇 Store (存放)。

步驟 2：更新您的程式碼

您的程式碼必須擔任 IAM 角色 *RoleToRetrieveSecretAtRuntime*，才能擷取密碼。如需詳細資訊，請參閱切換至 IAM 角色 (AWS API)。

接下來，您可以使用機密管理員提供的範本程式碼更新您的程式碼，以便從機密管理員中擷取機密。

若要尋找範本程式碼

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 在 Secrets (機密) 頁面中，選擇機密。
3. 向下捲動至 Sample code (範本程式碼)。選擇您的語言，然後複製程式碼片段。

在您的應用程式中，刪除寫死憑證並貼上程式碼片段。根據您的程式碼語言，您可能需要向程式碼片段中的函數或方法新增調用。

測試您的應用程式是否按預期運作，並使用機密代替寫死憑證。

步驟 3：輪換機密

最後一步是透過輪換機密撤消寫死憑證。輪換是定期更新機密的過程。當您輪換機密時，會更新機密和資料庫中的憑證。機密管理員可以在您設定的時程上自動為您輪換機密。

設定輪換的一部分是確保 Lambda 輪換函數可以存取機密管理員和您的資料庫。在您開啟自動輪換後，機密管理員將在與資料庫相同的 VPC 中建立 Lambda 輪換函數，好讓它有網路可以存取資料庫。Lambda 輪換函數還必須能夠調用機密管理員來更新機密。我們建議您在 VPC 中建立 Secrets Manager 端點，以便從 Lambda 呼叫至機 Secrets Manager 時不會離開 AWS 基礎結構。如需說明，請參閱VPC 端點。

若要打開輪換功能

1. 請開啟位於 https://console.aws.amazon.com/secretsmanager/ 的機密管理員控制台。
2. 在 Secrets (機密) 頁面中，選擇機密。
3. 在 Secret details (機密詳細資訊) 頁面的 Rotation configuration (輪換組態) 區段中，選擇 Edit rotation (編輯輪換)。
4. 在 Edit rotation configuration (編輯輪換組態) 對話方塊中，執行以下動作：
   a. 開啟 Automatic rotation (自動輪換)。
   b. 在 Rotation schedule (輪換排程) 下，請以 UTC 時區輸入您的排程。
   c. 選擇 Rotate immediately when the secret is stored (儲存機密時立即輪換) 以在儲存變更時輪換您的機密。
   d. 在 Rotation function (輪換函數) 下，選擇 Create a new Lambda function (建立 Lambda 函數) 並輸入您的新函數名稱。機密管理員會將「SecretsManager」新增到函數名稱的開頭。
   e. 對於輪換策略，選擇單一使用者。
   f. 選擇儲存。

若要檢查機密是否輪換

1. 請開啟位於 https://console.aws.amazon.com/secretsmanager/ 的機密管理員控制台。
2. 選擇 Secrets (機密)，然後選擇該機密。
3. 在 Secret details (機密詳細資訊) 頁面，向下捲動並選擇 Retrieve secret value (擷取機密值)。

   如果機密值更改好了，則輪換成功。如果秘密值沒有更改，則需要輪換疑難排解查看旋轉功能的 CloudWatch 日誌。

測試您的應用程式在輪換的機密下是否能按預期運作。

後續步驟

從程式碼中刪除已寫死機密後，接下來需要思考一些構想：

- 您可以透過快取機密來提高效能並降低成本。如需詳細資訊，請參閱 擷取秘密。
- 您可以選擇不同的輪換計畫。如需詳細資訊，請參閱 the section called “排程表達式”。
- 若要在 Java 和 Python 應用程式中尋找硬式編碼的秘密，我們建議您使用 Amazon CodeGuru 審核者。

為 AWS Secrets Manager 設定交替使用者輪換

在此教學中，您將學習如何為包含資料庫憑證的秘密設定交替使用者輪換。交替使用者輪換是一種輪換策略，在該策略中，Secrets Manager 會複製使用者，然後交替使用更新的使用者憑證。如果您需要秘密高可用性，此策略是不錯的選擇，因為其中一個交替使用者具有資料庫的最新憑證，而另一個則正在更新。如需更多詳細資訊，請參閱 the section called “交替使用者”。

若要設定交替使用者輪換，您需要兩個秘密：

- 一個秘密包含您要輪換的憑證。
- 具有管理員憑證的第二個秘密。

  此使用者具有複製第一個使用者並變更第一個使用者密碼的許可。在本教學中，您有 Amazon RDS 為管理員使用者建立此秘密。Amazon RDS 也管理管理員密碼輪換。如需更多詳細資訊，請參閱 the section called “受管輪換”。

本教學的第一部分是設定一個真實的環境。為了向您展示輪換的運作方式，此教學使用了一個範例 Amazon RDS MySQL 資料庫。為安全起見，資料庫位於限制傳入網際網路存取的 VPC 中。若要透過網際網路從本機電腦連線至資料庫，請使用堡壘主機，這是 VPC 中可連線至資料庫的伺服器，而且允許從網際網路進行 SSH 連線。此教學中的堡壘主機是 Amazon EC2 執行個體，該執行個體的安全群組阻止其他類型的連線。

完成本教學後，建議您清除本教學的資源。請勿在生產環境中使用。

Secrets Manager 輪換使用 AWS Lambda 函數來更新機密和資料庫。如需有關使用 Lambda 函數成本的資訊，請參閱 定價。

教學：

- 許可

許可

對於教學必備條件，您需要 AWS 帳戶 的管理許可。在生產設定中，最佳實務是針對每個步驟使用不同的角色。例如，具有資料庫管理員許可的角色將會建立 Amazon RDS 資料庫，具有網路管理員許可的角色將設定 VPC 和安全群組。對於教學步驟，我們建議您繼續使用相同的身分。

如需如何在生產環境中設定許可的詳細資訊，請參閱 身分驗證與存取控制。

先決條件

在此教學課程中，您需執行下列項目：

先決條件 A：Amazon VPC

在此步驟中，您會建立 VPC，可在其中啟動 Amazon RDS 資料庫和 Amazon EC2 執行個體。在稍後的步驟中，您會使用電腦透過網際網路連線到堡壘，然後連線到資料庫，因此您需要允許流量傳出 VPC。為此，Amazon VPC 會將網際網路閘道連接至 VPC，並在路由表中新增路由，以便將目的地為 VPC 外的流量傳送至網際網路閘道。

在 VPC 中，您會建立 Secrets Manager 端點和 Amazon RDS 端點。在稍後的步驟中設定自動輪換時，Secrets Manager 會在 VPC 內建立 Lambda 輪換函數，以便能夠存取資料庫。Lambda 輪換函數也會呼叫 Secrets Manager 來更新秘密，並呼叫 Amazon RDS 來取得資料庫連線資訊。透過在 VPC 中建立端點，您可以確保從 Lambda 函數到 Secrets Manager 和 Amazon RDS 的呼叫不會離開 AWS 基礎設施，而是路由至 VPC 內的端點。

建立 VPC

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 選擇 Create VPC (建立 VPC)。
3. 在 Create VPC (建立 VPC) 頁面上，選擇 VPC and more (VPC 和更多)。
4. 在 Name tag auto-generation (自動產生名稱標籤) 的 Auto-generate (自動產生) 中，輸入 **`SecretsManagerTutorial`**。
5. 對於 DNS options (DNS 選項)，請選擇 **`Enable DNS hostnames`** 和 **`Enable DNS resolution`**。
6. 選擇 Create VPC (建立 VPC)。

在 VPC 內建立 Secrets Manager 端點

1. 在 Amazon VPC 主控台的 Endpoints (端點) 中，選擇 Create Endpoint (建立端點)。
2. 在 Endpoint settings (端點設定) 中，針對 Name (名稱) 輸入 **`SecretsManagerTutorialEndpoint`**。
3. 在 Services (服務) 中，輸入 **`secretsmanager`** 以篩選清單，然後在 AWS 區域 中選取 Secrets Manager 端點。例如，在美國東部 (維吉尼亞北部) 中，選擇 `com.amazonaws.us-east-1.secretsmanager`。
4. 對於 VPC，請選擇 **`vpc**** (SecretsManagerTutorial)`**。
5. 對於 Subnets (子網路)，選取所有 Availability Zones (可用區域)，然後針對每個選項選擇一個要包含的 Subnet ID (子網路 ID)。
6. 對於 IP address type (IP 地址類型)，請選擇 **`IPv4`**。
7. 對於 Security Groups (安全群組)，請選擇預設的安全群組。
8. 對於 Policy (政策)，請選擇 **`Full access`**。
9. 選擇 Create endpoint (建立端點)。

在 VPC 內建立 Amazon RDS 端點

1. 在 Amazon VPC 主控台的 Endpoints (端點) 中，選擇 Create Endpoint (建立端點)。
2. 在 Endpoint settings (端點設定) 中，針對 Name (名稱) 輸入 **`RDSTutorialEndpoint`**。
3. 在 Services (服務) 中，輸入 **`rds`** 以篩選清單，然後在 AWS 區域 中選取 Amazon RDS 端點。例如，在美國東部 (維吉尼亞北部) 中，選擇 `com.amazonaws.us-east-1.rds`。

4. 對於 VPC，請選擇 **vpc**** **(SecretsManagerTutorial)**。
5. 對於 Subnets (子網路)，選取所有 Availability Zones (可用區域)，然後針對每個選項選擇一個要包含的 Subnet ID (子網路 ID)。
6. 對於 IP address type (IP 地址類型)，請選擇 **IPv4**。
7. 對於 Security Groups (安全群組)，請選擇預設的安全群組。
8. 對於 Policy (政策)，請選擇 **Full access**。
9. 選擇 Create endpoint (建立端點)。

先決條件 B：Amazon EC2 執行個體

您在稍後步驟中建立的 Amazon RDS 資料庫會位於 VPC 中，因此若要存取，您需有堡壘主機。堡壘主機也位於 VPC 中，但在稍後的步驟中，您會設定安全群組，允許您的本機電腦使用 SSH 連線到堡壘主機。

為堡壘主機建立 EC2 執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
2. 選擇 Instances (執行個體)，然後選擇 Launch Instances (啟動執行個體)。
3. 在 Name and tags (名稱與標籤) 下，對於 Name (名稱)，輸入 **SecretsManagerTutorialInstance**。
4. 在 Application and OS Images (應用程式和作業系統映像) 中，保留預設值 **Amazon Linux 2 AMI (HMV) Kernel 5.10**。
5. 在 Instance type (執行個體類型) 中，保留預設值 **t2.micro**。
6. 在 Key pair (金鑰對) 中，選擇 Create key pair (建立金鑰對)。

   在 Create Key Pair (建立金鑰對) 對話方塊中，針對 Key pair name (金鑰對名稱) 輸入 **SecretsManagerTutorialKeyPair**，然後選擇 Create key pair (建立金鑰對)。

   系統會自動下載該金鑰對。
7. 在 Network settings (網路設定) 中，選擇 Edit (編輯)，接著執行下列動作：

   a. 對於 VPC，請選擇 **vpc-**** SecretsManagerTutorial**。
   b. 在 Auto-assign Public IP (自動指派公有 IP) 中，選擇 **Enable**。
   c. 對於 Firewall (防火牆)，請選擇 Select existing security group (選取現有的安全群組)。
   d. 對於 Common security groups (常見安全群組)，請選擇 **default**。

8. 選擇 Launch Instance (啟動執行個體)。

先決條件 C：管理員憑證的 Amazon RDS 資料庫和 Secrets Manager 秘密

在此步驟中，您會建立 Amazon RDS MySQL 資料庫並加以設定，以便 Amazon RDS 建立秘密以加入管理憑證。Amazon RDS 接著會自動為您管理輪換管理員秘密。如需更多詳細資訊，請參閱 受管輪換。

在建立資料庫時，您會指定在上一個步驟中建立的堡壘主機。Amazon RDS 接著會設定安全群組，以便資料庫和執行個體可彼此存取。您會將規則新增至連接執行個體的安全群組，允許本機電腦也連線到該執行個體。

使用包含管理員憑證的 Secrets Manager 秘密建立 Amazon RDS 資料庫

1. 在 Amazon RDS 主控台中，選擇 Create database (建立資料庫)。
2. 在 Engine options (引擎選項) 區段中，針對 Engine type (引擎類型) 選擇 **MySQL**。
3. 在 Templates (範本) 區段中，選擇 **Free tier**。
4. 在 Settings (設定) 區段中，執行下列動作：
   a. 對於 DB instance identifier (資料庫執行個體識別符)，請輸入 **SecretsManagerTutorial**。
   b. 在憑證設定下，選取在 AWS Secrets Manager 中管理主要憑證。
5. 在 Connectivity (連線) 區段中，針對 Computer resource (電腦資源)，選擇 Connect to an EC2 computer resource (連線到 EC2 電腦資源)，然後針對 EC2 Instance (EC2 執行個體)，選擇 **SecretsManagerTutorialInstance**。
6. 選擇 Create database (建立資料庫)。

先決條件 D：允許您的本機電腦連線到 EC2 執行個體

在此步驟中，您會將先決條件 B 中建立的 EC2 執行個體，設定為允許本機電腦連線。為此，您會編輯 Amazon RDS 在先決條件 C 中新增的安全群組，加入規則，允許電腦 IP 地址使用 SSH 連線。此規則允許您的本機電腦 (以您目前的 IP 地址識別) 透過網際網路使用 SSH 連線到堡壘主機。

允許您的本機電腦連線到 EC2 執行個體

1. 在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。

2. 在 EC2 執行個體 SecretsManagerTutorialInstance Security (安全) 索引標籤的 Security groups (安全群組) 中，選擇 **sg-*** (ec2-rds-X)**。
3. 在 Inbound Rules (傳入規則) 中，選擇 Edit inbound rules (編輯傳入規則)。
4. 選擇 Add rule (新增規則)，然後針對該規則執行下列動作：
   a. 針對 Type (類型)，選擇 **SSH**。
   b. 針對 Source type (來源類型)，選擇 **My IP**。

步驟 1：建立 Amazon RDS 資料庫使用者

首先，您需要其憑證存放在秘密中的使用者。若要建立使用者，請使用管理員憑證登入 Amazon RDS 資料庫。為簡單起見，在本教學中，您會建立具有資料庫完整許可的使用者。在生產環境中，這並不典型，建議您遵守最低權限原則。

若要連線至資料庫，請使用 MySQL 用戶端工具。在此教學中，您將使用 MySQL Workbench，這是一個 GUI 型應用程式。若要安裝 MySQL Workbench，請參閱 Download MySQL Workbench (下載 MySQL Workbench)。

若要連線至資料庫，請在 MySQL Workbench 中建立連線組態。對於組態，您需要 Amazon EC2 和 Amazon RDS 提供的部分資訊。

在 MySQL Workbench 中建立資料庫連線

1. 在 MySQL Workbench 的 MySQL Connections (MySQL 連線) 旁邊，選擇 (+) 按鈕。
2. 在 Setup New Connection (設定新連線) 對話方塊中，請執行下列動作：
   a. 對於 Connection Name (連線名稱)，請輸入 **SecretsManagerTutorial**。
   b. 對於 Connection Method (連線方法)，請選擇 **Standard TCP/IP over SSH**。
   c. 在 Parameters (參數) 索引標籤上，請執行下列動作：
      i. 對於 SSH Hostname (SSH 主機名稱)，請輸入 Amazon EC2 執行個體的公有 IP 地址。

         您可以在 Amazon EC2 主控台上，透過選擇 SecretsManagerTutorialInstance 執行個體來尋找 IP 地址。在 Public IPv4 DNS (公有 IPv4 DNS) 下複製 IP 地址。
      ii. 對於 SSH Username (SSH 使用者名稱)，請輸入 **ec2-user**。
      iii. 對於 SSH Keyfile (SSH 金鑰檔案)，請選擇您在之前的先決條件中下載的 SecretsManagerTutorialKeyPair.pem 金鑰對檔案。

iv. 對於 MySQL Hostname (MySQL 主機名稱)，請輸入 Amazon RDS 端點地址。

您可以在 Amazon RDS 主控台上，透過選擇 secretsmanagertutorialdb 資料庫執行個體來尋找端點地址。在 Endpoint (端點) 下複製地址。

v. 對於 User name (使用者名稱)，請輸入 **admin**。

d. 選擇 OK (確定)。

擷取管理員密碼

1. 在 Amazon RDS 主控台中，導覽至您的資料庫。
2. 在 Configuration (組態) 索引標籤的 Master Credentials ARN (主要憑證 ARN) 中，選擇 Manage in Secrets Manager (在 Secrets Manager 中管理)。

   Secrets Manager 主控台隨即開啟。
3. 在秘密詳細資訊頁面上，選擇 Retrieve secret value (擷取秘密值)。
4. 密碼會顯示在 Secret value (秘密值) 區段。

建立資料庫使用者

1. 在 MySQL Workbench 中，選擇 SecretsManagerTutorial 連線。
2. 輸入您從秘密中擷取的管理員密碼。
3. 在 MySQL Workbench 的 Query (查詢) 視窗中，輸入下列命令 (包括一個強密碼)，然後選擇 Execute (執行)。

```
CREATE DATABASE myDB;
CREATE USER 'appuser'@'%' IDENTIFIED BY 'EXAMPLE-PASSWORD';
GRANT ALL PRIVILEGES ON myDB . * TO 'appuser'@'%';
```

   在 Output (輸出) 視窗中，您會看到命令已成功。

步驟 2：為使用者的憑證建立秘密

接著，您可以建立一個秘密來存放剛建立使用者的憑證。這就是您將要輪換的秘密。開啟自動輪換，並指示交替使用者策略，您可以選擇單獨的進階使用者秘密，該秘密具有變更第一個使用者密碼的許可。

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。

2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：
   a. 對於 Secret type (秘密類型)，請選擇 Credentials for Amazon RDS database (Amazon RDS 資料庫的憑證)。
   b. 對於 Credentials (憑證)，請輸入使用者名稱 **appuser** 以及您為使用 MySQL Workbench 建立的資料庫使用者輸入的密碼。
   c. 對於 Database (資料庫)，請選擇 secretsmanagertutorialdb。
   d. 選擇 Next (下一步)。
4. 在 Configure secret (設定秘密) 頁面上，對於 Secret name (秘密名稱)，請輸入 **SecretsManagerTutorialAppuser**，然後選擇 Next (下一步)。
5. 在 Configure rotation (設定輪換) 頁面上，執行下列動作：
   a. 開啟 Automatic rotation (自動輪換)。
   b. 對於 Rotation schedule (輪換排程)，將排程設定為 Days (天數) ：**2** Duration (持續時間) ：**2h**。保持選取 Rotate immediately (立即輪換)。
   c. 對於 Rotation function (輪換函數)，請選擇 Create a rotation function (建立輪換函數)，然後對於函數名稱，請輸入 **tutorial-alternating-users-rotation**。
   d. 對於輪換策略，選擇交替使用者，然後在管理員憑證秘密之下，選擇名為 rds!cluster... 的秘密，其具有描述，其中包括您在本教學 **secretsmanagertutorial** 中所建立資料庫的名稱，例如 `Secret associated with primary RDS DB instance: arn:aws:rds:`*`Region`*`:`*`AccountId`*`:db:secretsmanagertutorial`。
   e. 選擇 Next (下一步)。
6. 在 Review (檢閱) 頁面，選擇 Store (存放)。

   Secrets Manager 會返回到秘密詳細資訊頁面。在頁面頂端，您可以看到輪換組態狀態。Secrets Manager 會使用 CloudFormation 建立資源，如 Lambda 輪換函數和執行 Lambda 函數的執行角色。在 CloudFormation 完成後，橫幅將變更為 Secret scheduled for rotation (排程輪換的秘密)。第一次輪換完成。

步驟 3：測試輪換的秘密

既然已輪換秘密，您可以檢查秘密是否包含有效的新憑證。秘密中的密碼已變更為原始憑證。

從秘密中擷取新密碼

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Secrets (秘密)，然後選擇秘密 **SecretsManagerTutorialAppuser**。
3. 在 Secret details (秘密詳細資訊) 頁面，向下捲動並選擇 Retrieve secret value (擷取秘密值)。
4. 在 Key/value (鍵/值) 中，複製 **password** 的 Secret value (秘密值)。

測試憑證

1. 在 MySQL Workbench 中，在 SecretsManagerTutorial 連線上按一下滑鼠右鍵，然後選擇 Edit Connection (編輯連線)。
2. 在 Manage Server Connections (管理伺服器連線) 對話方塊，對於 Username (使用者名稱)，請輸入 **appuser**，然後選擇 Close (關閉)。
3. 返回 MySQL Workbench 中，選擇 SecretsManagerTutorial 連線。
4. 在 Open SSH Connection (開啟 SSH 連線) 對話方塊中，對於 Password (密碼)，貼上從秘密擷取的密碼，然後選擇 OK (確定)。

   如果憑證有效，則 MySQL Workbench 會開啟資料庫的設計頁面。

這表明秘密輪換是成功的。秘密中的憑證已更新，並且是連線至資料庫的有效密碼。

步驟 4：清除資源

如果您想要嘗試另一種輪換策略單一使用者輪換，請跳過清除資源並前往 the section called “單一使用者輪換”。

否則，為避免潛在的費用，以及移除可存取網際網路的 EC2 執行個體，請刪除您在此教學中建立的下列資源及其先決條件：

- Amazon RDS 資料庫執行個體。如需說明，請參閱《Amazon RDS 使用者指南》中的刪除資料庫執行個體。
- Amazon EC2 執行個體。如需說明，請參閱《Amazon EC2 Linux 執行個體使用者指南》中的終止執行個體。
- Secrets Manager 秘密 `SecretsManagerTutorialAppuser`。如需說明，請參閱 the section called “刪除秘密”。
- Secrets Manager 端點。如需說明，請參閱《AWS PrivateLink 指南》中的刪除 VPC 端點。

- VPC 端點。如需說明，請參閱《AWS PrivateLink 指南》中刪除您的 VPC。

後續步驟

- 瞭解如何在應用程式中擷取秘密。
- 瞭解其他輪換排程。

為 AWS Secrets Manager 設定單一使用者輪換

在此教學中，您會學習如何為包含資料庫憑證的秘密，設定單一使用者輪換。單一使用者輪換是一種輪換策略，使用此策略，Secrets Manager 會在秘密和資料庫中更新使用者的憑證。如需更多詳細資訊，請參閱 the section called “單一使用者”。

完成本教學後，建議您清除本教學的資源。請勿在生產環境中使用。

Secrets Manager 輪換使用 AWS Lambda 函數來更新機密和資料庫。如需有關使用 Lambda 函數成本的資訊，請參閱 定價。

內容

許可

對於教學必備條件，您需要 AWS 帳戶 的管理許可。在生產設定中，最佳實務是針對每個步驟使用不同的角色。例如，具有資料庫管理員許可的角色將會建立 Amazon RDS 資料庫，具有網路管理員許可的角色將設定 VPC 和安全群組。對於教學步驟，我們建議您繼續使用相同的身分。

如需如何在生產環境中設定許可的詳細資訊，請參閱 身分驗證與存取控制。

先決條件

此教學的先決條件是 the section called “交替使用者輪換”。請勿在第一個教學結束時清除資源。在該教學之後，您會擁有一個真實環境，具有 Amazon RDS 資料庫和包含資料庫管理員憑證的 Secrets Manager 秘密。您也有第二個秘密，其中包含資料庫使用者的憑證，但在本教學中不會使用該秘密。

此外，您還在 MySQL Workbench 中設定了一個連線，以便使用管理員憑證連線至資料庫。

步驟 1：建立 Amazon RDS 資料庫使用者

首先，您需要其憑證存放在秘密中的使用者。若要建立使用者，請使用存放在秘密中的管理員憑證，登入 Amazon RDS 資料庫。為簡單起見，在本教學中，您會建立具有資料庫完整許可的使用者。在生產環境中，這並不典型，建議您遵守最低權限原則。

擷取管理員密碼

1. 在 Amazon RDS 主控台中，導覽至您的資料庫。
2. 在 Configuration (組態) 索引標籤的 Master Credentials ARN (主要憑證 ARN) 中，選擇 Manage in Secrets Manager (在 Secrets Manager 中管理)。

   Secrets Manager 主控台隨即開啟。
3. 在秘密詳細資訊頁面上，選擇 Retrieve secret value (擷取秘密值)。
4. 密碼會顯示在 Secret value (秘密值) 區段。

建立資料庫使用者

1. 在 MySQL Workbench 中，在 SecretsManagerTutorial 連線上按一下滑鼠右鍵，然後選擇 Edit Connection (編輯連線)。
2. 在 Manage Server Connections (管理伺服器連線) 對話方塊，對於 Username (使用者名稱)，請輸入 **admin**，然後選擇 Close (關閉)。
3. 返回 MySQL Workbench 中，選擇 SecretsManagerTutorial 連線。
4. 輸入您從秘密中擷取的管理員密碼。
5. 在 MySQL Workbench 的 Query (查詢) 視窗中，輸入下列命令 (包括一個強密碼)，然後選擇 Execute (執行)。

```
CREATE USER 'dbuser'@'%' IDENTIFIED BY 'EXAMPLE-PASSWORD';
```

```
GRANT ALL PRIVILEGES ON myDB . * TO 'dbuser'@'%';
```

在 Output (輸出) 視窗中，您會看到命令已成功。

步驟 2：為資料庫使用者憑證建立秘密

接著，您會建立秘密，存放剛剛所建立使用者的憑證，然後開啟自動輪換 (包括立即輪換)。Secrets Manager 會輪換秘密，這表示系統會以程式設計方式產生密碼，沒有人看過這組新密碼。立即開始輪換也可協助您判斷是否正確設定輪換。

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：
   a. 對於 Secret type (秘密類型)，請選擇 Credentials for Amazon RDS database (Amazon RDS 資料庫的憑證)。
   b. 對於 Credentials (憑證)，請輸入使用者名稱 **dbuser** 以及您為使用 MySQL Workbench 建立的資料庫使用者輸入的密碼。
   c. 對於 Database (資料庫)，請選擇 secretsmanagertutorialdb。
   d. 選擇 Next (下一步)。
4. 在 Configure secret (設定秘密) 頁面上，對於 Secret name (秘密名稱)，請輸入 **SecretsManagerTutorialDbuser**，然後選擇 Next (下一步)。
5. 在 Configure rotation (設定輪換) 頁面上，執行下列動作：
   a. 開啟 Automatic rotation (自動輪換)。
   b. 對於 Rotation schedule (輪換排程)，將排程設定為 Days (天數) ：**2** Duration (持續時間) ：**2h**。保持選取 Rotate immediately (立即輪換)。
   c. 對於 Rotation function (輪換函數)，請選擇 Create a rotation function (建立輪換函數)，然後對於函數名稱，請輸入 **tutorial-single-user-rotation**。
   d. 對於輪換策略，選擇單一使用者。
   e. 選擇 Next (下一步)。
6. 在 Review (檢閱) 頁面，選擇 Store (存放)。

Secrets Manager 會返回到秘密詳細資訊頁面。在頁面頂端，您可以看到輪換組態狀態。Secrets Manager 會使用 CloudFormation 建立資源，如 Lambda 輪換函數和執行 Lambda 函數的執行角

色。在 CloudFormation 完成後，橫幅將變更為 Secret scheduled for rotation (排程輪換的秘密)。第一次輪換完成。

步驟 3：測試輪換密碼

在第一次秘密輪換之後，這可能需要幾秒鐘，您可以檢查秘密是否仍包含有效憑證。秘密中的密碼已變更為原始憑證。

從秘密中擷取新密碼

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Secrets (秘密)，然後選擇秘密 **SecretsManagerTutorialDbuser**。
3. 在 Secret details (秘密詳細資訊) 頁面，向下捲動並選擇 Retrieve secret value (擷取秘密值)。
4. 在 Key/value (鍵/值) 中，複製 **password** 的 Secret value (秘密值)。

測試憑證

1. 在 MySQL Workbench 中，在 SecretsManagerTutorial 連線上按一下滑鼠右鍵，然後選擇 Edit Connection (編輯連線)。
2. 在 Manage Server Connections (管理伺服器連線) 對話方塊，對於 Username (使用者名稱)，請輸入 **dbuser**，然後選擇 Close (關閉)。
3. 返回 MySQL Workbench 中，選擇 SecretsManagerTutorial 連線。
4. 在 Open SSH Connection (開啟 SSH 連線) 對話方塊中，對於 Password (密碼)，貼上從秘密擷取的密碼，然後選擇 OK (確定)。

   如果憑證有效，則 MySQL Workbench 會開啟資料庫的設計頁面。

步驟 4：清除資源

為避免潛在的費用，請刪除您在此教學中建立的秘密。如需說明，請參閱 the section called “刪除秘密”。

若要清除在上一個教學中建立的資源，請參閱 the section called “步驟 4：清除資源”。

後續步驟

- 瞭解如何在應用程式中擷取秘密。請參閱擷取秘密。

- 瞭解其他輪換排程。請參閱 the section called “排程表達式”。

AWS Secrets Manager 的身分驗證與存取控制

Secrets Manager 會使用 AWS Identity and Access Management (IAM) 來保護秘密的存取權。IAM 提供身分驗證與存取控制。身分驗證會驗證個人請求的身分。Secrets Manager 會使用登入程序、密碼、存取金鑰和多重要素驗證 (MFA) 字符來驗證使用者的身分。請參閱登入 AWS。存取控制可確保只有經核准的個人可對 AWS 資源 (例如秘密) 執行操作。Secrets Manager 使用政策來定義誰可以存取哪些資源，以及相應身分可以對那些資源採取哪些動作。請參閱 IAM 中的政策和許可。

您可以使用 AWS Identity and Access Management Roles Anywhere 取得 IAM 中的暫時安全憑證，以用於在 AWS 外部執行的伺服器、容器和應用程式等工作負載。工作負載可以透過與 AWS 應用程式搭配使用的相同 IAM 政策和 IAM 角色來存取 AWS 資源。借助 IAM Roles Anywhere，您可以透過 Secrets Manager 來儲存和管理可供 AWS 中的資源存取的憑證，以及應用程式伺服器等內部部署裝置存取的憑證。如需詳細資訊，請參閱《IAM Roles Anywhere 使用者指南》。

Secrets Manager 管理員許可

若要授予 Secrets Manager 管理員許可，請遵循新增與移除 IAM 身分許可中的說明，並連接下列政策：

- SecretsManagerReadWrite
- IAMFullAccess

建議您不要將管理員許可授予最終使用者。雖然這樣做可讓使用者建立及管理其秘密，但啟用輪換所需的許可 (IAMFullAccess) 會授予最終使用者不適用的重要許可。

存取秘密的許可

您可以利用 IAM 許可政策，藉此控制可以存取秘密的使用者或服務。許可政策描述哪些人可在哪些資源上執行哪些動作。您可以：

- the section called “將許可政策連接至身分”
- the section called “將許可政策連接至秘密”

Lambda 輪換函數的許可

Secrets Manager 會使用 AWS Lambda 函數來輪換秘密。Lambda 函數必須能夠存取秘密，以及該秘密包含其憑證的資料庫或服務。請參閱 輪換的許可。

用於加密金鑰的許可

Secrets Manager 會使用 AWS Key Management Service (AWS KMS) 金鑰來加密秘密。AWS 受管金鑰 `aws/secretsmanager` 會自動擁有正確的許可。如果使用不同的 KMS 金鑰，Secrets Manager 需要該金鑰的許可。請參閱 the section called “KMS 金鑰的許可”。

將許可政策連接至身分

將許可政策連接到 IAM 身分：使用者、使用者群組和角色。在身分型政策中，您指定該身分可以存取哪些秘密以及該身分可以對該秘密執行哪些動作。如需更多資訊，請參閱《新增和移除 IAM 身分許可》。

您可以向代表其他服務中應用程式或使用者的角色授予許可。例如，在 Amazon EC2 執行個體上執行的應用程式可能需要存取資料庫。您可以建立與 EC2 執行個體設定檔相連的 IAM 角色，然後使用許可政策授予角色存取含有資料庫憑證的機密。如需相關資訊，請參閱《利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式》。其他您可以附加角色以加入 Amazon Redshift、AWS Lambda，和 Amazon ECS 的服務。

您也可以將許可授予經過 IAM 以外的身分系統驗證的使用者。例如，您可將 IAM 角色與使用 Amazon Cognito 登入的行動應用程式使用者建立關聯。角色將利用角色許可政策中的許可來授予應用程式暫時登入資料。然後，您可以使用許可政策將對秘密的存取權授予角色。如需相關資訊，請參閱《身分提供者和聯合》。

您可以使用身分型政策：

- 授予身分對多個秘密的存取權。
- 控制誰可以建立新秘密，以及誰可以存取尚未建立的秘密。
- 授予 IAM 群組對秘密的存取權。

如需更多詳細資訊，請參閱 the section called “許可政策範例”。

將許可政策連接至 AWS Secrets Manager 秘密

在資源型政策中，您可以指定能夠存取秘密的人員，以及他們可以對秘密執行的動作。您可以使用資源型政策：

- 將單一秘密的存取權授予多個使用者和角色。
- 將存取權授與其他 AWS 帳戶中的使用者或角色。

請參閱the section called “許可政策範例”。

在將資源型政策連接至主控台中的秘密時，Secrets Manager 會使用自動推理引擎 Zelkova 和 API `ValidateResourcePolicy`，防止您將秘密的存取權授予各種 IAM 委託人。您也可以透過 CLI 或開發套件呼叫帶有 `BlockPublicPolicy` 參數的 `PutResourcePolicy` API。

> **Important**
>
> 資源原則驗證和`BlockPublicPolicy`參數可防止透過直接附加至密碼的資源原則授予公用存取權，協助保護您的資源。除了使用這些功能之外，請仔細檢查下列原則，以確認它們不會授予公開存取權：
>
> - 附加至關聯 AWS 主體 (例如 IAM 角色) 的身分識別型政策
> - 附加至關聯 AWS 資源的以資源為基礎的政策 (例如， AWS Key Management Service (AWS KMS) 鍵)
>
> 若要檢閱密碼的權限，請參閱判斷誰有存取 秘密的許可。

若要檢視、變更或刪除秘密的資源政策 (主控台)

1. 於 https://console.aws.amazon.com/secretsmanager/ 開啟 Secrets Manager 主控台。
2. 從秘密清單中選擇秘密。
3. 在秘密詳細資訊頁面的概觀分頁上，在資源許可區段中，選擇編輯許可。
4. 在程式碼欄位中，執行以下其中一項作業，然後選擇 Save (儲存)：
    - 若要連接或修改資源政策，請輸入該政策。
    - 若要刪除政策，請清除程式碼欄位。

AWS CLI

Example 擷取資源政策

下列 get-resource-policy 範例會擷取連接至機密的以資源為基礎的政策。

```
aws secretsmanager get-resource-policy \
    --secret-id MyTestSecret
```

Example 刪除資源政策

下列 delete-resource-policy 範例會刪除連接至機密的以資源為基礎的政策。

```
aws secretsmanager delete-resource-policy \
    --secret-id MyTestSecret
```

Example 新增資源政策

下列 put-resource-policy 範例會將許可政策新增至機密，首先檢查政策是否不提供機密的廣泛存取權限。系統會從檔案讀取政策。若要取得更多資訊，請AWS CLI 參閱《 AWS CLI 使用指南》中的〈從檔案載入參數〉。

```
aws secretsmanager put-resource-policy \
    --secret-id MyTestSecret \
    --resource-policy file://mypolicy.json \
    --block-public-policy
```

mypolicy.json 的內容：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
```

```
    ]
}
```

AWS SDK

若要擷取與秘密相連的政策，請使用 GetResourcePolicy。

若要刪除與秘密相連的政策，請使用 DeleteResourcePolicy。

若要將政策連接至秘密，請使用 PutResourcePolicy。如果原本就已連接政策，命令會以新政策取而代之。政策必須格式化為 JSON 結構化文字。請參閱 JSON 政策文件結構。使用 the section called "許可政策範例" 開始撰寫政策。

如需更多詳細資訊，請參閱 the section called "AWS 開發套件"。

AWS 受管理的政策 AWS Secrets Manager

受 AWS 管理的策略是由建立和管理的獨立策略 AWS。 AWS 受管理的策略旨在為許多常見使用案例提供權限，以便您可以開始將權限指派給使用者、群組和角色。

請記住， AWS 受管理的政策可能不會為您的特定使用案例授與最低權限權限，因為這些權限可供所有 AWS 客戶使用。我們建議您定義使用案例專屬的客戶管理政策，以便進一步減少許可。

您無法變更受 AWS 管理策略中定義的權限。如果 AWS 更新 AWS 受管理原則中定義的權限，則此更新會影響附加原則的所有主體識別 (使用者、群組和角色)。 AWS 當新的啟動或新 AWS 服務 的 API 操作可用於現有服務時，最有可能更新 AWS 受管理策略。

如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策。

AWS 受管理的策略： SecretsManagerReadWrite

此政策提供讀取/寫入存取權 AWS Secrets Manager，包括描述 Amazon RDS、Amazon Redshift 和 Amazon DocumentDB 資源的權限，以及用於加密和解密密碼 AWS KMS 的許可。此政策還提供建立 AWS CloudFormation 變更集、從受管理的 Amazon S3 儲存貯體取得輪替範本 AWS、列出 AWS Lambda 函數以及說明 Amazon EC2 VPC 的許可。控制台需要這些許可，才能使用現有的輪換函數來設定輪換。

若要建立新的循環函數，您還必須擁有建立 AWS CloudFormation 堆疊和 AWS Lambda 執行角色的權限。您可以指派 IAM FullAccess 受管政策。請參閱輪換的許可。

許可詳細資訊

此政策包含以下許可。

- `secretsmanager` – 允許主體執行所有 Secrets Manager 動作。
- `cloudformation`— 允許主參與者建立 AWS CloudFormation 堆疊。這是必要的，以便使用控制台打開旋轉的主體可以通過 AWS CloudFormation 堆棧創建 Lambda 旋轉函數。如需詳細資訊，請參閱 the section called “Secrets Manager 使用 AWS CloudFormation 的方式”。
- `ec2` – 允許主體描述 Amazon EC2 VPC。此為必要項目，如此使用主控台的主體才可以在與存放在秘密中的憑證之資料庫相同的 VPC 中建立輪換函數。
- `kms`— 允許主參與者使用 AWS KMS 金鑰進行密碼編譯作業。此為必要項目，如此 Secrets Manager 才可以加密和解密秘密。如需詳細資訊，請參閱 the section called “秘密加密和解密”。
- `lambda` – 允許主體列出 Lambda 輪換函數。此為必要項目，如此使用主控台的主體才可以選擇現有的輪換函數。
- `rds` – 允許主體描述 Amazon RDS 中的叢集和執行個體。此為必要項目，如此使用主控台的主體才可以選擇 Amazon RDS 叢集或執行個體。
- `redshift` – 允許主體描述 Amazon Redshift 中的叢集。此為必要項目，如此使用主控台的主體才可以選擇 Amazon Redshift 叢集。
- `redshift-serverless`— 允許主體描述 Amazon Redshift 無伺服器中的命名空間。這是必要的，以便使用主控台的主體可以選擇 Amazon Redshift 無伺服器命名空間。
- `docdb-elastic` – 允許主體描述 Amazon DocumentDB 中的彈性叢集。此為必要項目，如此使用主控台的主體才可以選擇 Amazon DocumentDB 彈性叢集。
- `tag` – 允許主體取得帳戶中已標記的所有資源。
- `serverlessrepo`— 允許主參與者建立 AWS CloudFormation 變更集。此為必要項目，如此使用主控台的主體才可以建立 Lambda 輪換函數。如需詳細資訊，請參閱 the section called “Secrets Manager 使用 AWS CloudFormation 的方式”。
- `s3`— 允許主體從受管理的 Amazon S3 儲存貯體取得物件 AWS。此儲存貯體包含 Lambda 輪換函數範本。此為必要許可，如此使用主控台的主體才可以根據儲存貯體中的範本建立 Lambda 輪換函數。如需詳細資訊，請參閱 the section called “Secrets Manager 使用 AWS CloudFormation 的方式”。

若要檢視原則，請參閱 SecretsManagerReadWrite JSON 政策文件。

Secrets Manager 更新 AWS 受管理的策略

檢視有關 Secret 管理員受 AWS 管理策略的更新詳細資料。

變更	描述	日期
SecretsManagerReadWrite – 更新現有政策	此政策已更新，允許描述存取 Amazon Redshift 無伺服器，以便主控台使用者在建立 Amazon Redshift 密碼時可以選擇 Amazon Redshift 無伺服器命名空間。	2024年3月12日
SecretsManagerReadWrite – 更新現有政策	此政策已更新，允許描述 Amazon DocumentDB 彈性叢集的存取權，如此主控台使用者才可以在建立 Amazon DocumentDB 秘密時選擇彈性叢集。	2023 年 9 月 12 日
SecretsManagerReadWrite – 更新現有政策	此政策已更新，允許描述 Amazon Redshift 的存取權，如此主控台使用者才可以在建立 Amazon Redshift 秘密時選擇 Amazon Redshift 叢集。此更新還新增了新的許可，允許讀取存取由存放 Lambda 輪替函數範本 AWS 所管理的 Amazon S3 儲存貯體。	2020 年 6 月 24 日
SecretsManagerReadWrite – 更新現有政策	此政策已更新，允許描述 Amazon RDS 叢集的存取權，如此主控台使用者才可以在建立 Amazon RDS 秘密時選擇叢集。	2018 年 5 月 3 日
SecretsManagerReadWrite – 新政策	Secrets Manager 建立了一個政策來授予使用控制台所需的	2018 年 4 月 04 日

變更	描述	日期
	許可，這些許可具有 Secrets Manager 的所有讀取 / 寫入存取權。	
Secrets Manager 已開始追蹤變更	Secrets Manager 開始追蹤其 AWS 受管理政策的變更。	2018 年 4 月 04 日

判斷誰有存取 AWS Secrets Manager 秘密的許可

依預設，IAM 身分沒有存取秘密的許可。在授予秘密的存取權時，Secrets Manager 會評估與秘密相連的資源型政策，以及與 IAM 使用者或傳送請求的角色相連的所有身分型政策。為了這麼做，Secrets Manager 使用類似於 IAM 使用者指南中的判斷是否允許或拒絕請求中所述的程序。

多個政策套用到請求時，Secrets Manager 會使用階層來控制許可：

1. 如果任何政策中具有明確 deny 的陳述式與請求動作和資源相符：

   明確 deny 會覆寫其他所有內容並阻止該動作。
2. 如果沒有明確 deny，而是具有明確 allow 的陳述式與請求動作和資源相符：

   明確 allow 會授予請求中的動作對陳述式中資源的存取權。

   如果身分和秘密位於兩個不同的帳戶中，則在秘密的資源政策中和連接到身分的政策中都必須有 allow，否則 AWS 會拒絕該請求。如需更多詳細資訊，請參閱 跨帳戶存取。
3. 如果沒有具有明確 allow 的陳述式與請求動作和資源相符：

   AWS 預設會拒絕請求，這被稱為隱含拒絕。

若要檢視秘密的資源型政策

- 執行下列任意一項：
  - 前往以下位置開啟 Secrets Manager 主控台：https://console.aws.amazon.com/secretsmanager/。進入秘密的秘密詳細資訊頁面，在 Resource permissions (資源使用權限) 區段選擇 Edit permissions (編輯許可)。
  - 使用 AWS CLI 呼叫 `get-resource-policy`，或使用 AWS SDK 呼叫 `GetResourcePolicy`。

透過身分型政策判斷誰擁有存取權

- 使用 IAM 政策模擬器。請參閱使用 IAM 政策模擬器測試 IAM 政策

不同帳戶中使用者 AWS Secrets Manager 秘密的許可

若要允許一個帳戶中的使用者存取另一個帳戶中的秘密 (跨帳戶存取)，您必須同時允許在資源政策和身分政策中的存取權。這與授予和秘密所在的同一帳戶中的身分存取權不同。

您還必須允許身分使用秘密加密所用的 KMS 金鑰。這是因為您無法使用 AWS 受管金鑰(aws/secretsmanager) 來進行跨帳戶存取。您必須使用建立的 KMS 金鑰來加密秘密，然後將金鑰政策連接至秘密。建立 KMS 金鑰需支付費用。若要變更秘密的加密金鑰，請參閱 the section called “修改秘密”。

下列範例政策假設您在 Account1 中有秘密和加密金鑰，在 Account2 中有想要允許存取秘密值的身分。

步驟 1：將資源政策連接至 Account1 中的秘密

- 以下政策允許 *Account2* 中的 *ApplicationRole* 存取 *Account1* 中的秘密。若要使用此政策，請參閱 the section called “將許可政策連接至秘密”。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Account2:role/ApplicationRole"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

步驟 2：在 Account1 中將陳述式新增至 KMS 金鑰的金鑰政策

- 下列金鑰政策陳述式允許 *Account2* 中的 *ApplicationRole* 使用 *Account1* 中的 KMS 金鑰來解密 *Account1* 中的秘密。若要使用此陳述式，請將其新增至 KMS 金鑰的金鑰政策。如需詳細資訊，請參閱變更金鑰政策。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::Account2:role/ApplicationRole"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

步驟 3：將身分政策連接至 Account2 中的身分

- 以下政策允許 *Account2* 中的 *ApplicationRole* 存取 *Account1* 中的秘密，並透過使用 *Account1* 中也有的加密金鑰來解密秘密值。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。在 Secret ARN (秘密 ARN) 下的秘密詳細資訊頁面中，您可以從 Secrets Manager 主控台找到秘密的 ARN。或者，您也可以呼叫 describe-secret。

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "SecretARN"
    },
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "arn:aws:kms:Region:Account1:key/EncryptionKey"
    }
  ]
}
```

AWS Secrets Manager 的 Lambda 輪換函數執行角色許可

Secrets Manager 使用 Lambda 函數來輪換秘密。為了執行 Lambda 函數，Lambda 會擔任 IAM 執行角色，並將這些憑證提供給 Lambda 函數程式碼。如需如何設定自動輪換的指示，請參閱：

- 資料庫秘密的自動輪換 (主控台)
- 自動輪換 (主控台)
- 自動輪換 (AWS CLI)

下列範例顯示 Lambda 輪換函數執行角色的內嵌政策。若要建立執行角色並附加許可政策，請參閱 AWS Lambda 執行角色。

範例：

- Lambda 輪換函數執行角色的政策
- 客戶管理金鑰的政策陳述式
- 交替使用者策略的政策陳述式

Lambda 輪換函數執行角色的政策

下列範例政策允許輪換函數：

- 為 *SecretARN* 執行 Secrets Manager 作業。
- 建立新密碼。
- 如果資料庫或服務在 VPC 內執行，則設定必要組態。請參閱設定 Lambda 函數以存取 VPC 中的資源。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue",
                "secretsmanager:UpdateSecretVersionStage"
```

```
            ],
            "Resource": "SecretARN"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword"
            ],
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DetachNetworkInterface"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

客戶管理金鑰的政策陳述式

如果使用 KMS 金鑰為秘密加密，而不是 AWS 受管金鑰 `aws/secretsmanager`，那麼您需要將使用該金鑰的許可授予 Lambda 執行角色。您可以透過 SecretARN 加密內容來限制使用解密函數，從而使輪換函數角色僅有權解密其負責輪換的秘密。下列範例顯示要新增至執行角色政策，以使用 KMS 金鑰解密秘密的陳述式。

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:DescribeKey",
                "kms:GenerateDataKey"
            ],
            "Resource": "KMSKeyARN"
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:SecretARN": "SecretARN"
                }
            }
```

```
    }
```

若要針對使用客戶受管金鑰加密的多組秘密使用輪換函數，請新增如下列範例所示的陳述式，以允許執行角色將秘密解密。

```
    {
        "Effect": "Allow",
        "Action": [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:GenerateDataKey"
        ],
        "Resource": "KMSKeyARN"
        "Condition": {
            "StringEquals": {
                "kms:EncryptionContext:SecretARN": [
                    "arn1",
                    "arn2"
                ]
            }
        }
    }
```

交替使用者策略的政策陳述式

如需交替使用者輪換策略的資訊，請參閱the section called “輪換策略”。

對於包含 Amazon RDS 憑證的機密，如果您使用交替使用者策略，且超級使用者機密由 Amazon RDS 管理，則還必須允許輪換函數呼叫 Amazon RDS 上的唯讀 API，以便取得資料庫的連線資訊。建議您將 AWS 受管政策連接至 AmazonRDSReadOnlyAccess。

下列範例政策允許函數：

- 為 *SecretARN* 執行 Secrets Manager 作業。
- 擷取超級使用者秘密中的憑證。Secrets Manager 會使用超級使用者秘密中的憑證，更新輪換後秘密中的憑證。
- 建立新密碼。
- 如果資料庫或服務在 VPC 內執行，則設定必要組態。如需詳細資訊，請參閱設定 Lambda 函數以存取 VPC 中的資源。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue",
                "secretsmanager:UpdateSecretVersionStage"
            ],
            "Resource": "SecretARN"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "SuperuserSecretARN"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetRandomPassword"
            ],
            "Resource": "*"
        },
        {
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DetachNetworkInterface"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

AWS Secrets Manager 的許可政策範例

許可政策是 JSON 結構化文字。請參閱 JSON 政策文件結構。

連接至資源和身分的許可政策非常相似。您在政策中包含以存取秘密的一些元素包括：

- `Principal`：授予存取權的人。請參閱 IAM 使用者指南中的指定委託人。將政策連接到身分時，您不會在該政策中包含 `Principal` 元素。
- `Action`：他們可以做什麼。請參閱the section called “Secrets Manager 動作”。
- `Resource`：他們可以存取哪些秘密。請參閱 the section called “Secrets Manager 資源”。

  萬用字元 (*) 具有不同的意義，具體意義視政策的連接目標而定：

  - 在連接至秘密的政策中，* 表示該政策適用於此秘密。
  - 在連接至身分的政策中，* 表示該政策適用於帳戶中的所有資源 (包括秘密)。

若要將政策連接至秘密，請參閱 the section called “將許可政策連接至秘密”。

若要將政策連接至身分，請參閱 the section called “將許可政策連接至身分”。

主題

範例：擷取每個秘密值的許可

若要授予擷取秘密值的許可，您可以將政策連接至秘密或身分。如需判斷要使用哪種政策的說明，請參閱身分型政策和資源型政策。如需連接政策的相關資訊，請參閱 the section called “將許可政策連接至秘密” 和 the section called “將許可政策連接至身分”。

下列範例展示了授予對秘密的存取權的兩種不同方式。第一個範例是您可以連接至秘密的資源型政策。如果想要將對單一秘密的存取權授予多個使用者或角色，此範例非常有用。第二個範例是身分型政策，您可以將其連接至 IAM 中的使用者或角色。如果想要將存取權授予 IAM 群組，此範例很有用。若要授與批次 API 呼叫中擷取一組秘密的權限，請參閱the section called “在批次中擷取一組秘密值的許可”。

Example 讀取一個秘密 (連接至秘密)

您可以透過將下列政策連接至秘密，授予對秘密的存取權。若要使用此政策，請參閱 the section called “將許可政策連接至秘密”。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:role/EC2RoleToAccessSecrets"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

Example 讀取一個秘密 (連接至身分)

您可以透過將下列政策連接至身分，授予對秘密的存取權。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。如果您將此政策連接至角色 *EC2RoleToAccessSecrets*，則會授予與先前政策相同的許可。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "SecretARN"
    }
  ]
}
```

Example 讀取使用客戶管理的金鑰 (連接至身分) 加密的秘密

如果秘密是以客戶受管金鑰加密，您可以將下列政策連接至身分，授予讀取秘密的存取權。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "SecretARN"
    },
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "KMSKeyARN"
    }
  ]
}
```

在批次中擷取一組秘密值的許可

Example 讀取批次中的秘密群組 (連接至身分)

您可以透過將下列政策連接至身分，授予在 API 呼叫擷取一組秘密的存取權。此原則會限制呼叫者，以便他們只能擷取 *SecretARN1*、*SecretARN2* 和 *SecretARN3* 所指定的密碼，即使批次呼叫包含其他機密也是如此。如果呼叫者也要求批次 API 呼叫中的其他秘密，秘密管理員將不會傳回它們。如需更多詳細資訊，請參閱 the section called “在批次中擷取密碼”。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:BatchGetSecretValue",
        "secretsmanager:ListSecrets"
      ],
      "Resource": "*"
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "SecretARN1",
        "SecretARN2",
        "SecretARN3"
        ]
    }
  ]
}
```

範例：萬用字元

您可以使用萬用字元在政策元素中包含一組值。

Example 存取路徑中的所有秘密 (連接至身分)

下列政策會授予存取權來擷取名稱開頭為 "*TestEnv/*" 的所有秘密。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:Region:AccountId:secret:TestEnv/*"
  }
}
```

Example 存取所有秘密的中繼資料 (連接至身分)

下列政策授予 `DescribeSecret` 和開頭為 `List` (`ListSecrets` 及 `ListSecretVersionIds`) 的許可。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
```

```
    "Action": [
      "secretsmanager:DescribeSecret",
      "secretsmanager:List*"
    ],
    "Resource": "*"
  }
}
```

Example 比對秘密名稱 (連接至身分)

下列政策會依名稱授予秘密的所有 Secrets Manager 許可。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

若要比對秘密名稱，您可以將區域、帳戶 ID、秘密名稱和萬用字元 (?) 放在一起，以此比對單個隨機字元，從而建立秘密的 ARN。Secrets Manager 會將六個隨機字元連接到秘密名稱作為其 ARN 的一部分，因此您可以使用此萬用字元來比對那些字元。如果您使用的是語法 `"another_secret_name-*"`，則 Secrets Manager 不僅會比對含有 6 個隨機字元的所需秘密，還會比對 `"another_secret_name-<anything-here>a1b2c3"`。

您可以預測 ARN 除了 6 個隨機字元以外的所有部分，因此使用萬元字元 `'??????'` 語法可讓您安全地將許可授予尚不存在的秘密。不過，請注意，如果您刪除秘密，然後以相同名稱將其重建，使用者會自動收到新秘密的許可，即使那 6 個字元已變更。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:*",
      "Resource": [
        "arn:aws:secretsmanager:Region:AccountId:secret:a_specific_secret_name-a1b2c3",
        "arn:aws:secretsmanager:Region:AccountId:secret:another_secret_name-??????"
      ]
    }
  ]
}
```

範例：建立秘密的許可

若要將建立秘密的許可授予使用者，建議您將許可政策連接至使用者所屬的 IAM 群組。請參閱 IAM 使用者群組。

Example 建立秘密 (連接至身分)

下列政策會授予建立秘密和檢視秘密清單的許可。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret",
        "secretsmanager:ListSecrets"
      ],
      "Resource": "*"
    }
  ]
}
```

範例：許可和 VPC

如果您需要從 VPC 中存取 Secrets Manager，則可以透過在許可政策中包含條件，來確保對 Secrets Manager 的請求都來自 VPC。如需更多詳細資訊，請參閱 VPC 端點條件 及 VPC 端點。

確保來自其他 AWS 服務的存取秘密的請求也來自 VPC，否則此政策將拒絕其存取。

Example 要求請求透過 VPC 端點提出 (連接至秘密)

例如，以下政策允許使用者執行 Secrets Manager 操作，但僅限請求是透過 VPC 端點 *vpce-1234a5678b9012c* 提出的情況。若要使用此政策，請參閱 the section called “將許可政策連接至秘密”。

```
{
  "Id": "example-policy-1",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictGetSecretValueoperation",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "secretsmanager:GetSecretValue",
```

```
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpce": "vpce-1234a5678b9012c"
        }
      }
    }
  ]
}
```

Example 要求請求來自 VPC (連接至秘密)

以下政策只在命令來自 *vpc-12345678* 時，才允許這些命令建立和管理秘密。此外，此政策只在請求來自 vpc-2b2b2b2b 時，才允許存取秘密加密值的操作。如果您在某個 VPC 中執行應用程式，但您使用第二個隔離的 VPC 來執行管理功能，您可能會使用如下的政策。若要使用此政策，請參閱 the section called “將許可政策連接至秘密”。

```
{
  "Id": "example-policy-2",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAdministrativeActionsfromONLYvpc-12345678",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "secretsmanager:Create*",
        "secretsmanager:Put*",
        "secretsmanager:Update*",
        "secretsmanager:Delete*",
        "secretsmanager:Restore*",
        "secretsmanager:RotateSecret",
        "secretsmanager:CancelRotate*",
        "secretsmanager:TagResource",
        "secretsmanager:UntagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpc": "vpc-12345678"
        }
      }
    },
```

```
    {
      "Sid": "AllowSecretValueAccessfromONLYvpc-2b2b2b2b",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:sourceVpc": "vpc-2b2b2b2b"
        }
      }
    }
  ]
}
```

範例：使用標籤控制對秘密的存取

您可以使用標籤來控制對秘密的存取。使用標籤控制許可，在成長快速的環境中相當有幫助，也能在政策管理變得繁瑣時提供協助。其中一種策略是將標籤連接至秘密，然後在秘密具有特定標籤時將許可授予身分。

Example 允許使用特定標籤存取秘密 (連接至身分)

下列政策允許在帶有金鑰 *ServerName* 和值 *ServerABC* 之標籤的秘密上使用 DescribeSecret。若要使用此政策，請參閱 the section called “將許可政策連接至身分”。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "secretsmanager:ResourceTag/ServerName": "ServerABC"
      }
    }
  }
}
```

範例：使用符合秘密標籤的標籤限制對身分的存取

其中一種策略是將標籤同時連接至秘密和 IAM 身分。然後您可以建立許可政策，從而在身分的標籤與秘密的標籤相符時允許操作。如需完整的教學課程，請參閱根據標籤定義秘密的存取許可。

使用標籤控制許可，在成長快速的環境中相當有幫助，也能在政策管理變得繁瑣時提供協助。如需詳細資訊，請參閱 AWS 的 ABAC 是什麼？

Example 允許存取與秘密具有相同標籤的角色 (連接至秘密)

下列政策僅在標籤 *AccessProject* 的秘密和角色值相同時，才會向帳戶 *123456789012* 授予 GetSecretValue。若要使用此政策，請參閱 the section called “將許可政策連接至秘密”。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "AWS": "123456789012"
    },
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/AccessProject": "${ aws:PrincipalTag/AccessProject }"
      }
    },
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "*"
  }
}
```

範例：服務委託人

如果秘密中連接的資源政策包含 AWS 服務委託人，建議您使用 aws:SourceArn 和 aws:SourceAccount 全域條件金鑰。ARN 和帳戶值只會在請求從另一個 AWS 服務來到 Secrets Manager 中時，包含在授權內容中。此條件組合會避免潛在的混淆代理人案例。

若資源 ARN 包含資源政策中不允許的字元，則您便不能將該資源 ARN 用於 aws:SourceArn 條件金鑰的值中。請改用 aws:SourceAccount 條件金鑰。如需更多資訊，請參閱《IAM 需求》。

服務委託人通常不會用作附加到秘密的策略中的委託人，但有些 AWS 服務需要它。如需服務要求您連接至秘密之資源政策的相關資訊，請參閱服務的說明文件。

Example 允許服務使用服務委託人存取秘密 (連接至秘密)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "service-name.amazonaws.com"
        ]
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "aws:sourceArn": "arn:aws:service-name::123456789012:*"
        },
        "StringEquals": {
          "aws:sourceAccount": "123456789012"
        }
      }

    }
  ]
}
```

AWS Secrets Manager 的許可參考

若要查看組成許可政策的元素，請參閱 JSON 政策文件結構和 IAM JSON 政策元素參考。

若要開始撰寫許可政策，請參閱 the section called “許可政策範例”。

Secrets Manager 動作

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
CancelRotateSecret	准許取消進行中的秘密輪換	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
CreateSecret	准許建立秘密，用於存放可供查詢和輪換的加密資料	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:Name secretsmanager:Description secretsmanager:KmsKeyId aws:RequestTag/${TagKey} aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:AddReplicaRegions secretsmanager:For	

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				ceOverwriteReplicaSecret	
DeleteResourcePolicy	准許刪除連接至秘密的資源政策	許可管理	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
DeleteSecret	准許刪除秘密	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:RecoveryWindowInDays secretsmanager:ForceDeleteWithoutRecovery secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:Sec	

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				retPrimaryRegion	
DescribeSecret	准許擷取秘密的中繼資料 (非加密資料)	讀取	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetRandomPassword	准許產生用於建立密碼的隨機字串	讀取			
GetResourcePolicy	准許取得連接至秘密的資源政策	讀取	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetSecretValue	准許擷取加密資料並解密	讀取	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:VersionId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ListSecretVersionIds	准許列出可用的秘密版本	讀取	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ListSecrets	准許列出可用的秘密	列出			
PutResourcePolicy	准許將資源政策連接至秘密	許可管理	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:BlockPublicPolicy secretsmanager:SecretPrimaryRegion	
PutSecretValue	准許使用新的加密資料建立新的秘密版本	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RemoveRegionsFromReplication	准許從複寫移除區域	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ReplicateSecretToRegions	准許將現有的秘密轉換為多區域秘密，並開始將秘密複寫到新區域清單	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:AddReplicaRegions secretsmanager:ForceOverwri	

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				teReplicaSecret	
RestoreSecret	准許取消刪除秘密	寫入	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RotateSecret	准許開始輪換秘密	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:RotationLambdaARN secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:ModifyRotationRules	

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:RotateImmediately	
StopReplicationToReplica	准許從複寫移除秘密，並將秘密升級到複本區域中的區域秘密	寫入	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
TagResource	准許將標籤新增至秘密	標記	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId aws:RequestTag/${TagKey} aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
UntagResource	准許從秘密移除標籤	標記	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
UpdateSecret	准許使用新的中繼資料或新的加密資料版本更新秘密	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:Description secretsmanager:KmsKeyId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
	准許將一個秘密的階段移至另一個秘密	寫入	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
UpdateSecretVersionStage				secretsmanager:SecretId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ValidateResourcePolicy	准許在連接政策之前驗證資源政策	許可管理	Secret*		

動作	描述	存取層級	資源類型 (*必填項目)	條件索引鍵	相依動作
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Secrets Manager 資源

資源類型	ARN	條件索引鍵
Secret	`arn:${Partition}:secretsmanager:${Region}:${Account}:secret:${SecretId}`	aws:RequestTag/${TagKey}

資源類型	ARN	條件索引鍵
		aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:resource/AllowRotationLambdaArn

Secrets Manager 會在秘密名稱尾端附加破折號和六個隨機英數字元，以此建構秘密 ARN 的最後部分。如果您刪除秘密，然後以相同名稱重新建立另一個秘密，此格式化有助於確保對於原始秘密具有許可的個人不會自動取得新秘密的存取權，因為 Secrets Manager 會產生六個新的隨機字元。

您可以在秘密詳細資訊頁面上的 Secrets Manager 主控台中或呼叫 `DescribeSecret` 找到秘密的 ARN。

條件索引鍵

如果您在許可政策中包含下表中的字串條件，則 Secrets Manager 的呼叫者必須傳遞相符參數，否則他們的存取會遭到拒絕。若要避免因缺少參數而拒絕呼叫者，請新增 `IfExists` 至條件運算子名稱的末尾，例如 `StringLikeIfExists`。如需詳細資訊，請參閱 IAM JSON 政策元素：Condition 運算子。

條件索引鍵	描述	類型
aws:RequestTag/${TagKey}	依使用者對 Secrets Manager 服務所做請求中存在的索引鍵來篩選存取權	字串
aws:ResourceTag/${TagKey}	依與資源關聯的標籤來篩選存取權	字串
aws:TagKeys	依使用者對 Secrets Manager 服務所做請求中存在的所有標籤索引鍵名稱清單來篩選存取權	ArrayOfString

條件索引鍵	描述	類型
secretsmanager:AddReplicaRegions	依複寫秘密的區域清單來篩選存取權	ArrayOfString
secretsmanager:BlockPublicPolicy	依資源政策是否封鎖廣泛的 AWS 帳戶 存取來篩選存取權	Bool
secretsmanager:Description	依請求中的描述文字來篩選存取權	字串
secretsmanager:ForceDeleteWithoutRecovery	依是否立即刪除秘密，而不使用任何復原時段來篩選存取權	Bool
secretsmanager:ForceOverwriteReplicaSecret	依是否覆寫目的地區域中名稱相同的秘密來篩選存取權	Bool
secretsmanager:KmsKeyId	依請求中 KMS 金鑰的 ARN 來篩選存取權	字串
secretsmanager:ModifyRotationRules	依是否修改秘密的輪換規則篩選存取權	Bool
secretsmanager:Name	依請求中秘密的易記名稱來篩選存取權	字串
secretsmanager:RecoveryWindowInDays	依 Secrets Manager 在能夠刪除秘密前等待的天數來篩選存取權	數值

條件索引鍵	描述	類型
secretsmanager:ResourceTag/tag-key	依標籤鍵值組來篩選存取	字串
secretsmanager:RotateImmediately	依是否立即輪換秘密篩選存取權	Bool
secretsmanager:RotationLambdaARN	依請求中 Lambda 函數輪換的 ARN 來篩選存取權	ARN
secretsmanager:SecretId	依請求中的 SecretID 值來篩選存取權	ARN
secretsmanager:SecretPrimaryRegion	依建立秘密的主要區域來篩選存取權	字串
secretsmanager:VersionId	依請求中秘密版本的唯一識別符來篩選存取權	字串
secretsmanager:VersionStage	依請求中版本階段的清單來篩選存取權	字串
secretsmanager:resource/AllowRotationLambdaArn	依與秘密關聯之 Lambda 函數輪換的 ARN 來篩選存取權	ARN

使用 BlockPublicPolicy 條件封鎖對秘密的廣泛存取

在允許 PutResourcePolicy操作的身分政策中，我們建議您使用 BlockPublicPolicy: true。此條件意味著只有在政策不允許廣泛存取時，使用者才能將資源政策連接到秘密。

Secrets Manager 使用 Zelkova 自動推理來分析廣泛存取的資源政策。如需 Zelkova 的詳細資訊，請參閱 AWS 安全部落格上的 AWS 如何使用自動化推理來協助您達到大規模的安全性。

下列範例示範如何使用 BlockPublicPolicy。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "secretsmanager:PutResourcePolicy",
        "Resource": "SecretId",
        "Condition": {
            "Bool": {
                "secretsmanager:BlockPublicPolicy": "true"
            }
        }
    }
}
```

IP 地址條件

在允許或拒絕對 Secrets Manager 之存取權的同一政策陳述式中，指定 IP 地址條件運算子或 aws:SourceIp 條件索引鍵時，請謹慎小心。例如，如果您連接政策，該政策會限制來自公司網路 IP 地址範圍內的請求對秘密執行 AWS 動作，則當 IAM 使用者從公司網路叫用請求時，請求會如預期般運作。不過，如果您讓其他服務代表您存取秘密 (例如，當您使用 Lambda 函數啟用輪換時)，該函數會從 AWS 內部地址空間呼叫 Secrets Manager 操作。使用 IP 地址篩選條件時受到政策影響的請求會失敗。

此外，當請求來自 Amazon VPC 端點時，aws:sourceIP 條件金鑰較無效。若要將請求限制為特定 VPC 端點，請使用 the section called “VPC 端點條件”。

VPC 端點條件

若要允許或拒絕對特定 VPC 或 VPC 端點所發出之請求的存取權，請使用 aws:SourceVpc 來限制對指定 VPC 所發出之請求的存取權，或使用 aws:SourceVpce 來限制對指定 VPC 端點所發出之請求的存取權。請參閱 the section called “範例：許可和 VPC”。

- `aws:SourceVpc` 會限制對指定 VPC 所發出之請求的存取權。
- `aws:SourceVpce` 會限制對指定 VPC 端點所發出之請求的存取權。

如果您在允許或拒絕對 Secrets Manager 秘密之存取權的資源政策陳述式中使用這些條件金鑰，則您可能會不小心拒絕對使用 Secrets Manager 之服務的存取權，而這些服務會代表您存取秘密。只有一些 AWS 服務可在您的 VPC 內搭配端點執行。如果您將秘密的請求限制為來自 VPC 或 VPC 端點，則從未設定的服務呼叫 Secrets Manager 可能會失敗。

請參閱 VPC 端點。

使用 AWS Secrets Manager 建立和管理秘密

秘密可以是以加密形式存放在 Secrets Manager 中的一個密碼、一組憑證，例如使用者名稱和密碼、OAuth 字符或其他秘密資訊。

主題

建立資 AWS Secrets Manager 料庫密碼

在 Amazon RDS、Amazon Aurora、Amazon Redshift 或 Amazon DocumentDB 中建立使用者之後，您可以執行以下步驟來將憑證儲存在 Secrets Manager 中。當您使用 AWS CLI 或其中一個 SDK 來儲存密碼時，您必須以正確的 JSON 結構提供密碼。當您使用控制台存放資料庫機密時，機密管理員會自動在正確的 JSON 結構中建立。

> **Tip**
>
> 對於 Amazon RDS 和亞 Amazon Redshift 管理員使用者登入資料，我們建議您使用受管機密。您可以透過管理 servce 建立受管理的密碼，然後您可以使用受管理的輪換。

當您針對可複製到其他區域的來源資料庫儲存資料庫憑證時，機密會包含來源資料庫的連線資訊。如果隨後複製機密，則複本是來源機密的副本，並包含相同的連線資訊。您可以將其他金鑰/值對新增到區域連線資訊的機密。

若要建立密碼，您需要授與的權限SecretsManagerReadWriteAWS 受管理政策。

Secrets Manager 會在您建立密碼時產生 CloudTrail 記錄項目。如需詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要建立秘密 (主控台)

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：
   a. 針對 Secret type (密碼類型)，選擇要存放的資料庫憑證的類型：
      - Amazon RDS 資料庫 (包括 Aurora)
      - Amazon DocumentDB 資料庫
      - Amazon Redshift 資料倉儲
   b. 針對 Credentials (憑證)，輸入資料庫的憑證。
   c. 對於加密金鑰，請選擇 AWS KMS key Secrets Manager 用來加密密碼值的金鑰。如需詳細資訊，請參閱 秘密加密和解密。
      - 大多數情況下，選擇 aws/secretsmanager 以將 AWS 受管金鑰 用於 Secrets Manager。使用此金鑰無需任何成本。
      - 如果您需要從另一個機密存取密碼 AWS 帳戶，或者您想要使用自己的 KMS 金鑰以輪換金鑰或套用金鑰政策，請從清單中選擇客戶管理的金鑰，或選擇 [新增金鑰] 建立金鑰。如需有關使用客戶受管金鑰的成本的資訊，請參閱 定價。

        您必須擁有the section called “KMS 金鑰的許可”。如需跨帳戶存取權的詳細資訊，請參閱the section called “跨帳戶存取”。
   d. 如需資料庫，請選擇您的資料庫。
   e. 選擇 Next (下一步)。
4. 在 Configure secret (設定秘密) 頁面上，執行下列動作：
   a. 輸入描述性的 Secret name (機密名稱) 和 Description (描述)。秘密名稱必須含有 1 至 512 個 Unicode 字元。

b. (選用) 在 Tags (標籤) 區段，將標籤新增到秘密。如需標記策略，請參閱 the section called "標籤 秘密"。請勿在標籤中存放敏感資訊，因為標籤並未加密。

c. (選用) 若要將資源政策新增至秘密，請在 Resource permissions (資源使用權限) 中選擇 Edit permissions (編輯許可)。如需詳細資訊，請參閱 the section called "將許可政策連接至秘密"。

d. (選擇性) 在複製密碼中，若要將您的密碼複製到另一個機密 AWS 區域，請選擇複製密碼。您可以立即複寫秘密，也可以稍後返回複寫。如需詳細資訊，請參閱 將機密複寫到其他區域。

e. 選擇下一步。

5. (選用) 在 Configure rotation (設定輪換) 頁面上，可開啟自動輪換。您也可以暫時關閉輪換，稍後再將其開啟。如需詳細資訊，請參閱 輪換 秘密。選擇下一步。

6. 在 Review (檢閱) 頁面上，檢閱機密詳細資訊，然後選擇 Store (存放)。

   Secrets Manager 會傳回秘密清單。如果您的新秘密沒有顯示，請選擇重新整理按鈕。

AWS CLI

在命令 shell 中輸入命令時，存在命令歷史記錄被存取或公用程式存取命令參數的風險。請參閱the section called "降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險"。

Example 透過 JSON 檔案中的憑證建立機密

下列 `create-secret` 範例會透過檔案中的憑證建立機密。若要取得更多資訊，請AWS CLI 參閱《 AWS CLI 使用指南》中的〈從檔案載入參數〉。

為了使 Secrets Manager 能夠輪換秘密，您必須確保 JSON 與 機密的 JSON 結構 相符。

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --secret-string file://mycreds.json
```

mycreds.json 的內容：

```
{
    "engine": "mysql",
    "username": "saanvis",
    "password": "EXAMPLE-PASSWORD",
    "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
    "dbname": "myDatabase",
```

```
    "port": "3306"
}
```

AWS SDK

若要使用其中一個 AWS SDK 建立密碼，請使用`CreateSecret`動作。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

AWS Secrets Manager 機密資料的 JSON 結構

您可以透過 Secrets Manager 機密存放任何文字或二進位檔案。如果要為 Secrets Manager 機密開啟自動輪換，則機密必須在正確的 JSON 結構中。在輪換期間，Secrets Manager 會使用機密中的資訊連線至憑證來源，並更新其中的憑證。JSON 金鑰名稱區分大小寫。

請注意，當您使用主控台儲存資料庫機密時，Secrets Manager 會自動在正確的 JSON 結構中建立機密。

您可以將更多金鑰/值對新增至機密，例如在資料庫機密中，包含其他區域中複本資料庫的連線資訊。

主題

- Amazon RDS Db2 秘密結構
- Amazon RDS MariaDB 機密結構
- Amazon RDS 和 Amazon Aurora MySQL 的秘密結構
- Amazon RDS Oracle 機密結構
- Amazon RDS 和 Amazon Aurora PostgreSQL 的秘密結構
- Amazon RDS Microsoft SQLServer 機密結構
- Amazon DocumentDB 機密結構
- Amazon Redshift 機密結構
- Amazon Redshift 無服務器秘密結構
- Amazon ElastiCache 秘密結構

Amazon RDS Db2 秘密結構

對於 Amazon RDS Db2 執行個體，因為使用者無法變更自己的密碼，因此必須使用單獨的密碼來提供管理員登入資料。

```
{
  "engine": "db2",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 3306>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon RDS MariaDB 機密結構

```
{
  "engine": "mariadb",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 3306>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "mariadb",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 3306>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon RDS 和 Amazon Aurora MySQL 的秘密結構

```
{
  "engine": "mysql",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
```

```
  "port": <TCP port number. If not specified, defaults to 3306>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "mysql",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 3306>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon RDS Oracle 機密結構

```
{
  "engine": "oracle",
  "host": "<required: instance host name/resolvable DNS name>",
  "username": "<required: username>",
  "password": "<required: password>",
  "dbname": "<required: database name>",
  "port": <optional: TCP port number. If not specified, defaults to 1521>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "oracle",
  "host": "<required: instance host name/resolvable DNS name>",
  "username": "<required: username>",
  "password": "<required: password>",
  "dbname": "<required: database name>",
  "port": <optional: TCP port number. If not specified, defaults to 1521>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon RDS 和 Amazon Aurora PostgreSQL 的秘密結構

```
{
```

```
  "engine": "postgres",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to 'postgres'>",
  "port": <TCP port number. If not specified, defaults to 5432>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "postgres",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to 'postgres'>",
  "port": <TCP port number. If not specified, defaults to 5432>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon RDS Microsoft SQLServer 機密結構

```
{
  "engine": "sqlserver",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to 'master'>",
  "port": <TCP port number. If not specified, defaults to 1433>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "sqlserver",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to 'master'>",
  "port": <TCP port number. If not specified, defaults to 1433>,
  "masterarn": "<the ARN of the elevated secret>"
```

```
}
```

Amazon DocumentDB 機密結構

```
{
  "engine": "mongo",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 27017>,
  "ssl": <true|false. If not specified, defaults to false>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "mongo",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 27017>,
  "masterarn": "<the ARN of the elevated secret>",
  "ssl": <true|false. If not specified, defaults to false>
}
```

Amazon Redshift 機密結構

```
{
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 5439>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
```

```
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 5439>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon Redshift 無服務器秘密結構

```
{
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "namespaceName": <namespace name>,
  "port": <TCP port number. If not specified, defaults to 5439>
}
```

若要使用the section called “交替使用者”，請包含包含管理員或超級使用者認證的密碼。masterarn

```
{
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "namespaceName": <namespace name>,
  "port": <TCP port number. If not specified, defaults to 5439>,
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon ElastiCache 秘密結構

```
{
  "password": "<password>",
  "username": "<username>"
  "user_arn": "ARN of the Amazon EC2 user"
}
```

如需詳細資訊，請參閱 Amazon ElastiCache 使用者指南中的自動輪替使用者的密碼。

建立 AWS Secrets Manager 秘密

若要在 Secrets Manager 中存放 API 金鑰、存取權杖、不適用於資料庫的憑證和其他秘密，請依下列步驟操作。對於 Amazon ElastiCache 機密，如果您想要開啟輪換，則必須將密碼儲存在預期的 JSON 結構中。

若要建立秘密，您需要由 SecretsManagerReadWrite AWS 受管理政策 所授予的許可。

建立機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要建立秘密 (主控台)

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇 Store a new secret (存放新機密)。
3. 在 Choose secret type (選擇秘密類型) 頁面上，執行下列動作：

   a. 針對 Secret type (機密類型)，選擇 Other type of secret (其他機密類型)。

   b. 在鍵值對中，您可以在 JSON 鍵值對中輸入秘密，也可以先選擇純文字索引標籤，再以任何格式輸入秘密。秘密當中最多可以存放 65536 個位元組。

   c. 針對加密金鑰，選擇 Secrets Manager 用來加密機密值的 AWS KMS key。如需更多詳細資訊，請參閱 秘密加密和解密。

      - 大多數情況下，選擇 aws/secretsmanager 以將 AWS 受管金鑰 用於 Secrets Manager。使用此金鑰無需任何成本。
      - 如果您需要從另一個 AWS 帳戶 中存取秘密，或者如果您想要使用自己的 KMS 金鑰，以便您可以輪換它或套用金鑰政策，請從清單中選擇客戶管理的金鑰，或選擇 Add new key (新增新的金鑰) 以建立一個金鑰。如需有關使用客戶受管金鑰的成本的資訊，請參閱 定價。

        您必須擁有the section called “KMS 金鑰的許可”。如需跨帳戶存取權的詳細資訊，請參閱the section called “跨帳戶存取”。

   d. 選擇 Next (下一步)。
4. 在 Configure secret (設定秘密) 頁面上，執行下列動作：

   a. 輸入描述性的 Secret name (機密名稱) 和 Description (描述)。秘密名稱必須含有 1 至 512 個 Unicode 字元。

b. (選用) 在 Tags (標籤) 區段，將標籤新增到秘密。如需標記策略，請參閱 the section called "標籤 秘密"。請勿在標籤中存放敏感資訊，因為標籤並未加密。

c. (選用) 若要將資源政策新增至秘密，請在 Resource permissions (資源使用權限) 中選擇 Edit permissions (編輯許可)。如需更多詳細資訊，請參閱 the section called "將許可政策連接至秘密"。

d. (選用) 若要將秘密複寫到另一個 AWS 區域 中，請在 Replicate secret (複寫秘密) 中選擇 Replicate secret (複寫秘密)。您可以立即複寫秘密，也可以稍後返回複寫。如需更多詳細資訊，請參閱 將機密複寫到其他區域。

e. 選擇 Next (下一步)。

5. (選用) 在 Configure rotation (設定輪換) 頁面上，可開啟自動輪換。您也可以暫時關閉輪換，稍後再將其開啟。如需更多詳細資訊，請參閱 輪換 秘密。選擇 Next (下一步)。

6. 在 Review (檢閱) 頁面上，檢閱機密詳細資訊，然後選擇 Store (存放)。

   Secrets Manager 會傳回秘密清單。如果您的新秘密沒有顯示，請選擇重新整理按鈕。

AWS CLI

在命令 shell 中輸入命令時，存在命令歷史記錄被存取或公用程式存取命令參數的風險。請參閱 the section called "降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險"。

Example 建立秘密

下列 `create-secret` 範例會建立具有兩個金鑰值對的機密。

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --description "My test secret created with the CLI." \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

Example 透過 JSON 檔案中的憑證建立機密

下列 `create-secret` 範例會透過檔案中的憑證建立機密。如需詳細資訊，請參閱《AWS CLI 使用者指南》中的從檔案載入 AWS CLI 參數。

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --secret-string file://mycreds.json
```

mycreds.json 的內容：

```
{
    "username": "diegor",
    "password": "EXAMPLE-PASSWORD"
}
```

AWS SDK

若要使用其中一個 AWS 開發套件來建立秘密，請使用 `CreateSecret` 動作。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

更新 AWS Secrets Manager 秘密的值

若要更新秘密的值，您可以使用主控台、CLI 或軟體開發套件。在更新秘密值時，Secrets Manager 會使用預備標籤 AWSCURRENT 建立新的秘密版本。您仍然可以存取具有標籤 AWSPREVIOUS 的舊版本。您還可以新增自己的標籤。如需詳細資訊，請參閱 Secrets Manager 版本控制。

若要更秘密值 (主控台)

1. 於 https://console.aws.amazon.com/secretsmanager/ 開啟 Secrets Manager 主控台。
2. 從秘密清單中選擇秘密。
3. 在秘密詳細資訊頁面的概觀分頁上，在秘密值區段中，選擇擷取秘密值，然後選擇編輯。

AWS CLI

若要更新秘密值 (AWS CLI)

- 在命令 shell 中輸入命令時，存在命令歷史記錄被存取或公用程式存取命令參數的風險。請參閱the section called “降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險”。

  下列 `put-secret-value` 會建立具有兩個金鑰值對之新版本的機密。

  ```
  aws secretsmanager put-secret-value \
        --secret-id MyTestSecret \
        --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
  ```

以下 put-secret-value 會建立具有自訂預備標籤的新版本。新版本將具有標籤 MyLabel 和 AWSCURRENT。

```
aws secretsmanager put-secret-value \
      --secret-id MyTestSecret \
      --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
      --version-stages "MyLabel"
```

AWS SDK

建議您避免以超過每 10 分鐘一次的持續速率呼叫 PutSecretValue 或 UpdateSecret。在呼叫 PutSecretValue 和 UpdateSecret 更新機密值時，機密管理員會建立機密的新版本。當版本超過 100 個時，Secrets Manager 會移除未標記的版本，但不會移除建立時間不到 24 小時的版本。如果以超過每 10 分鐘一次的速率更新機密值，您建立的版本比機密管理員移除的版本更多，且您將達到機密版本的配額。

若要更新秘密值，請使用以下動作：UpdateSecret 或 PutSecretValue。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

變更密碼的加密金鑰 AWS Secrets Manager

Secrets Manager 使用包絡加密與 AWS KMS 金鑰和資料金鑰來保護每個密碼值。對於每個秘密，您可以選擇要使用的 KMS 金鑰。您可以使用 AWS 受管金鑰 aws/秘密管理器，也可以使用客戶管理的密鑰。在大多數情況下，我們建議使用 aws/secretsmanager，使用這個無需付費。如果您需要從其他人存取密碼 AWS 帳戶，或者想要使用自己的 KMS 金鑰，以便輪換或套用金鑰原則，請使用 客戶受管金鑰. 您必須擁有the section called “KMS 金鑰的許可”。如需有關使用客戶受管金鑰的成本的資訊，請參閱 定價。

您可以變更秘密的加密金鑰。例如，如果您想要從其他帳戶存取密碼，而該密碼目前已使用 AWS 受管理金鑰加密aws/secretsmanager，您可以切換到 客戶受管金鑰.

> **Tip**
>
> 如果您想要旋轉 客戶受管金鑰，我們建議您使用 AWS KMS 自動按鍵旋轉。如需詳細資訊，請參閱旋轉 AWS KMS 按鍵。

當您變更加密金鑰時，Secrets Manager 會使用新金鑰重新加密AWSCURRENT和AWSPREVIOUS版本。AWSPENDING為了避免將您鎖定在密碼之外，Secrets Manager 會使用先前的金鑰加密所有現有版本。這表示您可以使用先前的金鑰或新金鑰來解密AWSCURRENTAWSPENDING、和AWSPREVIOUS版本。

為了使其只AWSCURRENT能通過新的加密密鑰進行解密，請使用新密鑰創建一個新版本的密鑰。然後，為了能夠解密密鑰版本，您必須具有新密鑰的權限。AWSCURRENT

如果停用先前的加密金鑰，則除了 AWSCURRENT、AWSPENDING 和 AWSPREVIOUS 之外，您將無法解密任何秘密版本。如果您有其他要保留存取的標記秘密版本，則需要使用 the section called “AWS CLI”，以新的加密金鑰重新建立這些版本。

若要變更秘密的加密金鑰 (主控台)

1. 於 https://console.aws.amazon.com/secretsmanager/ 開啟 Secrets Manager 主控台。
2. 從秘密清單中選擇秘密。
3. 在秘密詳細資訊頁面上，在秘密詳細資訊區段中，選擇動作，然後選擇編輯加密金鑰。

AWS CLI

如果您變更秘密的加密金鑰，然後停用先前的加密金鑰，則除了 AWSCURRENT、AWSPENDING 和 AWSPREVIOUS 之外，您將無法解密任何秘密版本。如果您有其他要保留存取的標記秘密版本，則需要使用 the section called “AWS CLI”，以新的加密金鑰重新建立這些版本。

若要變更秘密的加密金鑰 (AWS CLI)

1. 下列 `update-secret` 範例會更新用於加密機密值的 KMS 金鑰。KMS 金鑰必須位於與機密相同的區域。

   ```
   aws secretsmanager update-secret \
         --secret-id MyTestSecret \
         --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
   ```

2. (選擇性) 如果您有具有自訂標籤的秘密版本，若要使用新金鑰重新加密，則您必須重新建立這些版本。

   在命令 shell 中輸入命令時，存在命令歷史記錄被存取或公用程式存取命令參數的風險。請參閱the section called “降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險”。

a. 取得秘密版本的值。

```
aws secretsmanager get-secret-value \
      --secret-id MyTestSecret \
      --version-stage MyCustomLabel
```

記下秘密值。

b. 使用該值建立新版本。

```
aws secretsmanager put-secret-value \
    --secret-id testDescriptionUpdate \
    --secret-string "SecretValue" \
    --version-stages "MyCustomLabel"
```

修改 AWS Secrets Manager 秘密

您可以在建立秘密後修改中繼資料，具體取決於建立秘密的人員。對於由其他服務建立的秘密，您可能需要使用其他服務來更新或輪換它。

若要決定管理秘密的人員，您可以檢閱秘密名稱。由其他服務管理的秘密以該服務的 ID 為字首。或者，在 AWS CLI 中，呼叫 describe-secret，然後檢閱 `OwningService` 欄位。如需更多詳細資訊，請參閱 由其他服務管理的秘密。

對於您管理的秘密，您可以修改說明、資源型政策、加密金鑰和標籤。雖然我們建議您使用輪換來更新含有憑證的秘密值，但您也可以變更加密的秘密值。輪換會同時更新 Secrets Manager 中的秘密以及資料庫或服務上的憑證。這可讓秘密自動同步，以便在用戶端請求秘密值時，隨時都能擷取運作中的一組憑證。如需更多詳細資訊，請參閱 輪換 秘密。

修改機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要更新您管理的秘密 (主控台)

1. 於 https://console.aws.amazon.com/secretsmanager/ 開啟 Secrets Manager 主控台。
2. 從秘密清單中選擇秘密。
3. 在秘密詳細資訊頁面上，執行下列動作：

   請注意，您無法變更秘密的名稱或 ARN。

- 若要更新描述，請在 Secrets details (秘密詳細資訊) 區段，選擇 Actions (動作)，然後選擇 Edit description (編輯描述)。
- 若要更新加密金鑰，請參閱 the section called “變更秘密的加密金鑰”。
- 若要更新標籤，在標籤分頁上，選擇編輯標籤。請參閱the section called “標籤 秘密”。
- 若要更新秘密值，請參閱 the section called “更新秘密值”。
- 若要更新密秘的許可，在概觀分頁上，選擇編輯許可。請參閱the section called “將許可政策連接至秘密”。
- 若要更新機密的輪換，在輪換分頁上，選擇編輯輪換。請參閱輪換 秘密。
- 若要將您的秘密複寫到其他區域，請參閱將機密複寫到其他區域。
- 如果您的秘密有複本，您可以變更複本的加密金鑰。在複寫分頁上，選取複本的選項按鈕，然後在動作選單中，選擇編輯加密金鑰 。請參閱the section called “秘密加密和解密”。
- 若要變更機密，使其由其他服務管理，您需要在該服務中重新建立機密。請參閱由其他服務管理的秘密。

AWS CLI

Example 更新機密描述

下列 `update-secret` 範例會更新機密的描述。

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --description "This is a new description for the secret."
```

AWS SDK

建議您避免以超過每 10 分鐘一次的持續速率呼叫 PutSecretValue 或 UpdateSecret。在呼叫 PutSecretValue 和 UpdateSecret 更新機密值時，機密管理員會建立機密的新版本。當版本超過 100 個時，Secrets Manager 會移除未標記的版本，但不會移除建立時間不到 24 小時的版本。如果以超過每 10 分鐘一次的速率更新機密值，您建立的版本比機密管理員移除的版本更多，且您將達到機密版本的配額。

若要更新秘密，請使用以下動作：`UpdateSecret` 或 `ReplicateSecretToRegions`。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

在 AWS Secrets Manager 中查找秘密

當您在不使用篩選條件的情況下搜尋秘密時，Secrets Manager 會比對秘密名稱、描述、標籤金鑰和標籤值中的關鍵字。不使用篩選條件進行搜尋並不區分大小寫，而且會忽略特殊字元，例如空格、/、_、=、#，而且只使用數字和字母。在沒有篩選條件的情況下進行搜尋時，Secrets Manager 會分析搜尋字串以將其轉換為單獨的單詞。單詞透過從大寫到小寫、從字母到數字，或從數字/字母到標點符號的任何變化來分隔。例如，輸入搜尋詞 `credsDatabase#892` 以在名稱、描述、標籤索引鍵和值中搜尋`creds`、`Database` 和 892。

列出機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

您可以將下列篩選條件套用至您的搜尋：

名稱

比對秘密名稱的開頭；區分大小寫。例如：Name: **`Data`** 會傳回名為 `DatabaseSecret` 的秘密，而不是 `databaseSecret` 或者 `MyData`。

描述

比對秘密描述中的字詞，不區分大小寫。例如：Description: **`My Description`** 會將秘密與下列說明進行比對：

- `My Description`
- `my description`
- `My basic description`
- `Description of my secret`

擁有服務

比對管理服務 ID 前綴的開頭，不區分大小寫。例如，**`my-ser`** 使用前綴 `my-serv` 和 `my-service` 比對由服務管理的機密。如需更多詳細資訊，請參閱 由其他服務管理的秘密。

已複寫的秘密

您可以篩選主要秘密、複本秘密或未複寫的秘密。

標籤金鑰

比對標籤金鑰的開頭；區分大小寫。例如：Tag key: **`Prod`** 返回帶有標籤 `Production` 和 `Prod1` 的秘密，而不是帶有標籤 `prod` 或者 `1 Prod` 的秘密。

標籤值

比對標籤值的開頭；區分大小寫。例如：Tag value: **Prod** 返回帶有標籤 Production 和 Prod1 的秘密，而不是帶有標籤值 prod 或者 1 Prod 的秘密。

Secrets Manager 是一項區域服務，僅傳回所選區域中的秘密。

AWS CLI

Example 列出帳戶中的機密

下列 list-secrets 範例會取得您帳戶中的機密清單。

```
aws secretsmanager list-secrets
```

Example 篩選帳戶中的機密清單

下列 list-secrets 範例會取得您帳戶中名稱包含 Test 的機密清單。依名稱篩選區分大小寫。

```
aws secretsmanager list-secrets \
    --filter Key="name",Values="Test"
```

Example 尋找由其他 AWS 服務管理的密碼

下列 list-secrets 範例會取得由服務管理的秘密清單。您可以依 ID 指定服務。如需更多詳細資訊，請參閱 由其他服務管理的秘密。

```
aws secretsmanager list-secrets --filter Key="owning-service",Values="<service ID
 prefix>"
```

AWS SDK

若要使用其中一個 AWS 開發套件來查找秘密，請使用 ListSecrets。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

刪除 AWS Secrets Manager 秘密

因為秘密本身具有的重要性，AWS Secrets Manager 特地讓刪除秘密的操作變得困難。Secrets Manager 不會立即刪除秘密。反之，Secrets Manager 會立即使秘密無法存取，並排定在最少七天的

復原時段後刪除。在復原時段結束之前，您都可以恢復之前刪除的秘密。不會針對您已標示為刪除的秘密收取任何費用。

如果主要秘密複寫到其他區域，則無法刪除它。先刪除複本，然後再刪除主要秘密。當您刪除複本時，會立即刪除它。

您無法直接刪除某個版本的秘密。但您可以使用 AWS CLI 或 AWS SDK 移除版本的所有預備標籤。這會將版本標示為已棄用，然後 Secrets Manager 可在背景自動刪除該版本。

如果您不知道應用程式是否仍在使用秘密，則可以建立 Amazon CloudWatch 警示，來提醒您任何在復原時段期間嘗試存取秘密的情況。如需詳細資訊，請參閱 使用 Amazon CloudWatch 監控排定刪除的 AWS Secrets Manager 秘密。

若要刪除秘密，您必須擁有 `secretsmanager:ListSecrets` 和 `secretsmanager:DeleteSecret` 許可。

刪除機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要刪除秘密 (主控台)

1. 前往以下位置開啟 Secrets Manager 主控台：https://console.aws.amazon.com/secretsmanager/。
2. 在秘密清單中，選擇您要刪除的秘密。
3. 在 Secrets details (秘密詳細資訊) 區段，選擇 Actions (動作)，然後選擇 Delete description (刪除描述)。
4. 在 Disable secret and schedule deletion (停用秘密和排程刪除) 對話方塊中，於 Waiting period (等待期) 下，輸入永久刪除前要等待的天數。Secrets Manager 會連接稱為 `DeletionDate` 的欄位，並將欄位設為目前的日期和時間，加上復原時段的指定天數。
5. 選擇 Schedule deletion (排定刪除)。

若要檢視已刪除的秘密

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 在 Secrets (秘密) 頁面上，選擇 Preferences (偏好設定) ()。
3. 在「偏好設定」對話方塊中，選取顯示排程刪除的機密，然後選擇儲存。

若要刪除複本秘密

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇主要秘密。
3. 在 Replicate Secret (複寫秘密) 區段中，選擇複本秘密。
4. 從 Actions (動作) 選單中，選擇 Delete Replica (刪除複本)。

AWS CLI

Example 刪除秘密

下列 `delete-secret` 範例會刪除機密。您可以在 DeletionDate 回應欄位中的日期和時間之前使用 `restore-secret` 復原機密。若要刪除複寫至其他區域的機密，請先使用 `remove-regions-from-replication` 移除複本，然後呼叫 `delete-secret`。

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --recovery-window-in-days 7
```

Example 立即刪除機密

下列 `delete-secret` 範例會在沒有復原時段的情況下立即刪除機密。您無法復原此機密。

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --force-delete-without-recovery
```

Example 刪除複本秘密

下列 `remove-regions-from-replication` 範例會刪除 eu-west-3 中的複本機密。若要刪除複寫至其他區域的主要機密，請先刪除複本，然後呼叫 `delete-secret`。

```
aws secretsmanager remove-regions-from-replication \
    --secret-id MyTestSecret \
    --remove-replica-regions eu-west-3
```

AWS SDK

若要刪除秘密，請使用 DeleteSecret 命令。若要刪除秘密的某個版本，請使用 UpdateSecretVersionStage 命令。若要刪除複本，請使用 StopReplicationToReplica 命令。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

還原 AWS Secrets Manager 秘密

Secrets Manager 會將排定刪除的秘密視為已棄用且您不再能夠直接存取。復原時段過後，Secrets Manager 會永久刪除秘密。一旦 Secrets Manager 刪除秘密，您就無法將其復原。復原時段結束之前，您可以復原秘密讓它再度成為可存取。這會移除 DeletionDate 欄位，因而取消排定的永久刪除。

若要在主控台中還原秘密和中繼資料，您必須擁有 secretsmanager:ListSecrets 和 secretsmanager:RestoreSecret 許可。

還原機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要還原秘密 (主控台)

1. 前往以下位置開啟 Secrets Manager 主控台：https://console.aws.amazon.com/secretsmanager/。
2. 在秘密清單中，選擇您想要還原的秘密。

   如果秘密清單中沒有出現刪除的秘密，請選擇 Preferences (偏好設定) ()。
   在「偏好設定」對話方塊中，選取顯示排程刪除的機密，然後選擇儲存。
3. 在 Secret details (秘密詳細資訊) 頁面中，選擇 Cancel deletion (取消刪除)。
4. 在 Cancel secret deletion (取消秘密刪除) 對話方塊中，選擇 Cancel deletion (取消刪除)。

AWS CLI

Example 還原先前刪除的機密

下列 restore-secret 範例會還原先前排程刪除的機密。

```
aws secretsmanager restore-secret \
```

```
    --secret-id MyTestSecret
```

AWS SDK

若要還原標記為刪除的秘密，請使用 `RestoreSecret` 命令。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

將 AWS Secrets Manager 機密複寫到其他 AWS 區域

您可以在多個 AWS 區域 中複寫機密，以支援橫跨這些區域的應用程式，從而滿足區域存取和低延遲需求。如果稍後需要，您可以將複本機密提升為獨立機密，然後將其獨立設定為複寫。機密管理員會複寫已加密的機密資料和中繼資料，例如跨越指定區域的標籤和資源政策。

複寫秘密的 ARN 與主要秘密的 ARN 基本相同，唯一差異在於 Region 部分，示例如下：

- 主要機密：`arn:aws:secretsmanager:Region1:123456789012:secret:MySecret-a1b2c3`
- 複本秘密：`arn:aws:secretsmanager:Region2:123456789012:secret:MySecret-a1b2c3`

如需複本機密的定價資訊，請參閱 AWS Secrets Manager 定價。

當您針對可複製到其他區域的來源資料庫儲存資料庫憑證時，機密會包含來源資料庫的連線資訊。如果隨後複製機密，則複本是來源機密的副本，並包含相同的連線資訊。您可以將其他金鑰/值對新增到區域連線資訊的機密。

如果您對主要機密開啟輪換，則機密管理員會在主要區域中輪換機密，而新的機密值會傳播至所有相關聯的複本機密。您不必單獨管理所有複本機密的輪換。

您可以在所有啟用的 AWS 區域中複寫機密。但是，如果在特殊的 AWS 區域 (例如 AWS GovCloud (US) 或中國等區域) 中使用 Secrets Manager，您只能在這些特殊 AWS 區域中設定機密和副本。您無法將啟用的 AWS 區域中的機密複寫到特定區域，或將機密從特定區域複寫到商業區域。

在您可以將機密複寫到另一個區域之前，必須先啟用該區域。如需詳細資訊，請參閱管理 AWS 區域。

透過調用儲存機密之區域中的機密管理員端點，您可以在無需複製的情況下跨多個區域使用機密。如需端點清單，請參閱 the section called “Secrets Manager 端點”。若要使用複寫來改善工作負載的復原力，請參閱AWS 上的災難復原 (DR) 架構，第一部分：雲端復原策略。

Secrets Manager 會在您複寫密碼時產生 CloudTrail 記錄項目。如需詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要將機密複寫到其他區域 (控制台)

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 從秘密清單中選擇秘密。
3. 在秘密詳細資訊頁面的複寫分頁上，執行以下其中一項操作：
   - 如果尚未複寫您的機密，請選擇 Replicate secret (複寫機密)。
   - 如果已複寫您的機密，請在 Replicate secret (複寫機密) 區段中選擇 Add Region (新增區域)。
4. 在 Add replica regions (新增複寫區域) 對話方塊中，執行下列操作：

   a. 針對 AWS Region (AWS 區域)，選擇您要複寫機密的目標 Region (區域)。

   b. (選用) 針對 Encryption key (加密金鑰)，選擇要用於將機密加密的 KMS 金鑰。金鑰必須在複本區域中。

   c. (選用) 若要新增另一個區域，請選擇 Add more regions (新增其他區域)。

   d. 選擇 Replicate (複寫)。

   返回機密詳細資訊頁面。在 Replicate Secret (複寫機密) 區段中，會顯示每個區域的 Replication Status (複寫狀態)。

AWS CLI

Example 將機密複寫至其他區域

下列 `replicate-secret-to-regions` 範例會將機密複寫至 eu-west-3。複本會使用 AWS 受管金鑰 aws/secretsmanager 進行加密。

```
aws secretsmanager replicate-secret-to-regions \
    --secret-id MyTestSecret \
    --add-replica-regions Region=eu-west-3
```

AWS SDK

若要複寫機密，請使用 `ReplicateSecretToRegions` 命令。如需詳細資訊，請參閱 the section called “AWS 開發套件”。

故障診斷

以下為複寫可能失敗的一些原因。

選取的區域中存在具有相同名稱的秘密。

若要解決此問題，您可以覆寫複本區域中的重複名稱機密。重試複寫，接著在重試複寫對話方塊中，選擇覆寫。

KMS 金鑰上沒有可用的許可來完成複寫

Secrets Manager 會先解密秘密，然後再使用複本區域中的新 KMS 金鑰重新加密。如果您沒有主要區域中的加密金鑰的 `kms:Decrypt` 許可，則會遇到此錯誤。要使用 KMS 密鑰以外的密鑰加密複製的秘密`aws/secretsmanager`，你需要 `kms:GenerateDataKey` 和 `kms:Encrypt` 密鑰。請參閱 the section called “KMS 金鑰的許可”。

KMS 金鑰已停用或找不到 KMS 金鑰

如果主要區域中的加密金鑰已停用或已刪除，Secret Manager 就無法複寫秘密。如果秘密具有使用已停用或已刪除的加密金鑰進行加密的自訂標記版本，則即便您已變更加密金鑰，仍可能發生此錯誤。如需 Secrets Manager 如何加密的相關資訊，請參閱 the section called “秘密加密和解密”。若要解決這個問題，您可以重新建立秘密版本，讓 Secrets Manager 使用目前的加密金鑰進行加密。如需詳細資訊，請參閱變更秘密的加密金鑰。然後重試複寫。

```
aws secretsmanager put-secret-value \
    --secret-id testDescriptionUpdate \
    --secret-string "SecretValue" \
    --version-stages "MyCustomLabel"
```

尚未啟用要進行複寫的區域

如需如何啟用區域的相關資訊，請參閱《AWS 帳戶管理參考指南》中的管理 AWS 區域。

在 AWS Secrets Manager 中將複本秘密提升為獨立秘密

複本秘密是從另一個 AWS 區域 的主要秘密中複寫的秘密。它具有與主要秘密相同的秘密值和中繼資料，但可以使用不同的 KMS 金鑰加密。複本秘密不能獨立於其主要秘密更新，但其加密金鑰除外。升級複本秘密會中斷複本秘密與主要秘密的連線，並將複本秘密設為獨立秘密。主要秘密的任何變更均不會再複寫到獨立秘密。

您可以在主要秘密無法使用時，將複本秘密提升為獨立的秘密，以此作為災難復原解決方案。或者，如果您要開啟複本的輪換功能，您可以將複本提升為獨立的秘密。

如果要提升複本，請務必更新對應的應用程式，以便使用獨立的秘密。

提升機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

若要提升複本秘密 (主控台)

1. 登入 Secrets Manager，網址為 https://console.aws.amazon.com/secretsmanager/。
2. 導覽至複本區域。
3. 在 Secrets (機密) 頁面上，選擇複本機密。
4. 在複本機密詳細資訊頁面上，選擇 Promote to standalone secret (提升為獨立機密)。
5. 在 Promote replica to standalone secret (將複本提升為獨立機密) 對話方塊中，輸入 Region (區域)，然後選擇 Promote replica (提升複本)。

AWS CLI

Example 將複本機密提升為主要機密

下列 `stop-replication-to-replica` 範例會移除複本機密至主要機密之間的連結。複本機密會提升為複本區域中的主要機密。您必須從複本區域內呼叫 `stop-replication-to-replica`。

```
aws secretsmanager stop-replication-to-replica \
    --secret-id MyTestSecret
```

AWS SDK

若要將複本秘密提升為獨立的秘密，請使用 `StopReplicationToReplica` 命令。您必須從複本秘密區域呼叫此命令。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

標籤 AWS Secrets Manager 秘密

Secrets Manager 會將標籤定義為一個內含您定義的金鑰和選用值的標籤。您可以使用這些標籤來輕鬆管理、搜尋和篩選 AWS 帳戶中的秘密和其他資源。當您標記秘密時，所有資源均使用標準命名方式。如需詳細資訊，請參閱《標記最佳實務》白皮書。

您可以檢查連接至秘密的標籤來授予或拒絕秘密的存取。如需更多詳細資訊，請參閱 the section called "範例：使用標籤控制對秘密的存取"。

您可以透過主控台、AWS CLI 和開發套件中的標籤來查找秘密。AWS 也提供 Resource Groups 工具，用來建立可根據標籤來整合並整理資源的自訂主控台。若要尋找具有特定標籤的秘密，請參閱 the section called "查找秘密"。Secrets Manager 不支援以標籤為基礎的成本分配。

不要在標籤內存放秘密的敏感資訊。

如需標籤配額和命名限制的詳細資訊，請參閱《AWS 一般參考指南》中的標記 Service Quotas。標籤會區分大小寫。

標記或取消標記機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called "使用 AWS CloudTrail 記錄 "。

若要變更秘密的標籤 (主控台)

1. 前往以下位置開啟 Secrets Manager 主控台：https://console.aws.amazon.com/secretsmanager/。
2. 從秘密清單中選擇秘密。
3. 在秘密詳細資訊頁面的標籤區段中，選擇編輯標籤。標籤金鑰名稱與值皆區分大小寫，且標籤金鑰必須是唯一的。

AWS CLI

Example 將標籤新增至機密

下列 `tag-resource` 範例顯示如何使用速記語法連接標籤。

```
aws secretsmanager tag-resource \
            --secret-id MyTestSecret \
            --tags Key=FirstTag,Value=FirstValue
```

Example 將多個標籤新增至機密

下列 `tag-resource` 範例會將兩個金鑰值標籤連接至機密。

```
aws secretsmanager tag-resource \
            --secret-id MyTestSecret \
```

```
            --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",
 "Value": "SecondValue"}]'
```

Example 從機密移除標籤

下列 untag-resource 範例會從機密中移除兩個標籤。對於每個標籤，金鑰和值都會移除。

```
aws secretsmanager untag-resource \
            --secret-id MyTestSecret \
            --tag-keys '[ "FirstTag", "SecondTag"]'
```

AWS SDK

若要變更秘密的標籤，請使用 TagResource 或 UntagResource。如需更多詳細資訊，請參閱 the section called “AWS 開發套件”。

從 AWS Secrets Manager 中擷取秘密

您可以擷取您的秘密：

- 在程式碼中
- 在其他服務中
- 在 AWS CLI 中
- 在 AWS 主控台中

擷取機密時，Secrets Manager 會產生 CloudTrail 日誌項目。如需更多詳細資訊，請參閱 the section called “使用 AWS CloudTrail 記錄 ”。

在程式碼中

在應用程式中，您可以透過呼叫 `GetSecretValue` 或 `BatchGetSecretValue`在任何 AWS SDK 中擷取秘密。如需範例，請參閱 AWSSDK 程式碼範例程式庫中的取得秘密值。不過，建議您使用用戶端快取來快取您的秘密值。快取秘密可提高速度並降低成本。

- 對於 Java 應用程式：
  - 如果將資料庫憑證存放在秘密中，則請使用 Secrets Manager SQL 連線驅動程式，以使用秘密中的憑證連線至資料庫。
  - 對於其他類型的秘密，請使用 Secrets Manager Java 型快取元件或使用 `GetSecretValue` 直接呼叫 SDK。
- 對於 Python 應用程式，請使用 Secrets Manager Python 型快取元件或使用 `get_secret_value` 或 `batch_get_secret_value` 直接呼叫 SDK。
- 對於 .NET 應用程式，請使用 Secrets Manager .NET 型快取元件或使用 `GetSecretValue` 或 `BatchGetSecretValue` 直接呼叫 SDK。
- 對於 Go 應用程式，請使用 Secrets Manager Go 型快取元件或使用 `GetSecretValue` 或 `BatchGetSecretValue` 直接呼叫 SDK。
- 對於 JavaScript 應用程序，請使用 `getSecretValue` 或 `batchGetSecretValue` 直接呼叫 SDK。
- 對於 PHP 應用程序，請使用 `GetSecretValue` 或 `BatchGetSecretValue` 直接呼叫 SDK。
- 對於 Ruby 應用程序，請使用 `get_secret_value` 或 `batch_get_secret_value` 直接呼叫 SDK。

- 對於 GitHub Actions，請參閱the section called “GitHub 工作”。

在其他系統和 AWS 服務中

您也可以在下列項目中擷取秘密：

- 若為 AWS Batch，您可以在任務定義中參考秘密。
- 若為 AWS CloudFormation，您可以在 CloudFormation 堆疊中建立秘密和參考秘密。
- 若為 Amazon ECS，您可以在容器定義中參考秘密。
- 對於 Amazon EKS，您可以使用 AWS Secrets and Configuration Provider (ASCP) 在 Amazon EKS 中將秘密作為檔案掛載。
- 對於 GitHub，您可以使用 Secrets Manager GitHub 動作，在 GitHub 工作中將秘密新增為環境變數。
- 若為 AWS IoT Greengrass，您可以在 Greengrass 群組中參考秘密。
- 針對 AWS Lambda，您可以在 Lambda 函數中引用秘密。
- 若為 Parameter Store，您可以在參數中 參考秘密。

AWS CLI

Example 擷取機密的加密機密值

下列 `get-secret-value` 範例會取得目前機密值。

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
```

Example 擷取先前的機密值

下列 `get-secret-value` 範例會取得先前的機密值。

```
aws secretsmanager get-secret-value \
        --secret-id MyTestSecret
        --version-stage AWSPREVIOUS
```

AWS 主控台

若要擷取秘密 (主控台)

1. 前往以下位置開啟 Secrets Manager 主控台：https://console.aws.amazon.com/secretsmanager/。
2. 在秘密清單中，選擇您想要擷取的秘密。
3. 在 Secret value (秘密值) 區段，選擇 Retrieve secret value (擷取秘密值)。

   Secrets Manager 會顯示秘密的目前版本 (AWSCURRENT)。若要查看秘密的其他版本，例如 AWSPREVIOUS 或自訂標記版本，請使用 the section called “AWS CLI”。

從批次擷取一組密碼 AWS Secrets Manager

Secrets Manager 提供批次 API，BatchGetSecretValue以便在一個 API 呼叫中擷取一組密碼。若要選擇要擷取的密碼，您可以依名稱或 ARN 指定密碼清單，也可以使用篩選器。如果 Secrets Manager 遇到錯誤，例如嘗試擷取任何密碼AccessDeniedException時，您可以在回應Errors中看到錯誤。

批次擷取密碼的許可

您必須擁有您想要擷取的每個秘密的 secretsmanager:GetSecretValue 許可。您也必須擁有 secretsmanager:BatchGetSecretValue 許可。如果您使用過濾器，則還必須擁有 secretsmanager:ListSecrets。如需許可政策範例，請參閱 the section called “在批次中擷取一組秘密值的許可”。

> **Important**
>
> 如果您的 VPCE 原則拒絕擷取您要擷取之群組中個別密碼的權限，則 BatchGetSecretValue 不會傳回任何秘密值，而且會傳回錯誤。

AWS CLI

Example 擷取按名稱列出的密碼群組的密碼值

下列 batch-get-secret-value 範例會為三個秘密取得目前機密值。

```
aws secretsmanager batch-get-secret-value \
          --secret-id-list MySecret1 MySecret2 MySecret3
```

Example 擷取按篩選條件列出的密碼群組的密碼值

下列 batch-get-secret-value 範例會取得具有名為「Test」之標籤之密碼的密碼值。

```
aws secretsmanager batch-get-secret-value \
          --filters Key="tag-key",Values="Test"
```

使用憑證連線至 AWS Secrets Manager 機密中的 SQL 資料庫

在 Java 應用程式中，您可以使用 Secrets Manager SQL 連線驅動程式來連線到 MySQL、PostgreSQL、甲骨文、MSSQLServer、Db2 和 Redshift 資料庫，使用儲存在 Secrets Manager 中的認證。每個驅動程式都會包裝基本 JDBC 驅動程式，因此您可以使用 JDBC 呼叫來存取資料庫。但是，並非傳遞用於連線的使用者名稱和密碼，而是提供機密的 ID。該驅動程式調用機密管理員擷取機密值，然後使用機密中的憑證連線至資料庫。驅動程式還使用 Java 用戶端快取庫快取憑證，因此未來的連線不需要呼叫 Secrets Manager。預設快取每小時重新整理一次機密，並在輪換機密時重新整理一次。若要設定快取，請參閱 the section called “SecretCacheConfiguration”。

您可以從中下載源代碼GitHub。

使用機密管理員 SQL 連線驅動程式：

- 您的應用程式必須使用 Java 8 或更高版本。
- 您的機密必須為下列之一：
  - 預期 JSON 結構中的資料庫機密。若要查看該格式，請在機密管理員控制台中檢視您的機密，然後選擇 Retrieve secret value (擷取機密值)。或者，在 AWS CLI呼叫中get-secret-value。
  - Amazon RDS 受管機密。對於此類型的機密，您必須在建立連線時指定端點和連接埠。
  - 一個 Amazon Redshift 管理的秘密。對於此類型的機密，您必須在建立連線時指定端點和連接埠。

如果要將您的資料庫複製到其他區域，以連線到另一個區域中的複副本資料庫，請在建立連線時指定區域端點和端口。您可以將區域連線資訊作為額外的金鑰/值對、SSM 參數儲存參數或程式碼設定中儲存至機密中。

若要將驅動程式新增至您的專案中，請在 Maven 建置檔案 pom.xml 中，為驅動程式新增以下相依性。如需詳細資訊，請參閱 Maven Central Repository 網站上的 Secrets Manager SQL 連線庫。

```
<dependency>
    <groupId>com.amazonaws.secretsmanager</groupId>
    <artifactId>aws-secretsmanager-jdbc</artifactId>
    <version>1.0.12</version>
</dependency>
```

此驅動程式使用預設憑證供應商鏈。如果您在 Amazon EKS 上執行驅動程式，則其可能會取得正在執行之節點的憑證，而不是服務帳戶角色。為瞭解決此問題，請將 com.amazonaws:aws-java-sdk-sts 的版本 1 新增至 Gradle 或 Maven 專案檔案作為相依性。

若要在secretsmanager.properties檔案中設定 AWS PrivateLink DNS 端點 URL 和區域：

```
drivers.vpcEndpointUrl = endpoint URL
drivers.vpcEndpointRegion = endpoint region
```

若要覆寫主要區域，請設定 AWS_SECRET_JDBC_REGION 環境變數或對 secretsmanager.properties 檔案進行下列變更：

```
drivers.region = region
```

必要許可：

- secretsmanager:DescribeSecret
- secretsmanager:GetSecretValue

如需詳細資訊，請參閱 許可參考。

範例：

- 建立與資料庫的連線
- 透過指定端點和連接埠來建立連線
- 使用 c3p0 連線集區建立連線
- 使用 c3p0 連線集區透過指定端點和連接埠來建立連線

建立與資料庫的連線

以下範例顯示如何使用機密中的憑證和連線資訊建立與資料庫的連線。連線後，您就可使用 JDBC 呼叫來存取資料庫。如需詳細資訊，請參閱 Java 文件網站上的 JDBC 基礎知識。

MySQL

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerMySQLDriver" ).newInstance

// Retrieve the connection info from the secret using the secret ARN
String URL = "secretId";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

PostgreSQL

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerPostgreSQLDriver" ).newIns

// Retrieve the connection info from the secret using the secret ARN
String URL = "secretId";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

Oracle

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerOracleDriver" ).newInstanc

// Retrieve the connection info from the secret using the secret ARN
String URL = "secretId";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
```

```
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

MSSQLServer

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerMSSQLServerDriver" ).newIn

// Retrieve the connection info from the secret using the secret ARN
String URL = "secretId";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

Db2

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerDb2Driver" ).newInstance()

// Retrieve the connection info from the secret using the secret ARN
String URL = "secretId";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

Redshift

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerRedshiftDriver" ).newInsta
```

```
// Retrieve the connection info from the secret using the secret ARN
String URL = "secretId";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

透過指定端點和連接埠來建立連線

下列範例顯示如何使用機密中的憑證與您指定的端點和連接埠建立資料庫的連線。

Amazon RDS 受管機密不包括資料庫的端點和連接埠。若要使用 Amazon RDS 管理之機密中的主要憑證連線至資料庫，請在程式碼中指定這些憑證。

複製到其他區域的機密可以改善區域資料庫連線的延遲，但其不包含與來源機密不同的連線資訊。每個複本都是來源機密的副本。若要在機密中儲存區域連線資訊，請為區域的端點和端口資訊新增更多金鑰/值對。

連線後，您就可使用 JDBC 呼叫來存取資料庫。如需詳細資訊，請參閱 Java 文件網站上的 JDBC 基礎知識。

MySQL

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerMySQLDriver" ).newInstance

// Set the endpoint and port. You can also retrieve it from a key/value pair in the
 secret.
String URL = "jdbc-secretsmanager:mysql://example.com:3306";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

PostgreSQL

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerPostgreSQLDriver" ).newIns

// Set the endpoint and port. You can also retrieve it from a key/value pair in the
 secret.
String URL = "jdbc-secretsmanager:postgresql://example.com:5432/database";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

Oracle

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerOracleDriver" ).newInstanc

// Set the endpoint and port. You can also retrieve it from a key/value pair in the
 secret.
String URL = "jdbc-secretsmanager:oracle:thin:@example.com:1521/ORCL";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

MSSQLServer

```
// Load the JDBC driver
Class.forName( "com.amazonaws.secretsmanager.sql.AWSSecretsManagerMSSQLServerDriver" ).newIn

// Set the endpoint and port. You can also retrieve it from a key/value pair in the
 secret.
String URL = "jdbc-secretsmanager:sqlserver://example.com:1433";
```

```
// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

Db2

```
// Load the JDBC driver
Class.forName( "com.amazonaws.com.amazonaws.secretsmanager.sql.AWSSecretsManagerDb2Driver" )

// Set the endpoint and port. You can also retrieve it from a key/value pair in the
 secret.
String URL = "jdbc-secretsmanager:db2://example.com:50000";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

Redshift

```
// Load the JDBC driver
Class.forName( "com.amazonaws.com.amazonaws.secretsmanager.sql.AWSSecretsManagerRedshiftDriv

// Set the endpoint and port. You can also retrieve it from a key/value pair in the
 secret.
String URL = "jdbc-secretsmanager:redshift://example.com:5439";

// Populate the user property with the secret ARN to retrieve user and password from
 the secret
Properties info = new Properties( );
info.put( "user", "secretId" );

// Establish the connection
conn = DriverManager.getConnection(URL, info);
```

使用 c3p0 連線集區建立連線

下列範例顯示如何使用 `c3p0.properties` 檔案建立連線集區，此檔案使用驅動程式透過機密擷取憑證和連線資訊。對於 `user` 和 `jdbcUrl`，輸入機密 ID 以設定連線集區。然後，您可以從集區中擷取連線並將其用作任何其他資料庫連線。如需詳細資訊，請參閱 Java 文件網站上的 JDBC 基礎知識。

如需 c3p0 的詳細資訊，請參閱 Machinery For Change 網站上的 c3p0。

MySQL

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerMySQLDriver
c3p0.jdbcUrl=secretId
```

PostgreSQL

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerPostgreSQLDriver
c3p0.jdbcUrl=secretId
```

Oracle

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerOracleDriver
c3p0.jdbcUrl=secretId
```

MSSQLServer

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerMSSQLServerDriver
c3p0.jdbcUrl=secretId
```

Db2

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerDb2Driver
c3p0.jdbcUrl=secretId
```

Redshift

```
c3p0.user=secretId
```

```
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerRedshiftDriver
c3p0.jdbcUrl=secretId
```

使用 c3p0 連線集區透過指定端點和連接埠來建立連線

下列範例顯示如何使用 `c3p0.properties` 檔案建立連線集區，此檔案使用驅動程式透過您指定的端點和連接埠擷取機密中的憑證。然後，您可以從集區中擷取連線並將其用作任何其他資料庫連線。如需詳細資訊，請參閱 Java 文件網站上的 JDBC 基礎知識。

Amazon RDS 受管機密不包括資料庫的端點和連接埠。若要使用 Amazon RDS 管理之機密中的主要憑證連線至資料庫，請在程式碼中指定這些憑證。

複製到其他區域的機密可以改善區域資料庫連線的延遲，但其不包含與來源機密不同的連線資訊。每個複本都是來源機密的副本。若要在機密中儲存區域連線資訊，請為區域的端點和端口資訊新增更多金鑰/值對。

MySQL

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerMySQLDriver
c3p0.jdbcUrl=jdbc-secretsmanager:mysql://example.com:3306
```

PostgreSQL

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerPostgreSQLDriver
c3p0.jdbcUrl=jdbc-secretsmanager:postgresql://example.com:5432/database
```

Oracle

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerOracleDriver
c3p0.jdbcUrl=jdbc-secretsmanager:oracle:thin:@example.com:1521/ORCL
```

MSSQLServer

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerMSSQLServerDriver
c3p0.jdbcUrl=jdbc-secretsmanager:sqlserver://example.com:1433
```

Db2

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerDb2Driver
c3p0.jdbcUrl=jdbc-secretsmanager:db2://example.com:50000
```

Redshift

```
c3p0.user=secretId
c3p0.driverClass=com.amazonaws.secretsmanager.sql.AWSSecretsManagerRedshiftDriver
c3p0.jdbcUrl=jdbc-secretsmanager:redshift://example.com:5439
```

擷取 Java 應用程式中的 AWS Secrets Manager 秘密

擷取秘密時，您可以使用 Secrets Manager Java 型快取元件進行快取以供將來使用。擷取快取的秘密比從 Secrets Manager 中擷取要快。由於呼叫 Secrets Manager API 需要花費成本，因此使用快取可以降低成本。如需您可以擷取機密的所有方法，請參閱 擷取秘密。

快取政策是「最近最少使用的 (LRU)」，因此當快取必須丟棄秘密時，其會丟棄最近最少使用的秘密。預設情況下，快取每小時重新整理一次秘密。您可以設定在快取中重新整理機密的頻率，並可以掛接到機密擷取以新增更多功能。

一旦釋放快取參考，快取不會強制執行垃圾回收。快取實作不包括快取失效。快取實作著重於快取本身，而不是強化或著重於安全性。如果您需要額外的安全性，例如加密快取中的項目，請使用提供的介面和抽象方法。

若要使用元件，您必須擁有下列項目：

- Java 8 或更高版本的開發環境。請參閱 Oracle 網站上的 Java SE 下載。
- 適用於 Java 的 AWS SDK 1.x。您可以在自己的專案中使用這兩種適用於 Java 的 AWS SDK 版本。如需詳細資訊，請參閱 Using the SDK for Java 1.x and 2.x side-by-side (並用適用於 Java 的 SDK 1.x 和 2.x)。

若要下載開放程式碼，請參閱 Secrets Manager Java 型快取用戶端元件 (在 GitHub 上)。

要將元件新增至您的專案中，請在 Maven pom.xml 檔案中包含以下相依性。如需 Maven 的詳細資訊，請參閱 Apache Maven 專案網站上的入門指南。

```
<dependency>
```

```
  <groupId>com.amazonaws.secretsmanager</groupId>
  <artifactId>aws-secretsmanager-caching-java</artifactId>
  <version>1.0.2</version>
</dependency>
```

必要許可：

- `secretsmanager:DescribeSecret`
- `secretsmanager:GetSecretValue`

如需更多詳細資訊，請參閱 許可參考。

參考

- SecretCache
- SecretCacheConfiguration
- SecretCacheHook

Example 擷取秘密

下列程式碼範例顯示擷取秘密字串的 Lambda 函數。它遵循將函數處理常式之外的快取執行個體化的最佳實務，所以如果您再次呼叫 Lambda 函數，則不會繼續呼叫 API。

```
package com.amazonaws.secretsmanager.caching.examples;

    import com.amazonaws.services.lambda.runtime.Context;
    import com.amazonaws.services.lambda.runtime.RequestHandler;
    import com.amazonaws.services.lambda.runtime.LambdaLogger;

    import com.amazonaws.secretsmanager.caching.SecretCache;

    public class SampleClass implements RequestHandler<String, String> {

         private final SecretCache cache  = new SecretCache();

         @Override public String handleRequest(String secretId,  Context context) {
             final String secret  = cache.getSecretString(secretId);

            // Use the secret, return success;

        }
```

```
    }
```

SecretCache

從 Secrets Manager 請求的秘密記憶體內快取。您使用 the section called “getSecretString” 或 the section called “getSecretBinary” 從快取中擷取秘密。您可以透過在建構函式的 the section called “SecretCacheConfiguration” 物件中傳遞，設定快取設定。

如需包含範例的詳細資訊，請參閱 the section called “Java 應用程式”。

建構函式

`public SecretCache()`

SecretCache 物件的預設建構函式。

`public SecretCache(AWSSecretsManagerClientBuilder builder)`

使用以提供的 `AWSSecretsManagerClientBuilder` 建立的 Secrets Manager 用戶端建構新快取。使用此建構函式可以自訂 Secrets Manager 用戶端，例如使用特定區域或端點。

`public SecretCache(AWSSecretsManager client)`

使用提供的 `AWSSecretsManagerClient` 建構新的秘密快取。使用此建構函式可以自訂 Secrets Manager 用戶端，例如使用特定區域或端點。

`public SecretCache(SecretCacheConfiguration config)`

使用提供的 `the section called “SecretCacheConfiguration”` 建構新的秘密快取。

方法

getSecretString

`public String getSecretString(final String secretId)`

從 Secrets Manager 中擷取字串秘密。傳回 `String`。

getSecretBinary

`public ByteBuffer getSecretBinary(final String secretId)`

從 Secrets Manager 中擷取二進位秘密。傳回 `ByteBuffer`。

refreshNow

```
public boolean refreshNow(final String secretId) throws
InterruptedException
```

強制重新整理快取。如果重新整理完成且沒有錯誤，則會傳回 `true`，否則會傳回 `false`。

關閉

```
public void close()
```

關閉快取。

SecretCacheConfiguration

the section called “SecretCache” 的快取組態選項，例如最大快取大小和快取秘密的存留時間 (TTL)。

建構函數

```
public SecretCacheConfiguration
```

`SecretCacheConfiguration` 物件的預設建構函式。

方法

getClient

```
public AWSSecretsManager getClient()
```

傳回 `AWSSecretsManagerClient`，快取從中擷取秘密。

setClient

```
public void setClient(AWSSecretsManager client)
```

傳回 `AWSSecretsManagerClient` 用戶端，快取從中擷取秘密。

getCacheHook

```
public SecretCacheHook getCacheHook()
```

傳回用於與快取更新掛鉤的 the section called “SecretCacheHook” 介面。

setCacheHook

```
public void setCacheHook(SecretCacheHook cacheHook)
```

設定用於與快取更新掛鉤的 the section called “SecretCacheHook” 介面。

getMaxCacheSize

```
public int getMaxCacheSize()
```

傳回最大快取大小。預設值為 1024 個秘密。

setMaxCacheSize

```
public void setMaxCacheSize(int maxCacheSize)
```

設定最大快取大小。預設值為 1024 個秘密。

getCacheItemTTL

```
public long getCacheItemTTL()
```

傳回快取項目的 TTL (以毫秒為單位)。當快取的秘密超過此 TTL 時，快取會從 `AWSSecretsManagerClient` 擷取秘密的新複本。預設值為 1 小時 (以毫秒為單位)。

當在 TTL 之後請求秘密時，快取會同步重新整理秘密。如果同步重新整理失敗，則快取會傳回過時的秘密。

setCacheItemTTL

```
public void setCacheItemTTL(long cacheItemTTL)
```

設定快取項目的 TTL (以毫秒為單位)。當快取的秘密超過此 TTL 時，快取會從 `AWSSecretsManagerClient` 擷取秘密的新複本。預設值為 1 小時 (以毫秒為單位)。

getVersionStage

```
public String getVersionStage()
```

傳回要快取的秘密版本。如需詳細資訊，請參閱秘密版本。預設為 `"AWSCURRENT"`。

setVersionStage

```
public void setVersionStage(String versionStage)
```

設定要快取的秘密版本。如需詳細資訊，請參閱秘密版本。預設為 `"AWSCURRENT"`。

SecretCacheConfiguration withClient

```
public SecretCacheConfiguration withClient(AWSSecretsManager client)
```

設定 AWSSecretsManagerClient，以從中擷取秘密。使用新設定傳回已更新的 SecretCacheConfiguration 物件。

SecretCacheConfiguration withCacheHook

```
public SecretCacheConfiguration withCacheHook(SecretCacheHook cacheHook)
```

設定用於與記憶體內快取掛鉤的介面。使用新設定傳回已更新的 SecretCacheConfiguration 物件。

SecretCacheConfiguration withMaxCacheSize

```
public SecretCacheConfiguration withMaxCacheSize(int maxCacheSize)
```

設定最大快取大小。使用新設定傳回已更新的 SecretCacheConfiguration 物件。

SecretCacheConfiguration withCacheItemTTL

```
public SecretCacheConfiguration withCacheItemTTL(long cacheItemTTL)
```

設定快取項目的 TTL (以毫秒為單位)。當快取的秘密超過此 TTL 時，快取會從 AWSSecretsManagerClient 擷取秘密的新複本。預設值為 1 小時 (以毫秒為單位)。使用新設定傳回已更新的 SecretCacheConfiguration 物件。

SecretCacheConfiguration withVersionStage

```
public SecretCacheConfiguration withVersionStage(String versionStage)
```

設定要快取的秘密版本。如需詳細資訊，請參閱秘密版本。使用新設定傳回已更新的 SecretCacheConfiguration 物件。

SecretCacheHook

用於掛接到 the section called “SecretCache”，以對存放在快取中的秘密執行動作的介面。

put

```
Object put(final Object o)
```

準備要存放在快取中的物件。

傳回要存放在快取中的物件。

get

```
Object get(final Object cachedObject)
```

從快取的物件中衍生物件。

傳回要從快取傳回的物件

擷取 Python 應用程式中的 AWS Secrets Manager 秘密

擷取秘密時，您可以使用 Secrets Manager Python 型快取元件進行快取以供將來使用。擷取快取的秘密比從 Secrets Manager 中擷取要快。由於呼叫 Secrets Manager API 需要花費成本，因此使用快取可以降低成本。如需您可以擷取機密的所有方法，請參閱 擷取秘密。

快取政策是「最近最少使用的 (LRU)」，因此當快取必須丟棄秘密時，其會丟棄最近最少使用的秘密。預設情況下，快取每小時重新整理一次秘密。您可以設定在快取中重新整理機密的頻率，並可以掛接到機密擷取以新增更多功能。

一旦釋放快取參考，快取不會強制執行垃圾回收。快取實作不包括快取失效。快取實作著重於快取本身，而不是強化或著重於安全性。如果您需要額外的安全性，例如加密快取中的項目，請使用提供的介面和抽象方法。

若要使用元件，您必須擁有下列項目：

- Python 3.6 或更高版本。
- botocore 1.12 或更高版本。請參閱 AWS SDK for Python 和 Botocore。
- setuptools_scm 3.2 或更高版本。請參閱 https://pypi.org/project/setuptools-scm/。

若要下載開放程式碼，請參閱 Secrets Manager Python 型快取用戶端元件 (在 GitHub 上)。

若要安裝元件，請使用下列命令。

```
$ pip install aws-secretsmanager-caching
```

必要許可：

- `secretsmanager:DescribeSecret`
- `secretsmanager:GetSecretValue`

如需更多詳細資訊，請參閱 許可參考。

參考

- SecretCache
- SecretCacheConfig
- SecretCacheHook
- @InjectSecretString
- @InjectKeywordedSecretString

Example 擷取秘密

以下範例顯示如何為名為 *mysecret* 的秘密取得秘密值。

```
import botocore
import botocore.session
from aws_secretsmanager_caching import SecretCache, SecretCacheConfig

client = botocore.session.get_session().create_client('secretsmanager')
cache_config = SecretCacheConfig()
cache = SecretCache( config = cache_config, client = client)

secret = cache.get_secret_string('mysecret')
```

SecretCache

從 Secrets Manager 擷取的秘密記憶體內快取。您使用 the section called “get_secret_string” 或 the section called “get_secret_binary” 從快取中擷取秘密。您可以透過在建構函式的 the section called “SecretCacheConfig” 物件中傳遞，設定快取設定。

如需包含範例的詳細資訊，請參閱 the section called “Python 應用程式”。

```
cache = SecretCache(
    config = the section called “SecretCacheConfig”,
    client = client
)
```

可用的方法如下：

- get_secret_string

- get_secret_binary

get_secret_string

擷取秘密字串值。

請求語法

```
response = cache.get_secret_string(
    secret_id='string',
    version_stage='string' )
```

參數

- secret_id (字串) – [必需] 秘密的名稱或 ARN。
- version_stage (字串) – 您想要擷取的秘密版本。如需詳細資訊，請參閱秘密版本。預設值為 'AWSCURRENT'。

傳回類型

字串

get_secret_binary

擷取秘密二進位值。

請求語法

```
response = cache.get_secret_binary(
    secret_id='string',
    version_stage='string'
)
```

參數

- secret_id (字串) – [必需] 秘密的名稱或 ARN。
- version_stage (字串) – 您想要擷取的秘密版本。如需詳細資訊，請參閱秘密版本。預設值為 'AWSCURRENT'。

傳回類型

base64-encoded 字串

SecretCacheConfig

the section called “SecretCache” 的快取組態選項，例如最大快取大小和快取秘密的存留時間 (TTL)。

參數

`max_cache_size` (int)

最大快取大小。預設值為 1024 個秘密。

`exception_retry_delay_base` (int)

重試請求前遇到異常之後等待的秒數。預設值為 1。

`exception_retry_growth_factor` (int)pur

用於計算失敗請求重試之間等待時間的增長係數。預設值為 2。

`exception_retry_delay_max` (int)

在失敗請求之間等待的時間上限 (以秒為單位)。預設值為 3600。

`default_version_stage` (str)

要快取的秘密版本。如需詳細資訊，請參閱秘密版本。預設為 'AWSCURRENT'。

`secret_refresh_interval` (int)

重新整理快取秘密資訊之間的等待秒數。預設值為 3600。

`secret_cache_hook` (SecretCacheHook)

SecretCacheHook 摘要類別的實作。預設值為 None。

SecretCacheHook

用於掛接到 the section called “SecretCache”，以對存放在快取中的秘密執行動作的介面。

可用的方法如下：

- put
- get

put

準備要存放在快取中的物件。

請求語法

```
response = hook.put(
    obj='secret_object'
)
```

參數

- obj (物件) – [必需] 包含秘密的秘密或物件。

傳回類型

物件

get

從快取的物件中衍生物件。

請求語法

```
response = hook.get(
    obj='secret_object'
)
```

參數

- obj (物件) – [必需] 包含秘密的秘密或物件。

傳回類型

物件

@InjectSecretString

此裝飾項目需要秘密 ID 字串和 the section called “SecretCache” 作為第一個和第二個引數。裝飾項目傳回秘密字串值。秘密必須包含字串。

```
from aws_secretsmanager_caching import SecretCache
from aws_secretsmanager_caching import InjectKeywordedSecretString,
 InjectSecretString
```

```
cache = SecretCache()

@InjectSecretString ( 'mysecret' ,  cache )
def function_to_be_decorated( arg1,  arg2,  arg3):
```

@InjectKeywordedSecretString

此裝飾項目需要秘密 ID 字串和 the section called “SecretCache” 作為第一個和第二個引數。剩餘的引數將參數從包裝函數映射到秘密中的 JSON 金鑰。秘密必須包含 JSON 結構中的字串。

對於包含此 JSON 的秘密：

```
{
  "username": "saanvi",
  "password": "EXAMPLE-PASSWORD"
}
```

下列範例顯示如何從秘密擷取 username 和 password 的 JSON 值。

```
from aws_secretsmanager_caching import SecretCache
  from aws_secretsmanager_caching import InjectKeywordedSecretString,
 InjectSecretString

  cache = SecretCache()

  @InjectKeywordedSecretString ( secret_id = 'mysecret' ,  cache = cache ,
 func_username = 'username' ,  func_password = 'password' )
  def function_to_be_decorated( func_username,  func_password):
       print( 'Do something with the func_username and func_password parameters')
```

擷取 .NET 應用程式中的 AWS Secrets Manager 秘密

擷取秘密時，您可以使用 Secrets Manager .NET 型快取元件進行快取以供將來使用。擷取快取的秘密比從 Secrets Manager 中擷取要快。由於呼叫 Secrets Manager API 需要花費成本，因此使用快取可以降低成本。如需您可以擷取機密的所有方法，請參閱 擷取秘密。

快取政策是「最近最少使用的 (LRU)」，因此當快取必須丟棄秘密時，其會丟棄最近最少使用的秘密。預設情況下，快取每小時重新整理一次秘密。您可以設定在快取中重新整理機密的頻率，並可以掛接到機密擷取以新增更多功能。

一旦釋放快取參考，快取不會強制執行垃圾回收。快取實作不包括快取失效。快取實作著重於快取本身，而不是強化或著重於安全性。如果您需要額外的安全性，例如加密快取中的項目，請使用提供的介面和抽象方法。

若要使用元件，您必須擁有下列項目：

- .NET Framework 4.6.2 或更新版本，或 .NET Standard 2.0 或更新版本。請參閱 Microsoft 網站上的下載 .NET。
- 適用於 .NET 的 AWS SDK。請參閱 the section called “AWS 開發套件”。

若要下載開放程式碼，請參閱 .NET 快取用戶端 (在 GitHub 上)。

若要使用快取，請首先讓其成為執行個體，然後使用 `GetSecretString` 或 `GetSecretBinary` 擷取。在連續擷取時，快取會傳回秘密的快取複本。

若要取得快取套件

- 執行下列任意一項：
  - 在您的專案目錄中，執行下列 .NET CLI 命令。

    ```
    dotnet add package AWSSDK.SecretsManager.Caching --version 1.0.6
    ```

  - 將以下套件參考新增至您的 `.csproj` 檔案中。

    ```
    <ItemGroup>
        <PackageReference Include="AWSSDK.SecretsManager.Caching" Version="1.0.6" /
    >
    </ItemGroup>
    ```

必要許可：

- `secretsmanager:DescribeSecret`
- `secretsmanager:GetSecretValue`

如需更多詳細資訊，請參閱 許可參考。

參考

- SecretsManagerCache

- SecretCacheConfiguration
- ISecretCacheHook

Example 擷取秘密

下列程式碼範例顯示擷取名為 *MySecret* 之秘密的方式。

```
using Amazon.SecretsManager.Extensions.Caching;

namespace LambdaExample
{
    public class CachingExample
    {
        private const string MySecretName ="MySecret";

        private SecretsManagerCache cache = new SecretsManagerCache();

        public async Task<Response>  FunctionHandlerAsync(string input, ILambdaContext
 context)
        {
            string MySecret = await cache.GetSecretString(MySecretName);

            // Use the secret, return success

        }
    }
}
```

Example 設定存留時間 (TTL) 快取重新整理持續時間

下列程式碼範例說明擷取名為 *MySecret* 的秘密，並將 TTL 快取重新整理持續時間設定為 24 小時的方式。

```
using Amazon.SecretsManager.Extensions.Caching;

namespace LambdaExample
{
    public class CachingExample
    {
        private const string MySecretName = "MySecret";
```

```
        private static SecretCacheConfiguration cacheConfiguration = new
 SecretCacheConfiguration
        {
            CacheItemTTL = 86400000
        };
        private SecretsManagerCache cache = new
 SecretsManagerCache(cacheConfiguration);
        public async Task<Response> FunctionHandlerAsync(string input, ILambdaContext
 context)
        {
            string mySecret = await cache.GetSecretString(MySecretName);

            // Use the secret, return success
        }
    }
}
```

SecretsManagerCache

從 Secrets Manager 請求的秘密記憶體內快取。您使用 the section called “GetSecretString” 或 the section called “GetSecretBinary” 從快取中擷取秘密。您可以透過在建構函式的 the section called “SecretCacheConfiguration” 物件中傳遞，設定快取設定。

如需包含範例的詳細資訊，請參閱 the section called “.NET 應用程式”。

建構函式

```
public SecretsManagerCache()
```

SecretsManagerCache 物件的預設建構函式。

```
public SecretsManagerCache(IAmazonSecretsManager secretsManager)
```

使用以提供的 AmazonSecretsManagerClient 建立的 Secrets Manager 用戶端建構新快取。使用此建構函式可以自訂 Secrets Manager 用戶端，例如使用特定區域或端點。

參數

secretsManager

從中擷取秘密的 AmazonSecretsManagerClient。

```
public SecretsManagerCache(SecretCacheConfiguration config)
```

使用提供的 the section called “SecretCacheConfiguration” 建構新的秘密快取。使用此建構函式設定快取，例如要快取的秘密數量及其重新整理的頻率。

參數

config

包含快取組態資訊的 the section called “SecretCacheConfiguration”。

```
public SecretsManagerCache(IAmazonSecretsManager secretsManager,
SecretCacheConfiguration config)
```

使用以提供的 AmazonSecretsManagerClient 和 the section called “SecretCacheConfiguration” 建立的 Secrets Manager 用戶端建構新快取。使用此建構函式可以自訂 SSecrets Manager 用戶端，例如，使用特定區域或端點並設定快取，例如要快取的秘密數量及其重新整理的頻率。

參數

secretsManager

從中擷取秘密的 AmazonSecretsManagerClient。

config

包含快取組態資訊的 the section called “SecretCacheConfiguration”。

方法

GetSecretString

```
public async Task<String> GetSecretString(String secretId)
```

從 Secrets Manager 中擷取字串秘密。

參數

secretId

要擷取之秘密的 ARN 或名稱。

GetSecretBinary

```
public async Task<byte[]> GetSecretBinary(String secretId)
```

從 Secrets Manager 中擷取二進位秘密。

參數

secretId

要擷取之秘密的 ARN 或名稱。

RefreshNowAsync

```
public async Task<bool> RefreshNowAsync(String secretId)
```

請求 Secrets Manager 的秘密值，並使用任何變更來更新快取。如果沒有現有的快取項目，則請建立一個新的。如果重新整理成功，則會傳回 `true`。

參數

secretId

要擷取之秘密的 ARN 或名稱。

GetCachedSecret

```
public SecretCacheItem GetCachedSecret(string secretId)
```

如果存在於快取中，則傳回指定秘密的快取項目。否則，從 Secrets Manager 中擷取秘密並建立新的快取項目。

參數

secretId

要擷取之秘密的 ARN 或名稱。

SecretCacheConfiguration

the section called “SecretsManagerCache” 的快取組態選項，例如最大快取大小和快取秘密的存留時間 (TTL)。

屬性

CacheItemTTL

```
public uint CacheItemTTL { get; set; }
```

快取項目的 TTL (以毫秒為單位)。預設值為 3600000 毫秒或 1 小時。上限為 4294967295 毫秒，大約為 49.7 天。

MaxCacheSize

```
public ushort MaxCacheSize { get; set; }
```

最大快取大小。預設值為 1024 個秘密。最多 65,535 個。

VersionStage

```
public string VersionStage { get; set; }
```

要快取的秘密版本。如需詳細資訊，請參閱秘密版本。預設為 "AWSCURRENT"。

用戶端

```
public IAmazonSecretsManager Client { get; set; }
```

從中擷取秘密的 AmazonSecretsManagerClient。如果是 null，則快取會執行個體化一個新的用戶端。預設值為 null。

CacheHook

```
public ISecretCacheHook CacheHook { get; set; }
```

the section called “ISecretCacheHook”。

ISecretCacheHook

用於掛接到 the section called “SecretsManagerCache”，以對存放在快取中的秘密執行動作的介面。

方法

放置

```
object Put(object o);
```

準備要存放在快取中的物件。

傳回要存放在快取中的物件。

取得

```
object Get(object cachedObject);
```

從快取的物件中衍生物件。

傳回要從快取傳回的物件

擷取 Go 應用程式中的 AWS Secrets Manager 秘密

擷取秘密時，您可以使用 Secrets Manager Go 型快取元件進行快取以供將來使用。擷取快取的秘密比從 Secrets Manager 中擷取要快。由於呼叫 Secrets Manager API 需要花費成本，因此使用快取可以降低成本。如需您可以擷取機密的所有方法，請參閱 擷取秘密。

快取政策是「最近最少使用的 (LRU)」，因此當快取必須丟棄秘密時，其會丟棄最近最少使用的秘密。預設情況下，快取每小時重新整理一次秘密。您可以設定在快取中重新整理機密的頻率，並可以掛接到機密擷取以新增更多功能。

一旦釋放快取參考，快取不會強制執行垃圾回收。快取實作不包括快取失效。快取實作著重於快取本身，而不是強化或著重於安全性。如果您需要額外的安全性，例如加密快取中的項目，請使用提供的介面和抽象方法。

若要使用元件，您必須擁有下列項目：

- 適用於 Go 的 AWS SDK。請參閱 the section called “AWS 開發套件”。

若要下載開放程式碼，請參閱 Secrets Manager Go 快取用戶端 (在 GitHub 上)。

若要設定 Go 開發環境，請參閱 Go 程式設計語言網站上的 Golang 入門。

必要許可：

- `secretsmanager:DescribeSecret`
- `secretsmanager:GetSecretValue`

如需更多詳細資訊，請參閱 許可參考。

參考

- 輸入 Cache
- 輸入 CacheConfig
- 輸入 CacheHook

Example 擷取秘密

下列程式碼範例顯示擷取秘密的 Lambda 函數。

```
package main

import (
  "github.com/aws/aws-lambda-go/lambda"
  "github.com/aws/aws-secretsmanager-caching-go/secretcache"
)

var (
  secretCache, _ = secretcache.New()
)

func HandleRequest(secretId string) string {
  result, _ := secretCache.GetSecretString(secretId)

  // Use the secret, return success
}

 func main() {
  lambda. Start( HandleRequest)
}
```

輸入 Cache

從 Secrets Manager 請求的秘密記憶體內快取。您使用 the section called “GetSecretString” 或 the section called “GetSecretBinary” 從快取中擷取秘密。

下列範例顯示如何設定快取設定。

```
// Create a custom secretsmanager client
client := getCustomClient()

// Create a custom CacheConfig struct
config := secretcache. CacheConfig{
```

```
    MaxCacheSize:  secretcache.DefaultMaxCacheSize + 10,
    VersionStage:  secretcache.DefaultVersionStage,
    CacheItemTTL:  secretcache.DefaultCacheItemTTL,
}

// Instantiate the cache
cache, _ := secretcache.New(
    func( c *secretcache.Cache) {  c. CacheConfig = config },
    func( c *secretcache.Cache) {  c. Client = client },
)
```

如需包含範例的詳細資訊，請參閱 the section called “Go 應用程式”。

方法

全新

```
func New(optFns ...func(*Cache)) (*Cache, error)
```

使用功能選項建構新秘密快取，否則使用預設值。從新的工作階段初始化 SecretsManager 用戶端。將 CacheConfig 初始化為預設值。使用預設的最大大小起始 LRU 快取。

GetSecretString

```
func (c *Cache) GetSecretString(secretId string) (string, error)
```

GetSecretString 從快取中取得秘密字串值，用於指定秘密 ID。如果操作失敗，則傳回秘密字串和錯誤。

GetSecretStringWithStage

```
func (c *Cache) GetSecretStringWithStage(secretId string, versionStage
string) (string, error)
```

GetSecretStringWithStage 從快取中取得秘密字串值，用於指定秘密 ID 和版本階段。如果操作失敗，則傳回秘密字串和錯誤。

GetSecretBinary

```
func (c *Cache) GetSecretBinary(secretId string) ([]byte, error) {
```

GetSecretBinary 從快取中取得秘密二進位值，用於指定秘密 ID。如果操作失敗，則傳回秘密二進位和錯誤。

GetSecretBinaryWithStage

`func (c *Cache) GetSecretBinaryWithStage(secretId string, versionStage string) ([]byte, error)`

GetSecretBinaryWithStage 從快取中取得秘密二進位值，用於指定秘密 ID 和版本階段。如果操作失敗，則傳回秘密二進位和錯誤。

輸入 CacheConfig

快取的快取組態選項，例如最大快取大小、預設版本階段和快取秘密的存留時間 (TTL)。

```
type CacheConfig struct {

    // The maximum cache size. The default is 1024 secrets.
    MaxCacheSize int

    // The TTL of a cache item in nanoseconds. The default is
    // 3.6e10^12 ns or 1 hour.
    CacheItemTTL int64

    // The version of secrets that you want to cache. The default
    // is "AWSCURRENT".
    VersionStage string

    // Used to hook in-memory cache updates.
    Hook CacheHook
}
```

輸入 CacheHook

用於掛接到快取，以對存放在快取中的秘密執行動作的介面。

方法

放置

`Put(data interface{}) interface{}`

準備要存放在快取中的物件。

取得

`Get(data interface{}) interface{}`

從快取的物件中衍生物件。

使用 AWS Batch 中的 AWS Secrets Manager 秘密

AWS Batch 可協助您在 AWS 雲端 上執行批次運算工作負載。AWS Batch 讓您可在 AWS Secrets Manager 秘密中存放您的敏感資料，然後由您的任務定義加以參考，藉此將敏感資料注入您的任務。如需詳細資訊，請參閱使用 Secrets Manager 指定敏感資料。

在 AWS CloudFormation 資源中擷取 AWS Secrets Manager 秘密

透過 AWS CloudFormation，您可以擷取秘密以在另一個 AWS CloudFormation 資源中使用。常見的案例是首先使用 Secrets Manager 產生的密碼建立秘密，然後從秘密中擷取使用者名稱和密碼，以用作新資料庫的憑證。如需使用 AWS CloudFormation 建立秘密的資訊，請參閱AWS CloudFormation。

若要擷取 AWS CloudFormation 範本中的秘密，您可以使用動態參考。當您建立堆疊時，動態參考會將秘密值提取到 AWS CloudFormation 資源中，因此您不必對秘密資訊執行硬式編碼。相對地，您就必須依名稱或 ARN 來參考秘密。您可以在任何資源屬性中將動態參考用於機密。您不能在資源中繼資料中將動態參考用於機密 (例如 `AWS::CloudFormation::Init`)，因為這會使機密值在主控台中可見。

秘密的動態參考具有下列模式：

```
{{resolve:secretsmanager:secret-id:SecretString:json-key:version-stage:version-id}}
```

secret-id

秘密的名稱或 ARN。若要存取您的 AWS 帳戶中的秘密，您可以使用秘密名稱。若要存取不同 AWS 帳戶中的秘密，請使用秘密的 ARN。

json-key (選用)

您要擷取值的鍵值組的索引鍵名稱。如果您不指定 `json-key`，AWS CloudFormation 會擷取整個秘密文字。此區段可能不可包含冒號字元 (`:`)。

version-stage (選用)

要使用的秘密版本。Secrets Manager 在輪換程序期間使用預備標籤來追蹤不同版本。如果您使用 `version-stage`，請不要指定 `version-id`。如果您未指定 `version-stage` 或 `version-id`，則預設為 `AWSCURRENT` 版本。此區段可能不可包含冒號字元 (`:`)。

version-id (選用)

您要使用的秘密版本的唯一識別碼。如果您指定 `version-id`，則請不要指定 `version-stage`。如果您未指定 `version-stage` 或 `version-id`，則預設為 `AWSCURRENT` 版本。此區段可能不可包含冒號字元 (`:`)。

如需詳細資訊，請參閱使用動態參考指定 Secrets Manager 秘密。

> **Note**
>
> 請不要使用反斜線 (`\`) 建立動態參考作為最終值。AWS CloudFormation 無法解析導致資源故障的這些參考。

在 Amazon Elastic Container Service 中使用 AWS Secrets Manager 秘密

Amazon Elastic Container Service (Amazon ECS) 為全受管容器協同運作服務，可讓您輕鬆部署、管理和擴展容器化應用程式。您可以參考 Secrets Manager 秘密，將敏感資料插入容器中。如需詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的下列頁面：

- 教學課程：使用 Secrets Manager 秘密指定敏感資料
- 透過應用程式以程式設計方式擷取秘密
- 透過環境變數擷取秘密
- 擷取記錄組態的秘密

在 Amazon Elastic Kubernetes Service 中使用 AWS Secrets Manager 秘密

若要將來自機 Secrets Manager 的密碼顯示為裝載在 Amazon EKS 網繭中的檔案，您可以使用 Kubernetes 機 AWS 密存放區 CSI 驅動程式的機密和組態提供者 (ASCP)。ASCP 與運行 Amazon EC2 節點組的亞馬遜彈性庫伯尼特斯服務（亞馬遜 EKS）1.17 + 一起使用。 AWS Fargate 不支援節點群組。透過 ASCP，您可以在 Secrets Manager 中存放和管理您的秘密，然後透過在 Amazon EKS 上執行的工作負載擷取這些秘密。如果秘密包含 JSON 格式的多個金鑰/值對，您可以選擇要在 Amazon EKS 中掛載哪些金鑰/值對。ASCP 使用 JMESPath 語法來查詢秘密中的金鑰/值對。ASCP 也可與 Parameter Store 參數搭配使用。

您可以使用 IAM 角色和政策來限制對叢集中特定 Amazon EKS Pod 的機密存取。

若要描述要在 Amazon EKS Pod 中建立哪些檔案以及要放入哪些機密，您可以建立 the section called "SecretProviderClass" YAML 檔案。所以 SecretProviderClass 必須和其參考的 Amazon EKS Pod 位在同一命名空間。

如果您使用私有 Amazon EKS 叢集，則請確保叢集所在的 VPC 具有 Secrets Manager 端點。Secrets Store CSI Driver 使用端點呼叫 Secrets Manager。如需有關在 VPC 中建立端點的資訊，請參閱 VPC 端點。

如果使用 Secrets Manager 自動輪換秘密，您也可以使用 Secrets Store CSI Driver 輪換調解器功能，確保從 Secrets Manager 擷取最新的秘密。如需詳細資訊，請參閱自動輪換已掛載的內容和已同步的 Kubernetes 秘密。

如需如何使用 ASCP 的教學課程，請參閱 the section called "教學課程"。

安裝 ASCP

您可以在 secrets-store-csi-provider-aws 儲存庫 GitHub 中使用 ASCP。儲存庫還包含用於建立和掛載秘密的範例 YAML 檔案。

若要安裝 ASCP

- 若要使用 Helm 來安裝 Secrets Store CSI Driver 和 ASCP，請使用下列命令。為確保儲存庫指向最新圖表，請使用 helm repo update.

```
helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-store-csi-driver/charts
helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver

helm repo add aws-secrets-manager https://aws.github.io/secrets-store-csi-driver-provider-aws
helm install -n kube-system secrets-provider-aws aws-secrets-manager/secrets-store-csi-driver-provider-aws
```

  或者，若要使用部署目錄中的 YAML 檔案來安裝，請使用下列命令。

```
helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-store-csi-driver/charts
helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver
```

```
kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-
provider-aws/main/deployment/aws-provider-installer.yaml
```

步驟 1：設定存取控制

若要在 Secrets Manager 中授予 Amazon EKS Pod 存取機密的權限，您首先要建立一個許可政策來授予對 Pod 需要存取的機密的 `secretsmanager:GetSecretValue` 和 `secretsmanager:DescribeSecret` 許可。如需範例政策，請參閱 許可政策範例。

然後建立服務帳戶的 IAM 角色並將政策連接到該角色。如需詳細資訊，請參閱服務帳戶的 IAM 角色。

ASCP 會擷取 Pod 身分並將其交換為 IAM 角色。ASCP 會假設 Pod 的 IAM 角色，讓其存取您授權的秘密。除非您也將其與 IAM 角色建立關聯，否則其他容器無法存取秘密。

如果您使用私有 Amazon EKS 叢集，請確保叢集所在的 VPC 具有 AWS STS 端點。如需有關建立端點的資訊，請參閱AWS Identity and Access Management 使用指南中的介面 VPC 端點。

步驟 2：識別要掛載的機密

若要確定 ASCP 在 Amazon EKS 中掛載哪些機密作為檔案系統上的檔案，請建立 `SecretProviderClass` YAML 檔案。`SecretProviderClass` YAML 會列出要掛載的機密，以及將其掛載的檔案名稱。所以 `SecretProviderClass` 必須和其參考的 Amazon EKS Pod 位在同一命名空間。

下列範例顯示如何使用 `SecretProviderClass` 來描述您要掛載的機密，以及如何命名在 Amazon EKS Pod 中掛載的檔案。如需詳細資訊，請參閱 the section called “`SecretProviderClass`”。

範例：

- 範例：依名稱或 ARN 掛載機密
- 範例：透過機密掛載金鑰/值對
- 範例：定義多區域機密的容錯移轉區域
- 範例：選擇要掛載的容錯移轉機密

範例：依名稱或 ARN 掛載機密

下列範例顯示在 Amazon EKS 中掛載三個檔案的 `SecretProviderClass`：

1. 由完整 ARN 指定的秘密。

2. 依名稱指定的秘密。
3. 秘密的特定版本。

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
    objects: |
        - objectName: "arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret2-
d4e5f6"
        - objectName: "MySecret3"
          objectType: "secretsmanager"
        - objectName: "MySecret4"
          objectType: "secretsmanager"
          objectVersionLabel: "AWSCURRENT"
```

範例：透過機密掛載金鑰/值對

下列範例顯示在 Amazon EKS 中掛載三個檔案的 SecretProviderClass：

1. 由完整 ARN 指定的秘密。
2. 此 username 金鑰/值對來自相同的秘密。
3. 此 password 金鑰/值對來自相同的秘密。

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret-
a1b2c3"
        jmesPath:
            - path: username
              objectAlias: dbusername
```

```
            - path: password
              objectAlias: dbpassword
```

範例：定義多區域機密的容錯移轉區域

為了在連線中斷期間提供可用性或用於災難復原組態，ASCP 支援自動化容錯移轉功能以從次要區域擷取機密。

下列範例顯示可擷取複寫至多個區域之機密的 SecretProviderClass。在此範例中，ASCP 會嘗試同時從 us-east-1 和 us-east-2 擷取機密。如果任一區域傳回 4xx 錯誤 (例如身分驗證問題)，則 ASCP 不會掛載任一機密。如果已成功從 us-east-1 擷取機密，則 ASCP 會掛載該機密值。如果未成功從 us-east-1 擷取機密，但已成功從 us-east-2 擷取機密，則 ASCP 會掛載該機密值。

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "MySecret"
```

範例：選擇要掛載的容錯移轉機密

下列範例顯示用於指定在容錯移轉時要裝載之機密的 SecretProviderClass。容錯移轉機密不是複本。在此範例中，ASCP 會嘗試擷取 objectName 指定的兩個機密。如果任一項傳回 4xx 錯誤 (例如身分驗證問題)，則 ASCP 不會掛載任一機密。如果已成功從 us-east-1 擷取機密，則 ASCP 會掛載該機密值。如果未成功從 us-east-1 擷取機密，但已成功從 us-east-2 擷取機密，則 ASCP 會掛載該機密值。Amazon EKS 中掛載的檔案命名為 MyMountedSecret。

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
```

```
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "arn:aws:secretsmanager:us-east-1:111122223333:secret:MySecret-
a1b2c3"
        objectAlias: "MyMountedSecret"
        failoverObject:
          - objectName: "arn:aws:secretsmanager:us-
east-2:111122223333:secret:MyFailoverSecret-d4e5f6"
```

疑難排解

您可以透過描述 Pod 部署來檢視大多數錯誤。

若要查看容器的錯誤訊息

1. 使用下列命令取得 Pod 名稱清單。如果不使用預設命名空間，請使用 `-n <NAMESPACE>`。

   ```
   kubectl get pods
   ```

2. 若要描述 Pod，請在下列命令中，針對 *<PODID>* 使用您在上一個步驟中從 Pod 找到的 Pod ID。如果不使用預設命名空間，請使用 `-n <NAMESPACE>`。

   ```
   kubectl describe pod/<PODID>
   ```

若要查看 ASCP 的錯誤

- 若要在提供者記錄檔中尋找更多資訊，請在下列命令<PODID>中使用 csi-secrets-store-provider-aws pod 的識別碼。

  ```
  kubectl -n kube-system get pods
  kubectl -n kube-system logs pod/<PODID>
  ```

教學課程：在 Amazon EKS 網繭中建立和掛接機 AWS Secrets Manager 密

在本教學課程中，您會在 Secrets Manager 中建立範例秘密，然後將秘密掛載到 Amazon EKS Pod 中並進行部署。

開始之前，請先安裝 ASCP：the section called “安裝 ASCP”。

若要建立和掛載秘密

1. 將集群的名稱 AWS 區域 和名稱設置為 shell 變量，以便您可以在 bash 命令中使用它們。對於<REGION>，請輸入您 AWS 區域 的 Amazon EKS 叢集執行位置。針對 *<CLUSTERNAME>*，請輸入您叢集的名稱。

```
REGION=<REGION>
CLUSTERNAME=<CLUSTERNAME>
```

2. 建立測試秘密。如需詳細資訊，請參閱 建立和管理秘密。

```
aws --region "$REGION" secretsmanager  create-secret --name MySecret --secret-
string '{"username":"lijuan", "password":"hunter2"}'
```

3. 建立 Pod 的資源政策，其會限制對您在上個步驟建立的秘密的存取。針對 *<SECRETARN>*，請使用秘密的 ARN。將政策 ARN 儲存在 shell 變數中。

```
POLICY_ARN=$(aws --region "$REGION" --query Policy.Arn --output text iam create-
policy --policy-name nginx-deployment-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [ {
        "Effect": "Allow",
        "Action": ["secretsmanager:GetSecretValue",
 "secretsmanager:DescribeSecret"],
        "Resource": ["<SECRETARN>"]
    } ]
}')
```

4. 如果尚未建立，請為叢集建立 IAM OIDC 提供者。如需詳細資訊，請參閱為叢集建立 IAM OIDC 提供者。

```
eksctl utils associate-iam-oidc-provider --region="$REGION" --
cluster="$CLUSTERNAME" --approve # Only run this once
```

5. 建立 Pod 使用的服務帳戶，並將您在步驟 3 中建立的資源政策與該服務帳戶關聯起來。在本教學課程中，您可以使用服務帳戶名稱nginx-deployment-sa。如需詳細資訊，請參閱為服務帳戶建立 IAM 角色。

```
eksctl create iamserviceaccount --name nginx-deployment-sa --region="$REGION" --
cluster "$CLUSTERNAME" --attach-policy-arn "$POLICY_ARN" --approve --override-
existing-serviceaccounts
```

6. 建立 SecretProviderClass 來指定要在 Pod 中掛載的秘密。下列指令會ExampleSecretProviderClass.yaml在 ASCP GitHub repo 範例目錄中使用來掛載您在步驟 2 中建立的密碼。如需編寫自有 SecretProviderClass 的詳細資訊，請參閱 the section called “SecretProviderClass”。

```
kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-provider-aws/main/examples/ExampleSecretProviderClass.yaml
```

7. 部署您的 Pod。下列命令會ExampleDeployment.yaml在 ASCP GitHub repo 範例目錄中使用，在網繭/mnt/secrets-store中掛載密碼。

```
kubectl apply -f https://raw.githubusercontent.com/aws/secrets-store-csi-driver-provider-aws/main/examples/ExampleDeployment.yaml
```

8. 若要確認秘密是否已正確掛載，請使用下列命令並確認您的秘密值出現。

```
kubectl exec -it $(kubectl get pods | awk '/nginx-deployment/{print $1}' | head -1)
 cat /mnt/secrets-store/MySecret; echo
```

秘密值隨即會顯示。

```
{"username":"lijuan", "password":"hunter2"}
```

SecretProviderClass

您可以使用 YAML 來描述要使用 ASCP 在 Amazon EKS 中裝載的機密。如需範例，請參閱 識別要掛載的機密。

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: <NAME>
spec:
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    objects:
```

欄位 parameters 包含掛載請求的詳細資訊：

region

(選擇性) 密碼 AWS 區域 的。如果不使用此欄位，ASCP 會從節點上的註釋尋找區域。此查閱會增加掛載請求的額外負荷，因此建議您為使用大量 Pod 的叢集提供區域。

如果您也指定 failoverRegion，則 ASCP 會嘗試從這兩個區域擷取機密。如果任一區域傳回 4xx 錯誤 (例如身分驗證問題)，則 ASCP 不會掛載任一機密。如果已成功從 region 擷取機密，則 ASCP 會掛載該機密值。如果未成功從 region 擷取機密，但已成功從 failoverRegion 擷取機密，則 ASCP 會掛載該機密值。

failoverRegion

(選用) 如果您包含此欄位，則 ASCP 會嘗試從 region 中定義的區域和此欄位擷取機密。如果任一區域傳回 4xx 錯誤 (例如身分驗證問題)，則 ASCP 不會掛載任一機密。如果已成功從 region 擷取機密，則 ASCP 會掛載該機密值。如果未成功從 region 擷取機密，但已成功從 failoverRegion 擷取機密，則 ASCP 會掛載該機密值。如需如何使用此欄位的範例，請參閱 定義多區域機密的容錯移轉區域。

pathTranslation

(選用) 如果 Amazon EKS 中的檔案名稱將包含路徑分隔符號字元，例如 Linux 上的斜線 (/)，則要使用的單一替代字元。ASCP 無法建立包含路徑分隔符號字元的掛載檔案。相反地，ASCP 會將路徑分隔符號字元取代為其他字元。如果不使用此欄位，則取代字元為底線 (_)，例如 My/Path/Secret 掛載為 My_Path_Secret。

若要避免發生字元取代的情況，請輸入字串 False。

objects

包含待掛載秘密的 YAML 宣告的字串。建議使用 YAML 多行字串或分隔號 (|) 字元。

objectName

秘密的名稱或完整 ARN。如果使用 ARN，您可以省略 objectType。這欄位會成為 Amazon EKS Pod 中機密的檔案名稱，除非您指定 objectAlias。如果您使用 ARN，則 ARN 中的區域必須與欄位 region 相符。如果您包含 failoverRegion，則此欄位代表主要 objectName。

objectType

如果未針對 objectName 使用 Secrets Manager ARN，其則為必要欄位。可以是 secretsmanager 或 ssmparameter。

objectAlias

(選用) Amazon EKS Pod 中秘密的檔案名稱。如果您未指定此欄位，objectName 會顯示為檔案名稱。

objectVersion

(選用) 秘密的版本 ID。不建議，因為每次更新機密時都必須更新版本 ID。依預設，會使用最新版本。如果您包含 failoverRegion，則此欄位代表主要 objectVersion。

objectVersionLabel

(選用) 版本的別名。預設為最新版本 AWSCURRENT。如需詳細資訊，請參閱 the section called “版本”。如果您包含 failoverRegion，則此欄位代表主要 objectVersionLabel。

jmesPath

(選用) 要掛載在 Amazon EKS 中之檔案的秘密金鑰映射。若要使用此欄位，您的秘密值必須是 JSON 格式。如果使用此欄位，您必須包含 path 和 objectAlias。

路徑

來自秘密值 JSON 中金鑰/值對的金鑰。如果欄位包含連字號，請使用單引號將其逸出，例如：path: '"hyphenated-path"'。

objectAlias

要掛載在 Amazon EKS Pod 中的檔案名稱。如果欄位包含連字號，請使用單引號將其逸出，例如：objectAlias: '"hyphenated-alias"'。

failoverObject

(選用) 如果您指定此欄位，則 ASCP 會嘗試擷取在主要 objectName 中指定的機密和在 failoverObject objectName 子欄位中指定的機密。如果任一項傳回 4xx 錯誤 (例如身分驗證問題)，則 ASCP 不會掛載任一機密。如果已成功從主要 objectName 擷取機密，則 ASCP 會掛載該機密值。如果未成功從主要 objectName 擷取機密，但已成功從容錯移轉 objectName 擷取機密，則 ASCP 會掛載該機密值。如果包含此欄位，則您必須包含欄位 objectAlias。如需如何使用此欄位的範例，請參閱 選擇要掛載的容錯移轉機密。

當容錯移轉機密不是複本時，您通常會使用此欄位。如需如何指定複本的範例，請參閱 定義多區域機密的容錯移轉區域。

objectName

容錯移轉機密的名稱或完整 ARN。如果您使用 ARN，則 ARN 中的區域必須與欄位 failoverRegion 相符。

objectVersion

(選用) 秘密的版本 ID。必須與主要 `objectVersion` 相符。不建議，因為每次更新機密時都必須更新版本 ID。依預設，會使用最新版本。

objectVersionLabel

(選用) 版本的別名。預設為最新版本 AWSCURRENT。如需更多詳細資訊，請參閱 the section called “版本”。

在 GitHub 工作中使用 AWS Secrets Manager 秘密

若要在 GitHub 工作中使用密碼，您可以使用 GitHub 動作擷取密碼， AWS Secrets Manager 並將其新增為工作 GitHub 流程中的遮罩環境變數。如需有關 GitHub 動作的詳細資訊，請參閱瞭解GitHub 文件中的 GitHub 動作。

當您在 GitHub 環境中新增密碼時， GitHub 工作中的所有其他步驟都可以使用該密碼。請遵循動 GitHub 作的安全性強化中的指引，以協助防止您環境中的密碼遭到濫用。

您可以將秘密值的整個字串設定為環境變數值，或者如果字串為 JSON，您可以剖析 JSON，為每個 JSON 索引鍵值組設定個別環境變數。如果秘密值是二進位，此動作會將其轉換為字串。

若要檢視以您秘密建立的環境變數，請開啟偵錯記錄。如需詳細資訊，請參閱在GitHub 文件中啟用偵錯記錄。

若要使用密碼建立的環境變數，請參閱GitHub 文件中的環境變數。

必要條件

若要使用此動作，您首先需要設定 AWS 認證，並使用`configure-aws-credentials`步驟 AWS 區域 在您的 GitHub 環境中設定。依照設定 AWS 認證動作中的指示 GitHub 執行，以直接使用 GitHub OIDC 提供者承擔角色的動作。這可讓您使用短期憑證，避免將其他存取金鑰儲存在 Secrets Manager 之外。

此動作擔任的 IAM 角色必須擁有下列許可：

- 對您要擷取的秘密有 `GetSecretValue`。
- 對所有秘密有 `ListSecrets`。
- (選擇性) `Decrypt` KMS key 如果密碼使用 客戶受管金鑰.

如需詳細資訊，請參閱 身分驗證與存取控制。

用量

若要使用此動作，請新增步驟至使用下列語法的工作流程。

```
- name: Step name
  uses: aws-actions/aws-secretsmanager-get-secrets@v2
  with:
    secret-ids: |
      secretId1
      ENV_VAR_NAME, secretId2
    parse-json-secrets: (Optional) true|false
```

參數

`secret-ids`

秘密 ARN、名稱和名稱字首。

此步驟預設會以秘密名稱建立每個環境變數名稱，環境變數名稱已轉換為只包含大寫字母、數字和底線，因此不會以數字開頭。

若要設定環境變數名稱，請在秘密 ID 前輸入名稱，然後加上英文逗號。例如，`ENV_VAR_1, secretId` 會以秘密 `secretId` 建立名為 ENV_VAR_1 的環境變數。環境變數名稱可包含大寫字母、數字和底線。

若要使用字首，請輸入至少三個字元，然後加上星號。例如，`dev*` 會符合名稱以 dev 開頭的所有秘密。可擷取的相符秘密數上限為 100。如果您設定變數名稱，且字首與多個秘密相符，則動作會失敗。

`parse-json-secrets`

(選用) 此動作預設會將環境變數值設定為秘密值的整個 JSON 字串。將 `parse-json-secrets` 設定為 `true`，為 JSON 中的每個索引鍵/值組建立環境變數。

請注意，如果 JSON 使用區分大小寫的索引鍵 (例如 "name" 和 "Name")，動作會有重複名稱衝突。在此情況下，請將 `parse-json-secrets` 設定為 `false`，並分別剖析 JSON 秘密值。

環境變數命名

動作所建立的環境變數的命名方式與其來源的密碼相同。環境變數的命名需求比機密更嚴格，因此動作會轉換密碼名稱以符合這些需求。例如，此動作會將小寫字母轉換為大寫字母。例如，如果剖析機密的 JSON，則環境變數名稱會同時包含秘密名稱和 JSON 金鑰名稱MYSECRET_KEYNAME。

如果兩個環境變數以相同的名稱結束，則動作會失敗。在這種情況下，您必須將要用於環境變數的名稱指定為別名。

名稱可能發生衝突的例子：

- 名為 "MySecret" 的密碼和名為「mysecret」的機密都會成為名為「MYSECRET」的環境變數。
- 一個名為「Secret _KEYNAME」的密鑰和一個名為「秘密」的 JSON 解析密鑰，並帶有一個名為「密鑰名」的密鑰將都成為名為「SECRET_KEYNAME」的環境變量。

您可以透過指定別名來設定環境變數名稱，如下列範例所示，這會建立一個名為的變數ENV_VAR_NAME。

```
secret-ids: |
  ENV_VAR_NAME, secretId2
```

空白別名

- 如果您設定parse-json-secrets: true並輸入空白別名，後面接著逗號，然後輸入秘密 ID，動作會將環境變數命名為與剖析的 JSON 金鑰相同。變數名稱不包含密碼名稱。

  如果密碼不包含有效的 JSON，則動作會建立一個環境變數，並將其命名為與密碼名稱相同。
- 如果您設定parse-json-secrets: false並輸入空白別名，後面接著逗號和秘密 ID，則動作會將環境變數命名為如同您未指定別名一樣。

下面的例子顯示了一個空白的別名。

```
,secret2
```

範例

Example 1 依名稱和 ARN 取得秘密

以下範例會為依名稱和 ARN 識別的秘密，建立環境變數。

```
- name: Get secrets by name and by ARN
  uses: aws-actions/aws-secretsmanager-get-secrets@v2
  with:
    secret-ids: |
      exampleSecretName
      arn:aws:secretsmanager:us-east-2:123456789012:secret:test1-a1b2c3
      0/test/secret
      /prod/example/secret
      SECRET_ALIAS_1,test/secret
      SECRET_ALIAS_2,arn:aws:secretsmanager:us-east-2:123456789012:secret:test2-a1b2c3
      ,secret2
```

建立的環境變數：

```
EXAMPLESECRETNAME: secretValue1
TEST1: secretValue2
_0_TEST_SECRET: secretValue3
_PROD_EXAMPLE_SECRET: secretValue4
SECRET_ALIAS_1: secretValue5
SECRET_ALIAS_2: secretValue6
SECRET2: secretValue7
```

Example 2 取得以字首開頭的所有秘密

以下範例會為名稱以 *beta* 開頭的所有秘密，建立環境變數。

```
- name: Get Secret Names by Prefix
  uses: 2
  with:
    secret-ids: |
      beta*    # Retrieves all secrets that start with 'beta'
```

建立的環境變數：

```
BETASECRETNAME: secretValue1
BETATEST: secretValue2
BETA_NEWSECRET: secretValue3
```

Example 3 剖析秘密中的 JSON

以下範例會剖析秘密中的 JSON，藉此建立環境變數。

```
- name: Get Secrets by Name and by ARN
  uses: aws-actions/aws-secretsmanager-get-secrets@v2
  with:
    secret-ids: |
      test/secret
      ,secret2
    parse-json-secrets: true
```

秘密 `test/secret` 具有以下秘密值。

```
{
  "api_user": "user",
  "api_key": "key",
  "config": {
    "active": "true"
  }
}
```

秘密 `secret2` 具有以下秘密值。

```
{
  "myusername": "alejandro_rosalez",
  "mypassword": "EXAMPLE_PASSWORD"
}
```

建立的環境變數：

```
TEST_SECRET_API_USER: "user"
TEST_SECRET_API_KEY: "key"
TEST_SECRET_CONFIG_ACTIVE: "true"
MYUSERNAME: "alejandro_rosalez"
MYPASSWORD: "EXAMPLE_PASSWORD"
```

使用 AWS IoT Greengrass 中的 AWS Secrets Manager 秘密

AWS IoT Greengrass 是將雲端功能延伸至本機裝置的軟體。這能讓裝置收集與分析更接近資訊來源的資料、自主回應本機裝置，在本機網路上安全地互相通訊。

AWS IoT Greengrass 可讓您從 Greengrass 裝置向服務和應用程式進行身分驗證，而無需將密碼、字符或其他秘密寫入程式碼。您可以使用 AWS Secrets Manager 在雲端中安全地存放及管理秘密。AWS

IoT Greengrass 可將 Secrets Manager 擴展到 Greengrass 核心裝置，因此您的連接器和 Lambda 函數可以使用本機秘密與服務和應用程式互動。

若要將私密整合到 Greengrass 群組，您可以建立一個參考 Secrets Manager 私密的群組資源。此私密資源會使用相關聯的 ARN 來參考雲端私密。若要瞭解如何建立、管理及使用秘密資源，請參閱 AWS IoT 開發人員指南中的使用秘密資源。

若要將秘密部署至 AWS IoT Greengrass 核心，請參閱將秘密部署至 AWS IoT Greengrass 核心。

在 AWS Lambda 函數中使用 AWS Secrets Manager 密碼

您可以使用 AWS 參數和機密 Lambda 擴充功能擷取和快取 Lambda 函數中的 AWS Secrets Manager 密碼，而無需使用 SDK。擷取快取的秘密比從 Secrets Manager 中擷取要快。由於呼叫 Secrets Manager API 需要花費成本，因此使用快取可以降低成本。延伸模組可以擷取 Secrets Manager 秘密和 Parameter Store 參數。如需 Parameter Store 的資訊，請參閱《AWS Systems Manager 使用者指南》中的 Parameter Store integration with Lambda extensions (Parameter Store 與 Lambda 延伸模組整合)。

Lambda 延伸模組是新增至 Lambda 函數功能的隨附程序。如需詳細資訊，請參閱《Lambda 開發人員指南》中的 Lambda 延伸模組。如需在容器映像中使用延伸模組的詳細資訊，請參閱《在容器映像中使用 Lambda 層和延伸模組》。Lambda 使用 Amazon CloudWatch 日誌記錄有關擴展程序的執行信息以及該函數。根據預設，擴充功能會將最少量的資訊記錄到 CloudWatch。若要記錄更多詳細資訊，請將環境變數 `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` 設定為 `debug`。

為了提供參數和秘密的記憶體快取，擴充功能會將本機 HTTP 端點 (本機主機連接埠 2773) 開放給 Lambda 環境。您可以設定環境變數 `PARAMETERS_SECRETS_EXTENSION_HTTP_PORT` 來設定連接埠。

Lambda 會將與函數所要求並行層級相符的另外執行個體具現化。每個執行個體都彼此隔離，並維護自己組態資料的本機快取。如需 Lambda 執行個體和並行的詳細資訊，請參閱《Lambda 開發人員指南》中的管理 Lambda 函數的並行。

若要為 ARM 新增延伸模組，您必須為 Lambda 函數使用 `arm64` 架構。如需詳細資訊，請參閱《Lambda 開發人員指南》中的 Lambda 指令集架構。延伸模組在下列區域支援 ARM：亞太區域 (孟買)、美國東部 (俄亥俄)、歐洲 (愛爾蘭)、歐洲 (法蘭克福)、歐洲 (蘇黎世)、美國東部 (維吉尼亞北部)、歐洲 (倫敦)、歐洲 (西班牙)、亞太區域 (東京)、美國西部 (奧勒岡)、亞太區域 (新加坡)、亞太區域 (海德拉巴) 及亞太區域 (雪梨)。

擴充功能會使用用 AWS 戶端。如需有關設定 AWS 用戶端的資訊，請參閱 AWS SDK 和工具參考指南中的設定參考資料。如果您的 Lambda 函數在 VPC 中執行，則需要建立 VPC 端點，以便擴充功能可以呼叫 Secrets Manager。如需詳細資訊，請參閱 VPC 端點。

必要許可：

- Lambda 執行角色必須具有密碼的`secretsmanager:GetSecretValue`權限。
- 如果使用客戶受管金鑰而非使用客戶管理的金鑰加密密碼 AWS 受管金鑰 `aws/secretsmanager`，則執行角色也需要 KMS 金鑰的`kms:Decrypt`權限。

若要使用 AWS 參數和秘密 Lambda 擴充

1. 請執行下列任一操作，新增層至您的函數中：
   - 請在以下位置開啟 AWS Lambda 主控台。 https://console.aws.amazon.com/lambda/
     a. 選擇您的函數，選擇 Layers (層)，然後選擇 Add a layer (新增層)。
     b. 在新增層頁面上，針對 AWS 層，選擇 AWS 參數和秘密 Lambda 延伸，然後選擇新增。
   - 請使用下列 AWS CLI 指令搭配您所在地區的適當 ARN。如需 ARN 的清單，請參閱《AWS Systems Manager 使用者指南》中的AWS 參數和秘密 Lambda 延伸 ARN。

```
aws lambda update-function-configuration \
    --function-name my-function \
    --layers LayerARN
```

2. 將許可授予 Lambda 執行角色，以便能夠存取秘密：
   - 秘密的 `secretsmanager:GetSecretValue` 許可。請參閱the section called “範例：擷取每個秘密值的許可”。
   - (選擇性) 如果使用客戶管理的金鑰而非使用客戶管理金鑰加密密碼 AWS 受管金鑰 `aws/secretsmanager`，則執行角色也需要 KMS 金鑰的`kms:Decrypt`權限。
   - 您可以將屬性型存取控制 (ABAC) 與 Lambda 角色搭配使用，以便更精細地存取帳戶中的秘密。如需詳細資訊，請參閱 the section called “範例：使用標籤控制對秘密的存取” 及 the section called “範例：使用符合秘密標籤的標籤限制對身分的存取”。
3. 使用 Lambda 環境變數設定快取。
4. 若要從延伸模組快取擷取秘密，您必須先將 `X-AWS-Parameters-Secrets-Token` 新增至請求標頭。將字符設定為 `AWS_SESSION_TOKEN`，這會由 Lambda 針對所有運行中的函數提供。使用此標頭表示呼叫者位於 Lambda 環境中。

以下 Python 範例說明如何新增標頭。

```
import os
headers = {"X-Aws-Parameters-Secrets-Token": os.environ.get('AWS_SESSION_TOKEN')}
```

5. 若要擷取 Lambda 函數中的秘密，請使用下列任一 HTTP GET 請求：

   - 若要擷取機密，對於 `secretId`，請使用機密的 ARN 或名稱。

     ```
     GET: /secretsmanager/get?secretId=secretId
     ```

   - 若要透過預備標籤擷取先前的機密值或特定版本，對於 `secretId`，請使用機密的 ARN 或名稱；對於 `versionStage`，請使用預備標籤。

     ```
     GET: /secretsmanager/get?secretId=secretId&versionStage=AWSPREVIOUS
     ```

   - 若要依 ID 擷取特定機密版本，對於 `secretId`，請使用機密的 ARN 或名稱；對於 `versionId`，請使用版本 ID。

     ```
     GET: /secretsmanager/get?secretId=secretId&versionId=versionId
     ```

   Example 擷取機密 (Python)

   以下 Python 範例說明如何擷取秘密，並使用 `json.loads` 剖析結果。

   ```
   secrets_extension_endpoint = "http://localhost:" + \
       secrets_extension_http_port + \
       "/secretsmanager/get?secretId=" + \
       <secret_name>

     r = requests.get(secrets_extension_endpoint, headers=headers)

     secret = json.loads(r.text)["SecretString"] # load the Secrets Manager response
    into a Python dictionary, access the secret
   ```

AWS 參數和秘密 Lambda 擴充環境變數

您可以使用下列環境變數設定延伸模組。

如需如何使用環境變數的相關資訊，請參閱《Lambda 開發人員指南》中的使用 Lambda 環境變數。

PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED

設定為 true 以快取參數和秘密。設定為 false 則不快取。預設為 true。

PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE

要快取的秘密和參數數量上限。該值必須介於 0 到 1000 之間。值為 0 表示不快取。如果 SSM_PARAMETER_STORE _TTL 和 SECRETS_MANAGER_TTL 都是 0，則忽略此變數。預設值為 1000。

PARAMETERS_SECRETS_EXTENSION_HTTP_PORT

本機 HTTP 伺服器的連接埠。預設值為 2773。

PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL

延伸模組提供的記錄層級：debug、info、warn、error 或 none。設定為 debug 以查看快取組態。預設值為 info。

PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS

延伸模組用來向 Parameter Store 或 Secrets Manager 提出請求的 HTTP 用戶端連線數量上限。這是每個用戶端的組態。預設值為 3。

SECRETS_MANAGER_TIMEOUT_MILLIS

對 Secrets Manager 的請求逾時 (以毫秒為單位)。值為 0 表示沒有逾時。預設值為 0。

SECRETS_MANAGER_TTL

快取中秘密的 TTL (以秒為單位)。值為 0 表示不快取。最高為 300 秒。如果 PARAMETERS_SECRETS_CACHE_SIZE 為 0，則忽略此變數。預設為 300 秒。

SSM_PARAMETER_STORE_TIMEOUT_MILLIS

對 Parameter Store 的請求逾時 (以毫秒為單位)。值為 0 表示沒有逾時。預設值為 0。

SSM_PARAMETER_STORE_TTL

快取中參數的 TTL (以秒為單位)。值為 0 表示不快取。最高為 300 秒。如果 PARAMETERS_SECRETS_CACHE_SIZE 為 0，則忽略此變數。預設為 300 秒。

使用參數存放區中的 AWS Secrets Manager 秘密

AWS Systems Manager 參數存放區提供安全的階層式儲存空間，可進行組態資料管理和秘密管理。您可以存放密碼、資料庫字串和授權碼之類的資料做為參數值。不過，參數存放區不會為存放的秘密提供自動輪換服務。反之，參數存放區可讓您將秘密存放在 Secrets Manager 中，然後將該秘密當作參數存放區參數來參考。

當您使用 Secrets Manager 設定參數存放區時，`secret-id` 參數存放區需要在名稱字串之前加上正斜線 (/)。

如需詳細資訊，請參閱 AWS Systems Manager 使用者指南中的從參數存放區參數參考 AWS Secrets Manager 秘密。

輪換 AWS Secrets Manager 秘密

輪換是定期更新秘密的過程。當您輪換秘密時，會更新秘密和資料庫或服務中的憑證。在 Secrets Manager 中，您可以為秘密設定自動輪換。

主題

輪換的運作方式

> **Tip**
>
> 對於部分 由其他服務管理的秘密，您可以使用受管輪換。若要使用 受管輪換，您可以先透過管理服務建立機密。

Secrets Manager 輪換使用AWS Lambda函數來更新密碼和資料庫或服務。如需有關使用 Lambda 函數成本的資訊，請參閱 定價。

若要輪換秘密，Secrets Manager 會根據您設定的排程呼叫 Lambda 函數。您可以將排程設定為在一段時間後輪換一次，例如每 30 天輪換一次，也可以建立 Cron 表達式。請參閱 排程表達式。如果您在設定自動輪換時也手動更新機密值，則 Secrets Manager 會在計算下一個輪替日期時將其視為有效輪換。

為了安全起見，Secrets Manager 僅允許 Lambda 輪換函數直接輪換機密。輪換函數無法呼叫第二個 Lambda 函數來輪換機密。

輪換期間，Secrets Manager 會使用預備標籤標示秘密版本。輪換期間，Secrets Manager 會多次呼叫相同的函數，且每次使用不同的參數。Secrets Manager 使用參數的下列 JSON 請求結構來叫用函數：

```
{
    "Step" : "request.type",
    "SecretId" : "string",
    "ClientRequestToken" : "string"
}
```

輪換函數將執行輪換秘密的工作。輪換秘密有四個步驟，分別對應 Lambda 輪換函數中的下列四個步驟：

1. 建立秘密的新版本 (**`createSecret`**)

   輪換的第一步是建立秘密的新版本。在 Secrets Manager 提供的資料庫輪換範本中，Lambda 輪換函數會為新版本產生 32 個字元的密碼。新版本可能包含新密碼、新的使用者名稱和密碼，或更多秘密資訊。Lambda 輪換函數會標示新版本 `AWSPENDING`。

2. 變更資料庫或服務中的憑證 (**`setSecret`**)

   接下來，Lambda 輪換函數會變更資料庫或服務中的憑證，以符合 `AWSPENDING` 版本秘密中的新憑證。根據您的輪換策略，此步驟建立的新使用者可能與現有使用者具備相同的許可。

   Amazon RDS (Oracle 和 Db2 除外) 和 Amazon DocumentDB 的輪換函數自動使用 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 來連線到您的資料庫 (如果有)。否則，他們會使用未加密的連線。

   > **Note**
   >
   > 如果您在 2021 年 12 月 20 日之前設定了自動秘密輪換，則輪換功能可能基於不支援 SSL/TLS 的舊範本。請參閱確定輪換函數的建立時間。如果在 2021 年 12 月 20 日之前建立的輪換函數要支援使用 SSL/TLS 的連線，則您需要重建輪換函數。

3. 測試新的秘密版本 (**`testSecret`**)

   接下來，Lambda 輪換函數會使用 `AWSPENDING` 版本的秘密存取資料庫或服務，以進行測試。根據 輪換函數範本 建立的輪換函數會使用讀取權限測試新的秘密。視應用程式所需的存取權限類型而定，您可以更新函數以包含其他存取權限，例如寫入權限。

4. 完成輪換 (**`finishSecret`**)

最後，Lambda 輪換函數會將標籤 AWSCURRENT 從先前的秘密版本移至此版本，這也會移除相同 API 呼叫中的 AWSPENDING 標籤。您不應該在此之前移除 AWSPENDING，也不應該藉由使用另外的 API 呼叫來將其移除，因為對 Secrets Manager 而言這可能表示未成功完成輪換。Secrets Manager 會將 AWSPREVIOUS 預備標籤新增至先前版本，以便您保留上一個已知良好的秘密版本。

輪換期間，Secrets Manager 會記錄表示輪換狀態的事件。如需詳細資訊，請參閱 the section called "使用 AWS CloudTrail 記錄 "。

如果任何輪換步驟失敗，Secrets Manager 會多次重試整個輪換過程。

輪換成功時，AWSPENDING 預備標籤可能會連接至與 AWSCURRENT 版本相同的版本，或者可能未連接至任何版本。如果 AWSPENDING 預備標籤存在，但未連接至與 AWSCURRENT 相同的版本，則任何以後的輪換調用都會假定之前的輪換請求仍在進行中並傳回錯誤。輪換不成功時，AWSPENDING 預備標籤可能會連接至空的機密版本。如需詳細資訊，請參閱 輪換疑難排解。

輪換成功後，從 AWS Secrets Manager 中擷取秘密 來自 Secrets Manager 的應用程式會自動取得更新的憑證。如需有關各輪換步驟運作方式的詳細資訊，請參閱 the section called "輪換函數範本"。

AWS Secrets Manager 機密的受管輪換

部分服務提供受管輪換，服務會在其中為您設定和管理輪換。透過受管輪換，您無需使用 AWS Lambda 函數即可更新資料庫中的機密和憑證。下列服務提供受管輪換：

- Amazon ECS 服務 Connect 提供 AWS Private Certificate Authority TLS 憑證的受管輪替。如需詳細資訊，請參閱 Amazon 彈性容器服務開發人員指南中的 TLS 與服務 Connect。
- Amazon RDS 為主要使用者憑證提供受管輪換。如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Amazon RDS 和 AWS Secrets Manager 進行密碼管理。
- Amazon Aurora 為主使用者憑證提供受管輪換。如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的使用 Amazon Aurora 和 AWS Secrets Manager 進行密碼管理。
- Amazon Redshift 為管理員密碼提供受管輪換。如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的使用 AWS Secrets Manager 管理 Amazon Redshift 管理員密碼。

如需所有其他類型的機密，請參閱 輪換 秘密。

受管秘密的輪換通常會在一分鐘內完成。在輪換期間，擷取秘密的新連線可能會取得舊版的憑證。在應用程式中，我們強烈建議您遵循最佳實務，使用以應用程式要求的最低權限建立的資料庫使用者，而非使用主使用者。對於應用程式使用者，為取得最高的可用性，您可以使用交替使用者輪換策略。

變更受管輪換的排程 (主控台)

1. 在 Secrets Manager 主控台中開啟受管機密。您可以遵循來自管理服務的連結，或在 Secrets Manager 主控台中搜尋機密。
2. 在 Rotation schedule (輪換排程) 中，在 Schedule expression builder (排程表達式建置器)，或以 Schedule expression (排程表達式) 形式，輸入 UTC 時區的排程。Secrets Manager 會將您的排程儲存為 `rate( )` 或 `cron( )` 表達式。輪換時段會自動在午夜時開始，除非您指定 Start time (開始時間)。您可以每四小時輪換一次秘密。如需詳細資訊，請參閱 排程表達式。
3. (選用) 對於 Window duration (時段持續時間)，選擇您想要 Secrets Manager 輪換秘密的時段長度，例如，三個小時時段 **3h**。時段不得延伸到下一個輪換時段。如果您未指定 Window duration (時段持續時間)，則對於以小時為單位的輪換排程，時段會在一小時後自動關閉。對於以天為單位的輪換排程，時段會在一天結束時自動關閉。
4. 選擇儲存。

變更受管輪換的排程 (AWS CLI)

- 呼叫 `rotate-secret`。下列範例會在每個月的第 1 天和第 15 天 16:00 和 18:00 (UTC) 之間進行。如需更多詳細資訊，請參閱 排程表達式。

```
aws secretsmanager rotate-secret \
    --secret-id MySecret \
    --rotation-rules "{\"ScheduleExpression\": \"cron(0 16 1,15 * ? *)\",
 \"Duration\": \"2h\"}"
```

使用主控台為 Amazon RDS、Amazon Aurora、Amazon DocumentDB 或 Amazon Redshift 秘密設定自動輪換

輪換是定期更新機密的過程。當您輪換機密時，會更新機密和資料庫中的憑證。在 Secrets Manager 中，您可以為資料庫秘密設定自動輪換。

Secrets Manager 會使用 Lambda 函數來輪換秘密。如需概觀，請參閱 the section called “輪換的運作方式”。

> **Tip**
>
> 對於部分 由其他服務管理的秘密，您可以使用受管輪換。若要使用 受管輪換，您可以先透過管理服務建立機密。

若要使用主控台設定輪換，您必須先選擇輪換策略。接著設定秘密進行輪換，如果您還沒有 Lambda 輪換函數，這會建立一個。主控台也會為 Lambda 函數執行角色設定許可。最後一步是確保 Lambda 輪換函數可以透過網路，存取 Secrets Manager 和您的資料庫。

若要開啟自動輪換，您必須有權建立 IAM 執行角色並將許可政策附加到其中。您同時需要 `iam:CreateRole` 和 `iam:AttachRolePolicy` 許可。

> **Warning**
>
> 同時授予身分 `iam:CreateRole` 和 `iam:AttachRolePolicy` 許可會允許身分向自己授予任何許可。

步驟：

- 步驟 1：選擇輪換策略並 (選擇性) 建立超級使用者秘密
- 步驟 2：設定輪換並建立輪換函數
- 步驟 3：(選用) 對輪換函數設定其他許可條件
- 步驟 4：為輪換函數設定網路存取
- 步驟 5：(可選) 自訂旋轉功能
- 後續步驟

步驟 1：選擇輪換策略並 (選擇性) 建立超級使用者秘密

對於 Amazon RDS、Amazon Redshift 和 Amazon DocumentDB，Secrets Manager 提供兩種輪換策略：

單一使用者輪換策略

此策略會在一個秘密中更新一個使用者的憑證。對於 Amazon RDS Db2 執行個體，因為使用者無法變更自己的密碼，因此必須使用單獨的密碼來提供管理員登入資料。這是最簡單的輪換策略，適用於大多數使用案例。特別是，建議您將此策略用於一次性 (臨機操作) 或互動式使用者的憑證。

秘密輪換時，不會中斷開啟的資料庫連線。輪換正在進行時，資料庫中的密碼變更與更新秘密之間，有一小段時間落差。在此期間，資料庫有可能拒絕使用輪換後憑證的呼叫。您可以透過適當的重試策略降低這種風險。輪換之後，新的連線會使用新的憑證。

交替使用者輪換策略

此策略會在一個秘密中更新兩個使用者的憑證。您可以建立第一個使用者，然後在第一次輪換期間，輪換函數會複製該使用者，以建立第二個使用者。每當秘密輪換時，輪換函數都會交替要更新的使用者密碼。由於大多數使用者沒有複製自身的許可，所以您必須提供其他秘密中 `superuser` 的憑證。在資料庫中複製的使用者沒有與原始使用者相同的許可時，建議使用單一使用者輪換策略，以及將其用於一次性 (臨機操作) 或互動式使用者的憑證。

此策略適合具有許可模型的資料庫，其中一個角色擁有資料庫資料表，而第二個角色具有存取資料庫資料表的許可。這也適用於需要高可用性的應用程式。如果應用程式在輪換期間擷取秘密，應用程式仍會取得一組有效的憑證。輪換之後，`user` 和 `user_clone` 憑證都有效。在這種類型的輪換期間，應用程式遭到拒絕的機率比單一使用者輪換更低。如果資料庫託管於伺服器陣列，將密碼變更傳播到所有伺服器需要一段時間，則資料庫有可能拒絕使用新憑證的呼叫。您可以透過適當的重試策略降低這種風險。

Secrets Manager 會建立複製使用者，該使用者擁有與原始使用者相同的許可。如果在建立複製使用者後變更原始使用者的許可，您也必須變更複製使用者的許可。

> **Important**
>
> 如果您選擇交替使用者策略，必須建立資料庫秘密，並在其中儲存資料庫超級使用者憑證。您需要具有超級使用者憑證的秘密，因為輪換會複製第一個使用者，而大多數使用者沒有該許可。

步驟 2：設定輪換並建立輪換函數

Amazon RDS (Oracle 和 Db2 除外) 和 Amazon DocumentDB 的輪換函數自動使用 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 來連線到您的資料庫 (如果有)。否則，他們會使用未加密的連線。

若要為 Amazon RDS、Amazon DocumentDB 或 Amazon Redshift 秘密開啟輪換

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 在 Secrets (機密) 頁面中，選擇機密。

3. 在 Secret details (機密詳細資訊) 頁面的 Rotation configuration (輪換組態) 區段中，選擇 Edit rotation (編輯輪換)。
4. 在 Edit rotation configuration (編輯輪換組態) 對話方塊中，執行以下動作：
   a. 開啟 Automatic rotation (自動輪換)。
   b. 在 Rotation schedule (輪換排程) 中，在 Schedule expression builder (排程表達式建置器)，或以 Schedule expression (排程表達式) 形式，輸入 UTC 時區的排程。Secrets Manager 會將您的排程儲存為 `rate( )` 或 `cron( )` 表達式。輪換時段會自動在午夜時開始，除非您指定 Start time (開始時間)。您可以每四小時輪換一次秘密。如需更多詳細資訊，請參閱 排程表達式。
   c. (選用) 對於 Window duration (時段持續時間)，選擇您想要 Secrets Manager 輪換秘密的時段長度，例如，三個小時時段 **3h**。時段不得延伸到下一個輪換時段。如果您未指定 Window duration (時段持續時間)，則對於以小時為單位的輪換排程，時段會在一小時後自動關閉。對於以天為單位的輪換排程，時段會在一天結束時自動關閉。
   d. (選用) 選擇 Rotate immediately when the secret is stored (存放秘密時立即輪換) 以在儲存變更時輪換您的秘密。如果清除核取方塊，則第一次輪換將按照您設定的排程開始。

      如果輪換失敗，例如因為步驟 3 和 4 尚未完成，Secrets Manager 會多次重試輪換流程。
   e. 在 Rotation function (輪換函數) 下，請執行下列其中一項：
      - 選擇 Create a new Lambda function (新建 Lambda 函數)，然後輸入新函數的名稱。Secrets Manager 會將 `SecretsManager` 新增到函數名稱的開頭。Secrets Manager 會根據適當的範本建立函數，並為 Lambda 執行角色設定必要的許可。
      - 選擇 Use an existing Lambda function (使用現有的 Lambda 函數)，重複使用您用於其他秘密的輪換函數。Recommended VPC configurations (建議的 VPC 組態) 下列出的輪換函數，與資料庫具有相同的 VPC 和安全群組，有助於函數存取資料庫。
   f. 對於輪換策略，選擇單一使用者或交替使用者策略。如需更多詳細資訊，請參閱 the section called “步驟 1：選擇輪換策略並 (選擇性) 建立超級使用者秘密”。
5. 選擇 Save (儲存)。

步驟 3：(選用) 對輪換函數設定其他許可條件

在輪換函數的資源政策中，我們建議您包含內容金鑰 `aws:SourceAccount`，協助防止 Lambda 被當作混淆代理人。對於某些 AWS 服務，為了避免混淆代理人情況，AWS 建議您同時使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件金鑰。但是，如果在您的輪換函數政策中包

含 aws:SourceArn 條件，則輪換函數只能用於輪換該 ARN 指定的秘密。建議您僅包含內容金鑰 aws:SourceAccount，以便可以將輪換函數用於多個秘密。

若要更新輪換函數資源政策

1. 在 Secrets Manager 主控台中，選擇您的秘密，然後在詳細資訊頁面的 Rotation configuration (輪換組態) 之下，選擇 Lambda 輪換函數。Lambda 主控台開啟。
2. 遵循將資源型政策用於 Lambda 中的指示，新增 aws:sourceAccount 條件。

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

如果使用 KMS 金鑰而不是 AWS 受管金鑰 aws/secretsmanager 為秘密加密，Secrets Manager 便會授予 Lambda 執行角色使用該金鑰的許可。您可以透過 SecretARN 加密內容來限制使用解密函數，從而使輪換函數角色僅有權解密其負責輪換的秘密。

更新輪換函數執行角色

1. 在 Lambda 輪換函數中選擇組態，然後在執行角色下選擇角色名稱。
2. 按照修改角色許可政策中的指示新增 kms:EncryptionContext:SecretARN 條件。

```
"Condition": {
    "StringEquals": {
        "kms:EncryptionContext:SecretARN": "SecretARN"
    }
},
```

步驟 4：為輪換函數設定網路存取

若要能輪換秘密，Lambda 輪換函數必須能夠同時存取秘密以及資料庫或服務。

存取秘密

您的 Lambda 輪換函數必須能夠存取 Secrets Manager 服務端點。如果您的 Lambda 函數可以存取網際網路，那麼您可以使用公有端點。若要尋找端點，請參閱 the section called “Secrets Manager 端點”。

如果 Lambda 函數在無法存取網際網路的 VPC 中執行，建議您在 VPC 中設定 Secrets Manager 服務私有端點。然後，您的 VPC 可以攔截發送到公有區域端點的請求，並將其重新導向到私有端點。如需更多詳細資訊，請參閱 VPC 端點。

或者，您可以啟用 Lambda 函數來存取 Secrets Manager 公有端點，方法是將 NAT 閘道或網際網路閘道新增至您的 VPC，以便來自 VPC 的流量到達公有端點。這將使 VPC 暴露在較高的風險下，因為 IP 地址 (用於閘道) 可能會遭到來自公有網際網路的攻擊。

存取資料庫或服務

如果資料庫或服務正在 VPC 中的 Amazon EC2 執行個體上執行，建議您將 Lambda 函數設定為在相同的 VPC 中執行。然後輪換函數就能直接與您的服務進行通訊。如需詳細資訊，請參閱設定 VPC 存取 。

若要允許 Lambda 函數存取資料庫或服務，您必須確定附加至 Lambda 輪換函數的安全群組允許連至資料庫或服務的傳出連線。此外，您必須確定附加至資料庫或服務的安全群組允許來自 Lambda 輪換函數的傳入連線。

對於由其他 AWS 服務管理超級使用者機密的交替使用者輪換，Lambda 輪換函數必須能夠呼叫此服務端點以取得資料庫連線資訊。建議您為資料庫服務設定 VPC 端點。如需詳細資訊，請參閱：

- 在《Amazon RDS 使用者指南》中的 Amazon RDS API 和介面 VPC 端點。
- 在 Amazon Redshift 管理指南中使用 VPC 端點。

步驟 5：(可選) 自訂旋轉功能

在極少數情況下，您可能會想要自訂旋轉功能。例如，使用者交替輪換時，Secrets Manager 會複製第一個使用者的執行期組態參數來建立複製的使用者。如果您想要包含更多屬性，或變更哪些屬性以授予複製的使用者，則需要更新 `set_secret` 函數中的程式碼。

對於另一個例子，對於 Amazon RDS MySQL，在交替使用者輪換時，Secrets Manager 會建立名稱不超過 16 個字元的複製使用者。您可以修改旋轉功能以允許更長的用戶名。MySQL 版本 5.7 及更高版本支持用戶名最多 32 個字符，但是 Secrets Manager 附加「_clone」（六個字符）到用戶名的末尾，所以你必須保持用戶名最多 26 個字符。

開啟 Lambda 輪換函數以供編輯

1. 在 Secrets Manager 主控台中，選擇您的秘密。
2. 在 Rotation configuration (輪換組態) 區段的 Lambda rotation function (Lambda 輪換函數) 中，選擇您的輪換函數。

   Lambda 主控台開啟。

   - 若要變更函式中的程式碼，請向下捲動至程式碼來源區段。
   - 對於 MySQL 版本 5.7 及更高版本，為了交替用戶旋轉，要更改最大用戶名長度，請在環境變量下進行更改`USERNAME_CHARACTER_LIMIT`。

後續步驟

請參閱 the section called “ 輪換疑難排解”。

使用主控台為 AWS Secrets Manager 秘密設定自動輪換

輪換是定期更新機密的過程。當您輪換秘密時，會更新秘密以及該秘密所針對資料庫或服務中的憑證。

Secrets Manager 會使用 Lambda 函數來輪換秘密。如需概觀，請參閱 the section called “輪換的運作方式”。

您也可以使用 AWS CLI 設定輪換。如需詳細資訊，請參閱 自動輪換 (AWS CLI)。

若要使用主控台設定輪換，請先設定秘密以進行輪換。在該步驟中，您也會建立空白 Lambda 輪換函數。接下來，您可以設定輪換函數和 Lambda 執行角色的許可。接著撰寫輪換函數程式碼。最後一步是確保 Lambda 輪換函數可以透過網路，存取 Secrets Manager 和您的資料庫或服務。

如需資料庫秘密，請參閱the section called “資料庫秘密的自動輪換 (主控台)”。

若要開啟自動輪換，您必須有權建立 IAM 執行角色並將許可政策附加到其中。您同時需要 `iam:CreateRole` 和 `iam:AttachRolePolicy` 許可。

> **Warning**
>
> 同時授予身分 `iam:CreateRole` 和 `iam:AttachRolePolicy` 許可會允許身分向自己授予任何許可。

步驟：

步驟 1：設定秘密以進行輪換

在此步驟中，您會為秘密設定輪換排程，並建立空白輪換函數。在撰寫完輪換函數之前，系統不會輪換您的秘密。如果您在撰寫輪換函數之前排程輪換，或是因為任何原因而失敗，Secrets Manager 會多次重試輪換函數。

若要設定輪換並建立空白輪換函數

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 在 Secrets (機密) 頁面中，選擇機密。
3. 在 Secret details (機密詳細資訊) 頁面的 Rotation configuration (輪換組態) 區段中，選擇 Edit rotation (編輯輪換)。在 Edit rotation configuration (編輯輪換組態) 對話方塊中，執行以下動作：

    a. 開啟 Automatic rotation (自動輪換)。

    b. 在 Rotation schedule (輪換排程) 中，在 Schedule expression builder (排程表達式建置器)，或以 Schedule expression (排程表達式) 形式，輸入 UTC 時區的排程。Secrets Manager 會將您的排程儲存為 `rate( )` 或 `cron( )` 表達式。輪換時段會自動在午夜時開始，除非您指定 Start time (開始時間)。您可以每四小時輪換一次秘密。如需詳細資訊，請參閱 排程表達式。

    c. (選用) 對於 Window duration (時段持續時間)，選擇您想要 Secrets Manager 輪換秘密的時段長度，例如，三個小時時段 **3h**。時段不得延伸到下一個輪換時段。如果您未指定 Window duration (時段持續時間)，則對於以小時為單位的輪換排程，時段會在一小時後自動關閉。對於以天為單位的輪換排程，時段會在一天結束時自動關閉。

    d. (選用) 選擇 Rotate immediately when the secret is stored (存放秘密時立即輪換) 以在儲存變更時輪換您的秘密。如果清除核取方塊，則第一次輪換將按照您設定的排程開始。

    e. 在 Rotation function (輪換函數) 下，選擇 Create function (建立函數)。Lambda 主控台會在新視窗中開啟。

- 在 Lambda 主控台的 Create function (建立函數) 頁面上，執行下列任一操作：
  - 如果您看到 Browse serverless app repository (瀏覽無伺服器應用程式儲存庫)，請加以選擇。

    A. 在 [公用應用程式] 下方的搜尋方塊中，輸入SecretsManagerRotationTemplate。

    B. 選取 Show apps that create custom IAM roles or resource policies (顯示建立自訂 IAM 角色或資源政策的應用程式)。

    C. 選擇SecretsManagerRotationTemplate瓷磚。

    D. 在 Review, configure and deploy (檢閱、設定與部署) 頁面的 Application settings (應用程式設定) 圖磚中，填寫必要欄位，然後選擇 Deploy (部署)。如需端點清單，請參閱 the section called “Secrets Manager 端點”。
  - 如果您沒有看到 Browse serverless app repository (瀏覽無伺服器應用程式儲存庫)，您的 AWS 區域可能不支援 AWS Serverless Application Repository。選擇 從頭開始撰寫。

    A. 針對 Function name (函數名稱)，輸入您輪換函數的名稱。

    B. 針對 Runtime (執行階段)，選擇 Python 3.9。

    C. 新的 Lambda 函數開啟時，向下捲動以選擇 Configuration (組態)，然後在左側選擇 Permissions (許可)。

    D. 向下捲動至 Resource-based policy (基於資源的政策)，然後選擇 Add permissions (新增許可)，授予 Secrets Manager 叫用函數的許可。若要將資源政策連接到 Lambda 函數，請參閱將資源型政策用於 Lambda。

    下列政策說明如何允許 Secrets Manager 叫用 Lambda 函數。

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "secretsmanager.amazonaws.com"
            },
        "Action": "lambda:InvokeFunction",
        "Resource": "LambdaRotationFunctionARN"
```

```
    }
    ]
}
```

f. 切換回 Secrets Manager 主控台，將新的輪換函數連接至您的秘密。

g. 對於 Lambda rotation function (Lambda 輪換函數)，選擇重新整理按鈕。然後，在函數清單中，選擇您的新函數。

h. 選擇儲存。

步驟 2：為輪換函數設定許可

Lambda 輪換函數需要許可以存取 Secrets Manager 中的秘密，且需要許可以存取您的資料庫或服務。在此步驟中，您會將這些許可授予 Lambda 執行角色。如果使用 KMS 金鑰為秘密加密，而不是 AWS 受管金鑰 `aws/secretsmanager`，那麼您需要將使用該金鑰的許可授予 Lambda 執行角色。您可以透過 SecretARN 加密內容來限制使用解密函數，從而使輪換函數角色僅有權解密其負責輪換的秘密。如需政策範例，請參閱輪換的許可。

如需指示，請參閱《AWS Lambda 開發人員指南》中的 Lambda 執行角色。

步驟 3：(選用) 對輪換函數設定其他許可條件

在輪換函數的資源政策中，我們建議您包含內容金鑰 `aws:SourceAccount`，協助防止 Lambda 被當作混淆代理人。對於某些 AWS 服務，為了避免混淆代理人情況，AWS 建議您同時使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件金鑰。但是，如果在您的輪換函數政策中包含 `aws:SourceArn` 條件，則輪換函數只能用於輪換該 ARN 指定的秘密。建議您僅包含內容金鑰 `aws:SourceAccount`，以便可以將輪換函數用於多個秘密。

若要更新輪換函數資源政策

1. 在 Secrets Manager 主控台中，選擇您的秘密，然後在詳細資訊頁面的 Rotation configuration (輪換組態) 之下，選擇 Lambda 輪換函數。Lambda 主控台開啟。
2. 遵循將資源型政策用於 Lambda 中的指示，新增 `aws:sourceAccount` 條件。

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

步驟 4：為輪換函數設定網路存取

若要能輪換秘密，Lambda 輪換函數必須能夠存取秘密。如果您的秘密包含憑證，則 Lambda 函數也必須能夠存取這些憑證的來源，例如資料庫或服務。

存取秘密

您的 Lambda 輪換函數必須能夠存取 Secrets Manager 服務端點。如果您的 Lambda 函數可以存取網際網路，那麼您可以使用公有端點。若要尋找端點，請參閱 the section called “Secrets Manager 端點”。

如果 Lambda 函數在無法存取網際網路的 VPC 中執行，建議您在 VPC 中設定 Secrets Manager 服務私有端點。然後，您的 VPC 可以攔截發送到公有區域端點的請求，並將其重新導向到私有端點。如需詳細資訊，請參閱 VPC 端點。

或者，您可以啟用 Lambda 函數來存取 Secrets Manager 公有端點，方法是將 NAT 閘道或網際網路閘道新增至您的 VPC，以便來自 VPC 的流量到達公有端點。這將使 VPC 暴露在較高的風險下，因為 IP 地址 (用於閘道) 可能會遭到來自公有網際網路的攻擊。

(選用) 存取資料庫或服務

對於 API 金鑰等秘密，沒有需要與秘密一起更新的來源資料庫或服務。

如果資料庫或服務正在 VPC 中的 Amazon EC2 執行個體上執行，建議您將 Lambda 函數設定為在相同的 VPC 中執行。然後輪換函數就能直接與您的服務進行通訊。如需詳細資訊，請參閱設定 VPC 存取 。

若要允許 Lambda 函數存取資料庫或服務，您必須確定附加至 Lambda 輪換函數的安全群組允許連至資料庫或服務的傳出連線。此外，您必須確定附加至資料庫或服務的安全群組允許來自 Lambda 輪換函數的傳入連線。

步驟 5：撰寫輪換函數程式碼

您在步驟 1 建立的輪換函數是您函數的起點。您要為您的特定使用案例，撰寫程式碼。對於可以旋轉 Amazon 密碼的函數，您可以從 ElastiCache 秘 Sec rets Manager 提供的適當模板中複製代碼。

撰寫函數時，請謹慎納入偵錯或記錄陳述式。這些陳述式可能會導致函數中的資訊寫入 Amazon CloudWatch，因此您需要確保記錄檔不包含在開發期間收集的任何敏感資訊。

為了安全起見，Secrets Manager 僅允許 Lambda 輪換函數直接輪換機密。輪換函數無法呼叫第二個 Lambda 函數來輪換機密。

如需日誌陳述式的範例，請參閱the section called “輪換函數範本”原始程式碼。

如果您使用外部二進位檔案和程式庫 (例如連線至資源)，則需要管理修補並保留它們 up-to-date。

如需偵錯建議，請參閱 Testing and debugging serverless applications (測試與偵錯無伺服器應用程式)。

開啟 Lambda 輪換函數以供編輯

1. 在 Secrets Manager 主控台中，選擇您的秘密。
2. 在 Rotation configuration (輪換組態) 區段的 Lambda rotation function (Lambda 輪換函數) 中，選擇您的輪換函數。

   Lambda 主控台開啟。

   - 若要變更函式中的程式碼，請向下捲動至程式碼來源區段。
   - 對於 MySQL 版本 5.7 及更高版本，為了交替用戶旋轉，要更改最大用戶名長度，請在環境變量下進行更改USERNAME_CHARACTER_LIMIT。

如果您的函數還沒有，請從 SecretsManagerRotationTemplate.

輪換秘密有四個步驟，分別對應 Lambda 輪換函數的下列四個方法。

方法

- create_secret
- set_secret
- test_secret
- finish_secret

create_secret

在 `create_secret` 中，您會先使用傳入的 `ClientRequestToken` 呼叫 `get_secret_value`，確認是否有秘密。如果沒有秘密，您可以使用 `create_secret` 和字符，建立新的秘密當作 `VersionId`。您便可使用 `get_random_password`，產生新的秘密值。您必須確定新的秘密值只包含對資料庫或服務有效的字元。使用 `ExcludeCharacters` 參數排除字元。呼叫 `put_secret_value`，將其與預備標籤 `AWSPENDING` 一起儲存。將新的秘密值儲存在 `AWSPENDING` 中，有助於確保等冪性。如果輪換因任何原因而失敗，您可以在後續呼叫中參考該秘密值。請參閱 How do I make my Lambda function idempotent (如何讓 Lambda 函數等冪)。

在測試函數時，請使用 AWS CLI 查看版本階段：呼叫 `describe-secret` 並查看 `VersionIdsToStages`。

set_secret

在 `set_secret` 中，您會變更資料庫或服務中的憑證，以符合 AWSPENDING 版本秘密中的新秘密值。

如果您將陳述式傳遞至解譯陳述式的服務 (例如資料庫)，請使用查詢參數化。如需詳細資訊，請參閱 OWASP 網站上的 Query Parameterization Cheat Sheet (查詢參數化速查表)。

輪換函數是具有特殊權限的代理人，有權存取和修改 Secrets Manager 秘密和目標資源中的客戶憑證。為了防止潛在的混淆代理人攻擊，您必須確保攻擊者無法使用該函數存取其他資源。在您更新憑證之前：

- 查看 AWSCURRENT 版本秘密中的憑證是否有效。如果 AWSCURRENT 憑證無效，請放棄輪換嘗試。
- 查看 AWSCURRENT 和 AWSPENDING 秘密值是否針對相同的資源。對於使用者名稱和密碼，請查看 AWSCURRENT 和 AWSPENDING 使用者名稱是否相同。
- 檢查目的地服務資源是否相同。對於資料庫，請查看 AWSCURRENT 和 AWSPENDING 主機名稱是否相同。

test_secret

在 `test_secret` 中，您會使用 AWSPENDING 版本的秘密存取資料庫或服務，以進行測試。

finish_secret

在 `finish_secret` 中，您會使用 `update_secret_version_stage`，將預備標籤 AWSCURRENT 從先前的秘密版本移至新的秘密版本。Secrets Manager 會自動將 AWSPREVIOUS 預備標籤新增至先前版本，以便您保留上一個已知良好的秘密版本。

後續步驟

請參閱the section called “ 輪換疑難排解”。

使用 AWS CLI 為 AWS Secrets Manager 秘密設定自動輪換

輪換是定期更新機密的過程。當您輪換秘密時，會更新秘密以及該秘密所針對資料庫或服務中的憑證。

Secrets Manager 會使用 Lambda 函數來輪換秘密。如需概觀，請參閱 the section called “輪換的運作方式”。

您也可以使用主控台來設定輪換。如需詳細資訊，請參閱 自動輪換 (主控台)。

若要使用 AWS CLI 設定輪換，輪換 Amazon RDS、Amazon Redshift 或 Amazon DocumentDB 秘密時，您必須先選擇一個the section called “輪換策略”。如果您選擇交替使用者策略，必須儲存具有資料庫超級使用者憑證的另外秘密。接下來，您會撰寫輪換函數程式碼。Secrets Manager 提供範本，您可以用來撰寫函數。您接著可以使用程式碼建立 Lambda 函數，並為 Lambda 函數和 Lambda 執行角色設定許可。下一步是確保 Lambda 輪換函數可以透過網路，存取 Secrets Manager 和您的資料庫或服務。最後，您可以設定秘密以進行輪換。

若要開啟自動輪換，您必須有權建立 IAM 執行角色並將許可政策附加到其中。您同時需要 `iam:CreateRole` 和 `iam:AttachRolePolicy` 許可。

> **Warning**
>
> 同時授予身分 `iam:CreateRole` 和 `iam:AttachRolePolicy` 許可會允許身分向自己授予任何許可。

步驟：

- (選用) 步驟 1：建立超級使用者秘密
- 步驟 2：撰寫輪換函數程式碼
- 步驟 3：建立 Lambda 函數和執行角色
- 步驟 4：設定網路存取
- 步驟 5：設定秘密以進行輪換
- 後續步驟

(選用) 步驟 1：建立超級使用者秘密

對於 Amazon RDS、Amazon Redshift 和 Amazon DocumentDB，Secrets Manager 提供兩種輪換策略：

單一使用者輪換策略

此策略會在一個秘密中更新一個使用者的憑證。對於 Amazon RDS Db2 執行個體，因為使用者無法變更自己的密碼，因此必須使用單獨的密碼來提供管理員登入資料。這是最簡單的輪換策略，適用於大多數使用案例。特別是，建議您將此策略用於一次性 (臨機操作) 或互動式使用者的憑證。

秘密輪換時，不會中斷開啟的資料庫連線。輪換正在進行時，資料庫中的密碼變更與更新秘密之間，有一小段時間落差。在此期間，資料庫有可能拒絕使用輪換後憑證的呼叫。您可以透過適當的重試策略降低這種風險。輪換之後，新的連線會使用新的憑證。

交替使用者輪換策略

此策略會在一個秘密中更新兩個使用者的憑證。您可以建立第一個使用者，然後在第一次輪換期間，輪換函數會複製該使用者，以建立第二個使用者。每當秘密輪換時，輪換函數都會交替要更新的使用者密碼。由於大多數使用者沒有複製自身的許可，所以您必須提供其他秘密中 `superuser` 的憑證。在資料庫中複製的使用者沒有與原始使用者相同的許可時，建議使用單一使用者輪換策略，以及將其用於一次性 (臨機操作) 或互動式使用者的憑證。

此策略適合具有許可模型的資料庫，其中一個角色擁有資料庫資料表，而第二個角色具有存取資料庫資料表的許可。這也適用於需要高可用性的應用程式。如果應用程式在輪換期間擷取秘密，應用程式仍會取得一組有效的憑證。輪換之後，`user` 和 `user_clone` 憑證都有效。在這種類型的輪換期間，應用程式遭到拒絕的機率比單一使用者輪換更低。如果資料庫託管於伺服器陣列，將密碼變更傳播到所有伺服器需要一段時間，則資料庫有可能拒絕使用新憑證的呼叫。您可以透過適當的重試策略降低這種風險。

Secrets Manager 會建立複製使用者，該使用者擁有與原始使用者相同的許可。如果在建立複製使用者後變更原始使用者的許可，您也必須變更複製使用者的許可。

> **Important**
>
> 如果您選擇交替使用者策略，必須建立資料庫秘密，並在其中儲存資料庫超級使用者憑證。您需要具有超級使用者憑證的秘密，因為輪換會複製第一個使用者，而大多數使用者沒有該許可。

步驟 2：撰寫輪換函數程式碼

若要輪換秘密，您需要輪換函數。輪換函數是 Lambda 函數，Secrets Manager 會呼叫以輪換秘密。

對於可以旋轉 Amazon RDS，Amazon Aurora，亞馬 Amazon Redshift，亞 Amazon DocumentDB 或 Amazon ElastiCache 秘密的功能，您可以從 Secrets Manager 提供的適當模板複製代碼。

對於所有其他類型的秘密，請使用通用輪換範本，開始撰寫自己的輪換函數。

將您的輪換函數以及任何所需相依性，一起儲存在 ZIP 檔案 *my-function.zip* 中。

撰寫函數時，請謹慎納入偵錯或記錄陳述式。這些陳述式可能會導致函數中的資訊寫入 Amazon CloudWatch，因此您需要確保記錄檔不包含在開發期間收集的任何敏感資訊。

為了安全起見，Secrets Manager 僅允許 Lambda 輪換函數直接輪換機密。輪換函數無法呼叫第二個 Lambda 函數來輪換機密。

如需日誌陳述式的範例，請參閱the section called “輪換函數範本”原始程式碼。

如果您使用外部二進位檔案和程式庫 (例如連線至資源)，則需要管理修補並保留它們 up-to-date。

如需偵錯建議，請參閱 Testing and debugging serverless applications (測試與偵錯無伺服器應用程式)。

開啟 Lambda 輪換函數以供編輯

1. 在 Secrets Manager 主控台中，選擇您的秘密。
2. 在 Rotation configuration (輪換組態) 區段的 Lambda rotation function (Lambda 輪換函數) 中，選擇您的輪換函數。

   Lambda 主控台開啟。

   - 若要變更函式中的程式碼，請向下捲動至程式碼來源區段。
   - 對於 MySQL 版本 5.7 及更高版本，為了交替用戶旋轉，要更改最大用戶名長度，請在環境變量下進行更改USERNAME_CHARACTER_LIMIT。

如果您的函數還沒有，請從 SecretsManagerRotationTemplate.

輪換秘密有四個步驟，分別對應 Lambda 輪換函數的下列四個方法。

方法

- create_secret
- set_secret
- test_secret
- finish_secret

create_secret

在 create_secret 中，您會先使用傳入的 ClientRequestToken 呼叫 get_secret_value，確認是否有秘密。如果沒有秘密，您可以使用 create_secret 和字符，建立新的秘密當作 VersionId。您便可使用 get_random_password，產生新的秘密值。您必須確定新的秘密值只包含對資料庫或服務有效的字元。使用 ExcludeCharacters 參數排除字元。呼叫 put_secret_value，將其與預備標籤 AWSPENDING 一起儲存。將新的秘密值儲存在 AWSPENDING 中，有助於確保等冪性。如果輪換因任何原因而失敗，您可以在後續呼叫中參考該秘密值。請參閱 How do I make my Lambda function idempotent (如何讓 Lambda 函數等冪)。

在測試函數時，請使用 AWS CLI 查看版本階段：呼叫 describe-secret 並查看 VersionIdsToStages。

set_secret

在 set_secret 中，您會變更資料庫或服務中的憑證，以符合 AWSPENDING 版本秘密中的新秘密值。

如果您將陳述式傳遞至解譯陳述式的服務 (例如資料庫)，請使用查詢參數化。如需詳細資訊，請參閱 OWASP 網站上的 Query Parameterization Cheat Sheet (查詢參數化速查表)。

輪換函數是具有特殊權限的代理人，有權存取和修改 Secrets Manager 秘密和目標資源中的客戶憑證。為了防止潛在的混淆代理人攻擊，您必須確保攻擊者無法使用該函數存取其他資源。在您更新憑證之前：

- 查看 AWSCURRENT 版本秘密中的憑證是否有效。如果 AWSCURRENT 憑證無效，請放棄輪換嘗試。
- 查看 AWSCURRENT 和 AWSPENDING 秘密值是否針對相同的資源。對於使用者名稱和密碼，請查看 AWSCURRENT 和 AWSPENDING 使用者名稱是否相同。
- 檢查目的地服務資源是否相同。對於資料庫，請查看 AWSCURRENT 和 AWSPENDING 主機名稱是否相同。

test_secret

在 test_secret 中，您會使用 AWSPENDING 版本的秘密存取資料庫或服務，以進行測試。

finish_secret

在 finish_secret 中，您會使用 update_secret_version_stage，將預備標籤 AWSCURRENT 從先前的秘密版本移至新的秘密版本。Secrets Manager 會自動將 AWSPREVIOUS 預備標籤新增至先前版本，以便您保留上一個已知良好的秘密版本。

步驟 3：建立 Lambda 函數和執行角色

Lambda 執行角色是叫用函數時 Lambda 擔任的角色。

建立 Lambda 輪換函數和執行角色

1. 為 Lambda 執行角色建立信任政策，並將其儲存為 JSON 檔案。如需範例，請參閱 輪換的許可。該政策必須：
   - 允許角色對秘密呼叫 Secrets Manager 作業。
   - 如果秘密使用 aws/secretsmanager 以外的金鑰加密，請允許角色使用 KMS 金鑰。
   - 允許角色呼叫秘密針對的服務。
2. 建立 Lambda 執行角色，並透過呼叫 iam create-role 套用信任政策。

```
aws iam create-role \
    --role-name rotation-lambda-role \
    --assume-role-policy-document file://trust-policy.json
```

3. (選用) 對於包含 Amazon RDS 憑證的機密，如果您使用交替使用者策略，且超級使用者機密由 Amazon RDS 或 Aurora 管理，則必須允許輪換函數呼叫 Amazon RDS 上的唯讀 API，以便取得資料庫的連線資訊。若要這麼做，請透過呼叫，將AWS受管政策 AmazonRDS 附加 ReadOnlyAccess至 Lambda 函數執行角色。iam attach-role-policy

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/AmazonRDSReadOnlyAccess \
    --role-name rotation-lambda-role
```

4. 呼叫 lambda create-function，以 ZIP 檔案建立 Lambda 函數。

```
aws lambda create-function \
    --function-name my-rotation-function \
    --runtime python3.9 \
    --zip-file fileb://my-function.zip \
    --handler my-handler \
```

```
    --role arn:aws:iam::123456789012:role/service-role/rotation-lambda-role
```

5. 對 Lambda 函數設定資源政策，允許 Secrets Manager 透過呼叫 `lambda add-permission` 來叫用該函數。範例命令包括 `source-account`，協助防止 Lambda 被當作混淆代理人。

```
aws lambda add-permission \
    --function-name my-rotation-function \
    --action lambda:InvokeFunction \
    --statement-id SecretsManager \
    --principal secretsmanager.amazonaws.com \
    --source-account 123456789012
```

步驟 4：設定網路存取

若要能輪換秘密，Lambda 輪換函數必須能夠同時存取秘密以及資料庫或服務。

存取秘密

您的 Lambda 輪換函數必須能夠存取 Secrets Manager 服務端點。如果您的 Lambda 函數可以存取網際網路，那麼您可以使用公有端點。若要尋找端點，請參閱 the section called “Secrets Manager 端點”。

如果 Lambda 函數在無法存取網際網路的 VPC 中執行，建議您在 VPC 中設定 Secrets Manager 服務私有端點。然後，您的 VPC 可以攔截發送到公有區域端點的請求，並將其重新導向到私有端點。如需詳細資訊，請參閱 VPC 端點。

或者，您可以啟用 Lambda 函數來存取 Secrets Manager 公有端點，方法是將 NAT 閘道或網際網路閘道新增至您的 VPC，以便來自 VPC 的流量到達公有端點。這將使 VPC 暴露在較高的風險下，因為 IP 地址 (用於閘道) 可能會遭到來自公有網際網路的攻擊。

存取資料庫或服務

如果資料庫或服務正在 VPC 中的 Amazon EC2 執行個體上執行，建議您將 Lambda 函數設定為在相同的 VPC 中執行。然後輪換函數就能直接與您的服務進行通訊。如需詳細資訊，請參閱設定 VPC 存取 。

若要允許 Lambda 函數存取資料庫或服務，您必須確定附加至 Lambda 輪換函數的安全群組允許連至資料庫或服務的傳出連線。此外，您必須確定附加至資料庫或服務的安全群組允許來自 Lambda 輪換函數的傳入連線。

對於由其他 AWS 服務管理超級使用者機密的交替使用者輪換，Lambda 輪換函數必須能夠呼叫此服務端點以取得資料庫連線資訊。建議您為資料庫服務設定 VPC 端點。如需詳細資訊，請參閱：

- 在《Amazon RDS 使用者指南》中的 Amazon RDS API 和介面 VPC 端點。
- 在 Amazon Redshift 管理指南中使用 VPC 端點。

步驟 5：設定秘密以進行輪換

若要為您的秘密開啟自動輪換功能，請呼叫 `rotate-secret`。您可以使用 `cron()` 或 `rate()` 排程表達式來設定輪換排程，也可以設定輪換時段時長。您可以每四小時輪換一次秘密。如需詳細資訊，請參閱 排程表達式。

```
aws secretsmanager rotate-secret \
    --secret-id MySecret \
    --rotation-lambda-arn arn:aws:lambda:Region:123456789012:function:my-rotation-
function \
    --rotation-rules "{\"ScheduleExpression\": \"cron(0 16 1,15 * ? *)\", \"Duration\":
 \"2h\"}"
```

後續步驟

請參閱the section called “ 輪換疑難排解”。

立即輪換 AWS Secrets Manager 秘密

您只能輪換已設定輪換的秘密。若要確定是否已設定秘密進行輪換，請在主控台中檢視該秘密，然後向下捲動至 Rotation configuration (輪換組態) 部分。如果 Rotation status (輪換狀態) 為 Enabled (已啟用)，則會設定秘密進行輪換。或在 AWS CLI 中，呼叫 `describe-secret`。如果回應有 `RotationLambdaARN` 和 `RotationRules`，則會設定秘密進行輪換。如果沒有，您可以設定自動輪換：

- 資料庫秘密的自動輪換 (主控台)
- 自動輪換 (主控台)
- 自動輪換 (AWS CLI)

若要立即輪換秘密 (主控台)

1. 開啟位於的 Secrets Manager 主控台 https://console.aws.amazon.com/secretsmanager/。

2. 選擇您的秘密。
3. 在秘密詳細資訊頁面的 Rotation configuration (輪換組態) 中，選擇 Rotate secret immediately (立即輪換秘密)。
4. 在 Rotate secret (輪換秘密) 對話方塊中，選擇 Rotate (輪換)。

AWS CLI

Example 立即輪換秘密

下列 `rotate-secret` 範例會立即開始輪換。輸出會顯示輪換建立的新機密版本的 VersionId。機密必須已設定輪換。

```
aws secretsmanager rotate-secret \
    --secret-id MyTestSecret
```

AWS Secrets Manager 旋轉函數模板

Secrets Manager 提供以下項目的輪換函數範本：

- Amazon RDS 和 Amazon Aurora
- Amazon DocumentDB (with MongoDB compatibility)
- Amazon Redshift
- Amazon ElastiCache
- 其他類型的秘密

若要使用範本，請參閱：

- 輪換 Amazon RDS、Amazon Aurora、Amazon Redshift 和 Amazon DocumentDB 憑證
- 其他類型的憑證 (適用於主控台的指示)
- 其他類型的認證 (AWS CLI 指示)

範本支援 Python 3.9。

要編寫自己的旋轉函數，請參閱編寫旋轉函數。

Amazon RDS 和 Amazon Aurora

主題

Amazon RDS Db2 單用戶

- 樣板名稱:數據 SecretsManager庫 RotationSingleUser
- 輪換策略：輪換策略：單一使用者。
- **`SecretString`** 結構：the section called “Amazon RDS Db2 秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSDB2 SecretsManager RotationSingleUser /lambda_function.py
- 依賴關係：python-ibmdb

Amazon RDS Db2 交替用戶

- 樣板名稱:數據 SecretsManager庫 RotationMultiUser
- 輪換策略：the section called “交替使用者”。
- **`SecretString`** 結構：the section called “Amazon RDS Db2 秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSDB2 SecretsManager RotationMultiUser /lambda_function.py

- 依賴關係：python-ibmdb

Amazon RDS MariaDB 單一使用者

- 樣板名稱: SecretsManagerRDS B RotationSingleUser
- 輪換策略：輪換策略：單一使用者。
- **`SecretString`** 結構：the section called “Amazon RDS MariaDB 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSB SecretsManager RotationSingleUser /lambda_function.py
- 依賴關PyMy係：

Amazon RDS MariaDB 交替使用者

- 樣板名稱: SecretsManagerRDS B RotationMultiUser
- 輪換策略：the section called “交替使用者”。
- **`SecretString`** 結構：the section called “Amazon RDS MariaDB 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSB SecretsManager RotationMultiUser /lambda_function.py
- 依賴關PyMy係：

Amazon RDS 和 Amazon Aurora MySQL 單一使用者

- 樣板名稱: SecretsManager模板 RotationSingleUser
- 輪換策略：the section called “單一使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS 和 Amazon Aurora MySQL 的秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSYSQL SecretsManager RotationSingleUser /lambda_function.py
- 依賴關PyMy係：

Amazon RDS 和 Amazon Aurora MySQL 交替使用者

- 樣板名稱: SecretsManager模板 RotationMultiUser

- 輪換策略：the section called “交替使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS 和 Amazon Aurora MySQL 的秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSYSQL SecretsManager RotationMultiUser /lambda_function.py
- 依賴關PyMy係：

Amazon RDS Oracle 單一使用者

- 樣板名稱 SecretsManager:OracleRotationSingleUser
- 輪換策略：the section called “單一使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS Oracle 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDS SecretsManager OracleRotationSingleUser /lambda_function.py
- 依賴：蟒蛇座管理數據 2.0.1

Amazon RDS Oracle 交替使用者

- 樣板名稱 SecretsManager:OracleRotationMultiUser
- 輪換策略：the section called “交替使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS Oracle 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDS SecretsManager OracleRotationMultiUser /lambda_function.py
- 依賴：蟒蛇座管理數據 2.0.1

Amazon RDS 和 Amazon Aurora PostgreSQL 單一使用者

- 樣板名稱: SecretsManager格雷斯 RotationSingleUser
- 輪換策略：輪換策略：單一使用者。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS 和 Amazon Aurora PostgreSQL 的秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSPLCSQL SecretsManager RotationSingleUser /lambda_function.py

- 相依PyGre性：

Amazon RDS 和 Amazon Aurora PostgreSQL 交替使用者

- 樣板名稱: SecretsManager格雷斯 RotationMultiUser
- 輪換策略：the section called “交替使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS 和 Amazon Aurora PostgreSQL 的秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSPLCSQL SecretsManager RotationMultiUser /lambda_function.py
- 相依PyGre性：

Amazon RDS Microsoft SQLServer 單一使用者

- 樣板名稱: SecretsManager資料庫 ServerRotationSingleUser
- 輪換策略：the section called “單一使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS Microsoft SQLServer 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSSQL SecretsManager ServerRotationSingleUser /lambda_function.py
- 相依性：Pymssql 2.2.2

Amazon RDS Microsoft SQLServer 交替使用者

- 樣板名稱: SecretsManager資料庫 ServerRotationMultiUser
- 輪換策略：the section called “交替使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon RDS Microsoft SQLServer 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/RDSSQL SecretsManager ServerRotationMultiUser /lambda_function.py
- 相依性：Pymssql 2.2.2

Amazon DocumentDB (with MongoDB compatibility)

Amazon DocumentDB 單一使用者

- 樣板名稱: SecretsManagerMongoDB RotationSingleUser
- 輪換策略：the section called “單一使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon DocumentDB 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/DB / lambda_function.py SecretsManagerMongo RotationSingleUser
- 相依性：Pymongo 3.2

Amazon DocumentDB 交替使用者

- 樣板名稱: SecretsManagerMongoDB RotationMultiUser
- 輪換策略：the section called “交替使用者”。
- 預期的 **`SecretString`** 結構：the section called “Amazon DocumentDB 機密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主/DB / lambda_function.py SecretsManagerMongo RotationMultiUser
- 相依性：Pymongo 3.2

Amazon Redshift

Amazon Redshift 單一使用者

- 範本名稱： SecretsManagerRedshiftRotationSingleUser
- 輪換策略：the section called “單一使用者”。
- 預期的**`SecretString`**結構：the section called “Amazon Redshift 機密結構”或the section called “Amazon Redshift 無服務器秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主// lambda_function.py SecretsManagerRedshiftRotationSingleUser
- 相依PyGre性：

Amazon Redshift 交替使用者

- 範本名稱： SecretsManagerRedshiftRotationMultiUser
- 輪換策略：the section called “交替使用者”。
- 預期的**`SecretString`**結構：the section called “Amazon Redshift 機密結構”或the section called “Amazon Redshift 無服務器秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主// lambda_function.py SecretsManagerRedshiftRotationMultiUser
- 相依PyGre性：

Amazon ElastiCache

若要使用此範本，請參閱 Amazon 使用 ElastiCache 者指南中的自動輪替使用者密碼。

- 範本名稱： SecretsManagerElasticacheUserRotation
- 預期的 **`SecretString`** 結構：the section called “Amazon ElastiCache 秘密結構”。
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主// lambda_function.py SecretsManagerElasticacheUserRotation

其他類型的秘密

Secrets Manager 提供此範本做為起點，供您為任何類型的秘密建立輪換函數。

- 範本名稱： SecretsManagerRotationTemplate
- 源代碼:https://github.com/aws-samples/ aws-secrets-manager-rotation-羊肉/樹/主// lambda_function.py SecretsManagerRotationTemplate

撰寫函數時，請謹慎納入偵錯或記錄陳述式。這些陳述式可能會導致函數中的資訊寫入 Amazon CloudWatch，因此您需要確保記錄檔不包含在開發期間收集的任何敏感資訊。

為了安全起見，Secrets Manager 僅允許 Lambda 輪換函數直接輪換機密。輪換函數無法呼叫第二個 Lambda 函數來輪換機密。

如需日誌陳述式的範例，請參閱the section called “輪換函數範本”原始程式碼。

如果您使用外部二進位檔案和程式庫 (例如連線至資源)，則需要管理修補並保留它們 up-to-date。

如需偵錯建議，請參閱 Testing and debugging serverless applications (測試與偵錯無伺服器應用程式)。

輪換秘密有四個步驟，分別對應 Lambda 輪換函數的下列四個方法。

方法

- create_secret
- set_secret
- test_secret
- finish_secret

create_secret

在 create_secret 中，您會先使用傳入的 ClientRequestToken 呼叫 get_secret_value，確認是否有秘密。如果沒有秘密，您可以使用 create_secret 和字符，建立新的秘密當作 VersionId。您便可使用 get_random_password，產生新的秘密值。您必須確定新的秘密值只包含對資料庫或服務有效的字元。使用 ExcludeCharacters 參數排除字元。呼叫 put_secret_value，將其與預備標籤 AWSPENDING 一起儲存。將新的秘密值儲存在 AWSPENDING 中，有助於確保等冪性。如果輪換因任何原因而失敗，您可以在後續呼叫中參考該秘密值。請參閱 How do I make my Lambda function idempotent (如何讓 Lambda 函數等冪)。

在測試函數時，請使用 AWS CLI 查看版本階段：調用describe-secret並查看VersionIdsToStages。

set_secret

在 set_secret 中，您會變更資料庫或服務中的憑證，以符合 AWSPENDING 版本秘密中的新秘密值。

如果您將陳述式傳遞至解譯陳述式的服務 (例如資料庫)，請使用查詢參數化。如需詳細資訊，請參閱 OWASP 網站上的 Query Parameterization Cheat Sheet (查詢參數化速查表)。

輪換函數是具有特殊權限的代理人，有權存取和修改 Secrets Manager 秘密和目標資源中的客戶憑證。為了防止潛在的混淆代理人攻擊，您必須確保攻擊者無法使用該函數存取其他資源。在您更新憑證之前：

- 查看 AWSCURRENT 版本秘密中的憑證是否有效。如果 AWSCURRENT 憑證無效，請放棄輪換嘗試。
- 查看 AWSCURRENT 和 AWSPENDING 秘密值是否針對相同的資源。對於使用者名稱和密碼，請查看 AWSCURRENT 和 AWSPENDING 使用者名稱是否相同。

- 檢查目的地服務資源是否相同。對於資料庫，請查看 AWSCURRENT 和 AWSPENDING 主機名稱是否相同。

`test_secret`

在 `test_secret` 中，您會使用 AWSPENDING 版本的秘密存取資料庫或服務，以進行測試。

`finish_secret`

在 `finish_secret` 中，您會使用 `update_secret_version_stage`，將預備標籤 AWSCURRENT 從先前的秘密版本移至新的秘密版本。Secrets Manager 會自動將 AWSPREVIOUS 預備標籤新增至先前版本，以便您保留上一個已知良好的秘密版本。

Secret Manager 輪換中的排程表達式

當您開啟自動輪換時，您可以使用 cron() 或者 rate() 表達式來設定輪換秘密的排程。使用 rate 表達式，您可以建立一個在幾小時或幾天間隔內重複的輪換排程。使用 cron 表達式，您可以建立比輪換間隔更詳細的輪換排程。Secrets Manager 輪換排程使用 UTC 時區。您可以每四小時輪換一次機密。Secrets Manager 在輪換時段的任何時間輪換您的機密。

若要開啟輪換，請參閱：

- the section called “資料庫秘密的自動輪換 (主控台)”
- the section called “自動輪換 (主控台)”
- the section called “自動輪換 (AWS CLI)”

Rate 運算式

Secrets Manager rate 表達式的格式如下，*`Value`* 是正整數，*`Unit`* 可以是 `hour`、`hours`、`day` 或 `days`：

```
rate(Value Unit)
```

您可以每四小時輪換機密一次。範例：

- `rate(4 hours)` 表示每四小時輪換機密一次。
- `rate(1 day)` 表示每天輪換機密一次。
- `rate(10 days)` 表示每 10 天輪換機密一次。

對於以小時為單位的速率，預設輪換時段在午夜開始，並在一小時後關閉。您可以設定 Window duration (時段持續時間) 以變更輪換時段。輪換時段不得延伸到下一個輪換時段。檢查此情況的一種方法是確認輪換時段小於或等於輪換之間的小時數。

對於以天為單位的速率，預設輪換時段在午夜開始，並在一天結束時關閉。您可以設定 Window duration (時段持續時間) 以變更輪換時段。輪換時段不得延伸到第二天 (以 UTC 為準)。檢查此情況的一種方法是確認開始時刻加上時段持續時間小於或等於 24 小時。

Cron 表達式

Cron 表達式格式如下：

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

包含小時增量的 cron 表達式會每天重設。例如，`cron(0 4/12 * * ? *)` 表示凌晨 4:00 和下午 4:00，然後是第二天凌晨 4:00 和下午 4:00。Secrets Manager 輪換排程使用 UTC 時區。

對於以小時為單位的排程，預設輪換時段在一小時後關閉。您可以設定 Window duration (時段持續時間) 以變更輪換時段。輪換時段不得進入下一個輪換時段。您可以每四小時輪換一次秘密。

範例排程	表達式
每隔八小時在午夜開始。	`cron(0 /8 * * ? *)`
每隔八小時在上午 8:00 開始。	`cron(0 8/8 * * ? *)`
每隔十小時在凌晨 2:00 開始。 輪換時段將在 2:00、12:00 和 22:00 開始，然後在第二天 2:00、12:00 和 22:00 開始。	`cron(0 2/10 * * ? *)`
每天上午 10:00。	`cron(0 10 * * ? *)`
每週六下午 6:00。	`cron(0 18 ? * SAT *)`
每個月第一天上午 8:00。	`cron(0 8 1 * ? *)`
每三個月第一個星期天的凌晨 1:00。	`cron(0 1 ? 1/3 SUN#1 *)`
每個月最後一天下午 5:00。	`cron(0 17 L * ? *)`

範例排程	表達式
每週一至週五上午 8:00。	`cron(0 8 ? * MON-FRI *)`
每月的第一天和第十五天下午 4:00。	`cron(0 16 1,15 * ? *)`
每個月的第一個週日午夜。	`cron(0 0 ? * SUN#1 *)`

Secrets Manager 中的 cron 表達式要求

Secrets Manager 對您可以用於 cron 表達式的內容有一些限制。Secrets Manager 的 cron 表達式在分鐘欄位中必須有 0，因為 Secrets Manager 輪換時段會在整點時刻開始。在年份欄位中必須有 *，因為 Secrets Manager 不支援相隔一年以上的輪換排程。下列資料表顯示您可以使用的選項。

欄位	Values (數值)	Wildcards (萬用字元)
分鐘	必須為 0	無
小時	0–23	使用 / (正斜線) 指定增量。例如，2/10 表示從凌晨 2:00 開始每隔 10 小時一次。您可以每四小時輪換一次秘密。
月中的日	1–31	使用 , (逗號) 來包含其他值。例如，1,15 表示一個月的第 1 天和第 15 天。 使用 - (破折號) 指定範圍。例如，1–15 表示一個月的第 1 天至第 15 天。 使用 * (星號) 來包含欄位中的所有值。例如，* 表示一個月的每一天。 ? (問號) 萬用字元用於表示不限定任何一個。您無法在同一個 cron 表達式中指定 Day-

欄位	Values (數值)	Wildcards (萬用字元)
		of-month 和 Day-of-week 欄位。如果您在其中一個欄位指定了數值，就必須在另一個欄位中使用 ? (問號)。 使用 / (正斜線) 指定增量。例如，1/2 表示每兩天從第 1 天開始，換句話說，第 1 天、第 3 天、5 天等。 使用 L 指定一個月的最後一天。 使用 ***DAY***L 指定一個月的最後一個命名日期。例如，SUNL 表示一個月的最後一個週日。
月	1-12 或 JAN-DEC	使用 , (逗號) 來包含其他值。例如，JAN,APR,JUL,OCT 表示 1 月、4 月、7 月 和 10 月。 使用 - (破折號) 指定範圍。例如，1–3 表示一年的第 1 個月至第 3 個月。 使用 * (星號) 來包含欄位中的所有值。例如，* 表示每個月。 使用 / (正斜線) 指定增量。例如，1/3 表示每第三個月，從第 1 個月開始，換句話說第 1 個月、第 4 個月、第 7 個月和第 10 個月。

欄位	Values (數值)	Wildcards (萬用字元)
週中的日	1-7 或 SUN-SAT	使用 # 指定一個月內的星期幾。例如：TUE#3 表示當月的第三個週二。 使用 , (逗號) 來包含其他值。例如，1,4 表示一週的第 1 天和第 4 天。 使用 - (破折號) 指定範圍。例如，1–4 表示一週的第 1 天至第 4 天。 使用 * (星號) 來包含欄位中的所有值。例如，* 表示一週的每一天。 ? (問號) 萬用字元用於表示不限定任何一個。您無法在同一個 cron 表達式中指定 Day-of-month 和 Day-of-week 欄位。如果您在其中一個欄位指定了數值，就必須在另一個欄位中使用 ? (問號)。 使用 / (正斜線) 指定增量。例如，1/2 表示一週的每隔一天，從第一天開始，則第 1 天、第 3 天、第 5 天和第 7 天。 使用 L 指定一週的最後一天。
年	必須為 *	無

疑難排解 AWS Secrets Manager 旋

對許多服務而言，Secrets Manager 使用 Lambda 函數來輪換秘密。如需詳細資訊，請參閱 the section called “輪換的運作方式”。Lambda 輪換函數會與秘密適用的資料庫或服務，以及 Secrets Manager 互動。當輪換無法按照您預期的方式工作時，您應該首先檢查 CloudWatch 日誌。

> **Note**
>
> 某些服務可以為您管理秘密，包括管理自動輪換。如需詳細資訊，請參閱 the section called “受管輪換”。

若要檢視 Lambda 函數的 CloudWatch 記錄

1. 前往以下位置開啟機密管理員控制台：https://console.aws.amazon.com/secretsmanager/。
2. 選擇您的秘密，然後在詳細資訊頁面的 Rotation configuration (輪換設定) 之下，選擇 Lambda 輪換函數。Lambda 主控台開啟。
3. 在 [監控] 索引標籤上，選擇 [記錄檔]，然後選擇 [檢視記錄於] CloudWatch。

   CloudWatch 控制台打開並顯示您的功能的日誌。

解譯日誌

- 「在環境變數中找到憑證」之後沒有活動
- "createSecret" 之後沒有任何活動
- 錯誤：「不允許存取 KMS」
- 錯誤：「秘密 JSON 缺少金鑰」
- 錯誤：「setSecret：無法登入資料庫」
- 錯誤：「無法匯入模組 'lambda_function'」
- 將現有的輪換函數從 Python 3.7 升級至 3.9

「在環境變數中找到憑證」之後沒有活動

如果「在環境變數中找到憑證」之後沒有活動，且任務持續時間很長 (例如預設 Lambda 逾時為 30000 毫秒)，則 Lambda 函數可能會在嘗試連線 Secrets Manager 端點時逾時。

您的 Lambda 輪換函數必須能夠存取 Secrets Manager 服務端點。如果您的 Lambda 函數可以存取網際網路，那麼您可以使用公有端點。若要尋找端點，請參閱 the section called “Secrets Manager 端點”。

如果 Lambda 函數在無法存取網際網路的 VPC 中執行，建議您在 VPC 中設定 Secrets Manager 服務私有端點。然後，您的 VPC 可以攔截發送到公有區域端點的請求，並將其重新導向到私有端點。如需詳細資訊，請參閱 VPC 端點。

或者，您可以啟用 Lambda 函數來存取 Secrets Manager 公有端點，方法是將 NAT 閘道或網際網路閘道新增至您的 VPC，以便來自 VPC 的流量到達公有端點。這將使 VPC 暴露在較高的風險下，因為 IP 地址 (用於閘道) 可能會遭到來自公有網際網路的攻擊。

"createSecret" 之後沒有任何活動

以下是可能導致輪換在 CreateSecret 之後停止的問題：

VPC 網路 ACL 不允許傳入和傳出 HTTPS 流量。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用網路 ACL 控制子網路的流量。

Lambda 函數逾時組態太短，無法執行任務。

如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的設定 Lambda 函數選項。

Secrets Manager VPC 端點不允許在所指派安全群組的輸入上使用 VPC CIDR。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用安全群組控制到資源的流量。

Secrets Manager VPC 端點政策不允許 Lambda 使用 VPC 端點。

如需詳細資訊，請參閱 VPC 端點。

此機密使用交替使用者輪換，超級使用者機密由 Amazon RDS 管理，而 Lambda 函數無法存取 RDS API。

對於超級使用者密碼由其他 AWS 服務管理的交替使用者輪換，Lambda 輪換函數必須能夠呼叫服務端點以取得資料庫連線資訊。建議您為資料庫服務設定 VPC 端點。如需詳細資訊，請參閱：

- 在《Amazon RDS 使用者指南》中的 Amazon RDS API 和介面 VPC 端點。
- 在 Amazon Redshift 管理指南中使用 VPC 端點。

錯誤：「不允許存取 KMS」

如果顯示 `ClientError: An error occurred (AccessDeniedException) when calling the GetSecretValue operation: Access to KMS is not allowed`，則輪換函數沒有使用用於加密秘密的 KMS 金鑰來解密秘密的許可。許可政策可能包含一個將加密內容限制為特定秘密的條件。如需有關必要許可的資訊，請參閱the section called “客戶管理金鑰的政策陳述式”。

錯誤：「秘密 JSON 缺少金鑰」

Lambda 輪換函數需要秘密值位於特定的 JSON 結構。如果您看到此錯誤，JSON 可能缺少輪換函數先前嘗試存取的金鑰。如需每種秘密類型的 JSON 結構相關資訊，請參閱the section called “機密的 JSON 結構”。

錯誤：「setSecret：無法登入資料庫」

以下是可能導致此錯誤的問題：

輪換函數無法存取資料庫。

- 如果任務持續時間很長 (例如超過 5000 毫秒)，Lambda 輪換函數可能無法透過網路存取資料庫。
- 如果資料庫或服務正在 VPC 中的 Amazon EC2 執行個體上執行，建議您將 Lambda 函數設定為在相同的 VPC 中執行。然後輪換函數就能直接與您的服務進行通訊。如需詳細資訊，請參閱設定 VPC 存取 。
- 若要允許 Lambda 函數存取資料庫或服務，您必須確定附加至 Lambda 輪換函數的安全群組允許連至資料庫或服務的傳出連線。此外，您必須確定附加至資料庫或服務的安全群組允許來自 Lambda 輪換函數的傳入連線。

秘密中的憑證不正確。

- 如果任務持續時間很短，Lambda 輪換函數可能無法使用秘密中的憑證來驗證。使用 AWS CLI 指令以密碼`AWSCURRENT`和`AWSPREVIOUS`版本中的資訊手動登入，以檢查認證`get-secret-value`。

資料庫使用 **`scram-sha-256`** 來加密機密。

- 如果資料庫是 Aurora PostgreSQL 13 版或更新版本，並且使用 `scram-sha-256` 加密密碼，但輪換函數使用不支援 `scram-sha-256` 的 `libpq` 9 版或更舊版本，則輪換函數無法連接至資料庫。

判斷哪些資料庫使用者使用 **scram-sha-256** 加密

- 請參閱 SCRAM Authentication in RDS for PostgreSQL 13 (RDS for PostgreSQL 13 中的 SCRAM 身分驗證) 中的 Checking for users with non-SCRAM passwords (檢查具有非 SCRAM 密碼的使用者)。

判斷輪換函數使用的 **libpq** 版本

1. 在 Linux 電腦上的 Lambda 主控台上，導覽至輪換函數並下載部署套件。將 zip 檔案解壓縮到工作目錄。
2. 在命令列，在工作目錄中執行：

   ```
   readelf -a libpq.so.5 | grep RUNPATH
   ```

3. 如果您看到字串 *PostgreSQL-9.4.x* 或任何小於 10 的主要版本，則輪換函數不支援 scram-sha-256。
   - 不支援 scram-sha-256 的輪換函數的輸出：

     ```
     0x000000000000001d (RUNPATH) Library runpath: [/
     local/p4clients/pkgbuild-a1b2c/workspace/build/
     PostgreSQL/PostgreSQL-9.4.x_client_only.123456.0/AL2_x86_64/
     DEV.STD.PTHREAD/build/private/tmp/brazil-path/build.libfarm/lib:/
     local/p4clients/pkgbuild-a1b2c/workspace/src/PostgreSQL/build/
     private/install/lib]
     ```

   - 支援 scram-sha-256 的輪換函數的輸出：

     ```
     0x000000000000001d (RUNPATH) Library runpath: [/
     local/p4clients/pkgbuild-a1b2c/workspace/build/
     PostgreSQL/PostgreSQL-10.x_client_only.123456.0/AL2_x86_64/
     DEV.STD.PTHREAD/build/private/tmp/brazil-path/build.libfarm/lib:/
     local/p4clients/pkgbuild-a1b2c/workspace/src/PostgreSQL/build/
     private/install/lib]
     ```

> **Note**
>
> 如果您在 2021 年 12 月 30 日之前已設定自動機密輪換，則輪換函數隨附不支援 scram-sha-256 的舊版 libpq。若要支援 scram-sha-256，您需要重新建立輪換函數。

資料庫需要 SSL/TLS 存取權限。

如果資料庫需要 SSL/TLS 連線，但輪換函數使用未加密的連線，則輪換函數無法連線至資料庫。Amazon RDS (Oracle 和 Db2 除外) 和 Amazon DocumentDB 的輪換函數自動使用 Secure Socket Layer (SSL) 或 Transport Layer Security (TLS) 來連線到您的資料庫 (如果有)。否則，他們會使用未加密的連線。

> **Note**
>
> 如果您在 2021 年 12 月 20 日之前設定了自動秘密輪換，則輪換功能可能基於不支援 SSL/TLS 的舊範本。若要支援使用 SSL/TLS 的連線，您需要重建輪換函數。

確定輪換函數的建立時間

1. 在 Secrets Manager 主控台 https://console.aws.amazon.com/secretsmanager/，開啟您的秘密。在 Rotation configuration (輪換組態) 區段，在 Lambda rotation function (Lambda 輪換函數) 下，您會看到 Lambda function ARN (Lambda 函數 ARN)，例如 `arn:aws:lambda:aws-region:123456789012:function:SecretsManagerMyRotationFunction` 。在此範例 `SecretsManagerMyRotationFunction` 中，從 ARN 末尾複製函數名稱。
2. 在 AWS Lambda 主控台 https://console.aws.amazon.com/lambda/ 的 [函數] 下，將您的 Lambda 函數名稱貼到搜尋方塊中，選擇 [輸入]，然後選擇 Lambda 函數。
3. 在函數詳細資訊頁面上，Configuration (組態) 欄標中，Tags (標籤) 下，複製 aws:cloudformation:stack-name 索引鍵旁的值。
4. 在 AWS CloudFormation 主控台 https://console.aws.amazon.com/cloudformation 的「堆疊」下，將索引鍵值貼到搜尋方塊中，然後選擇「輸入」。
5. 篩選堆疊清單，以便只顯示建立 Lambda 輪換函數的堆疊。在 Created date (建立日期) 欄中，檢視堆疊建立的日期。這是建立 Lambda 輪換函數的日期。

錯誤：「無法匯入模組 'lambda_function'」

如果您執行的是先前的 Lambda 函數，由於函數版本已從 Python 3.7 自動升級到較新版本的 Python，則可能會收到此錯誤。若要解決錯誤，您可以將 Lambda 函數版本變更回 Python 3.7，然後 the section called “將現有的輪換函數從 Python 3.7 升級至 3.9”。如需詳細資訊，請參閱 AWS re:Post 中的為什麼我的 Secrets Manager Lambda 函數輪換失敗並顯示「找不到 pg 模組」錯誤？文章。

將現有的輪換函數從 Python 3.7 升級至 3.9

在 2022 年 11 月之前建立的一些輪換函數使用 Python 3.7。 AWS 開發套件於二零二三年十二月停止支援 Python 3.7。如需詳細資訊，請參閱適用於 AWS 開發套件和工具的 Python 支援原則更新。若要切換為使用 Python 3.9 的新輪換函數，您可以將執行期屬性新增至現有的輪換函數或重新建立輪換函數。

若要查看哪些 Lambda 輪換函數使用 Python 3.7

1. 請登入 AWS Management Console 並開啟 AWS Lambda 主控台，網址為 https://console.aws.amazon.com/lambda/。
2. 在函數清單中，篩選 **SecretsManager**。
3. 在篩選出的函數清單中，在執行期之下，尋找 Python 3.7。

若要升級至 Python 3.9：

- 選項 1：使用重新創建旋轉功能 AWS CloudFormation
- 選項 2：使用更新現有旋轉函數的運行時 AWS CloudFormation
- 選項 3：對於 AWS CDK 用戶，請升級 CDK 庫

選項 1：使用重新創建旋轉功能 AWS CloudFormation

當您使用 Secrets Manager 主控台開啟輪換時，Secrets Manager 會用 AWS CloudFormation 來建立必要的資源，包括 Lambda 輪換函數。如果您使用主控台開啟旋轉功能，或是使用 AWS CloudFormation 堆疊建立旋轉功能，您可以使用相同的 AWS CloudFormation 堆疊來重新建立具有新名稱的旋轉功能。新函數使用了較新版本的 Python。

若要尋找建立旋轉函數的 AWS CloudFormation 堆疊

- 在 Lambda 函數的詳細資訊頁面的組態分頁上，選擇標籤。檢視 aws:cloudformation:stack-id 旁邊的 ARN。

  堆疊名稱嵌入在 ARN 中，如以下範例所示。

  - ARN：`arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`

- 堆疊名稱：**SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda**

若要重新建立輪換函數 (AWS CloudFormation)

1. 在中 AWS CloudFormation，依名稱搜尋堆疊，然後選擇 [更新]。

   如果出現建議您更新根堆疊的對話方塊，請選擇前往根堆疊，然後選擇更新。
2. 在更新堆疊頁面上，選擇在設計工具中編輯範本，然後選擇在設計工具中檢視。
3. 在設計工具中，在範本程式碼的 SecretRotationScheduleHostedRotationLambda 中，將 "functionName": "SecretsManagerTestRotationRDS" 換成新函數名稱，例如在 JSON 中，**"functionName": "SecretsManagerTestRotationRDSupdated"**
4. 繼續執行 AWS CloudFormation 堆疊工作流程，然後選擇「提交」。

選項 2：使用更新現有旋轉函數的運行時 AWS CloudFormation

當您使用 Secrets Manager 主控台開啟輪換時，Secrets Manager 會用 AWS CloudFormation 來建立必要的資源，包括 Lambda 輪換函數。如果您使用主控台開啟旋轉功能，或者您使用 AWS CloudFormation 堆疊建立了旋轉函式，則可以使用相同的 AWS CloudFormation 堆疊來更新旋轉函式的執行階段。

若要尋找建立旋轉函數的 AWS CloudFormation 堆疊

- 在 Lambda 函數的詳細資訊頁面的組態分頁上，選擇標籤。檢視 aws:cloudformation:stack-id 旁邊的 ARN。

  堆疊名稱嵌入在 ARN 中，如以下範例所示。

  - ARN：arn:aws:cloudformation:us-west-2:408736277230:stack/**SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda**-3CUDHZMDMBO8/79fc9050-2eef-11ed-
  - 堆疊名稱：**SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda**

若要更新輪換函數的執行期 (AWS CloudFormation)

1. 在中 AWS CloudFormation，依名稱搜尋堆疊，然後選擇 [更新]。

如果出現建議您更新根堆疊的對話方塊，請選擇前往根堆疊，然後選擇更新。

2. 在更新堆疊頁面上，選擇在設計工具中編輯範本，然後選擇在設計工具中檢視。
3. 在設計器中，在範本 JSON 中，針對`SecretRotationScheduleHostedRotationLambda`、下`Properties`、下`Parameters`、新增 **`"runtime": "python3.9"`**
4. 繼續執行 AWS CloudFormation 堆疊工作流程，然後選擇「提交」。

選項 3：對於 AWS CDK 用戶，請升級 CDK 庫

如果您使用 v2.94.0 AWS CDK 之前的版本來設定密碼的輪換，則可以升級至 v2.94.0 或更新版本來更新 Lambda 函數。如需詳細資訊，請參閱《AWS Cloud Development Kit (AWS CDK) v2 開發人員指南》。

由其他 AWS 服務管理的 AWS Secrets Manager 秘密

許多 AWS 服務會儲存並使用 AWS Secrets Manager 中的秘密。在某些情況下，這些秘密是受管秘密，也就是說，建立秘密的服務有助於管理秘密。例如，有些受管秘密會包含受管輪換，因此您不必自行設定輪換。管理服務也可能會限制您在沒有復原期的情況下更新或刪除秘密，因為管理服務取決於秘密，因此這樣有助於避免外洩情況發生。

受管秘密的命名慣例是在名稱中加入管理服務 ID ，以便識別。

```
Secret name: ServiceID!MySecret
Secret ARN : arn:aws:us-east-1:ServiceID!MySecret-a1b2c3
```

管理秘密的服務之 ID

- `appflow` – the section called “Amazon AppFlow”
- `databrew` – the section called “AWS Glue DataBrew”
- `datasync` – the section called “AWS DataSync”
- `directconnect` – the section called “AWS Direct Connect”
- `ecs-sc` – the section called “Amazon Elastic Container Service”
- `events` – the section called “Amazon EventBridge”
- `marketplace-deployment` – the section called “AWS Marketplace”
- `opsworks-cm` – the section called “AWS OpsWorks for Chef Automate”
- `rds` – the section called “Amazon RDS 和 Aurora”
- `redshift` – the section called “Amazon Redshift”
- `sqlworkbench` – the section called “Amazon Redshift 查詢編輯器第 2 版”

若要尋找由其他 AWS 服務管理的秘密，請參閱《尋找受管秘密》。

如需使用密碼之服務的完整清單，請參閱the section called “AWS 使用 AWS Secrets Manager 機密的服務”。

Amazon AppFlow

在 Amazon 中 AppFlow，當您將 SaaS 應用程式設定為來源或目的地時，就會建立連線。這包括連線至 SaaS 應用程式所需的資訊，如身分驗證權杖、使用者名稱和密碼。Amazon 將您的連接數據

AppFlow 存儲在帶有前綴的秘密秘密中的秘密管理器中`appflow`。存儲秘密的費用已包含在 Amazon 的費用中 AppFlow。如需詳細資訊，請參閱 Amazon AppFlow 使用者指南 AppFlow中的 Amazon 中的資料保護。

AWS Glue DataBrew

AWS Glue DataBrew 提供了 `DETERMINISTIC_DECRYPT`、`DETERMINISTIC_ENCRYPT` 以及 `CRYPTOGRAPHIC_HASH` 配方步驟來對資料集中的個人身分識別資訊 (PII) 執行轉換，這些資訊使用 Secrets Manager 秘密中儲存的加密金鑰。如果您使用 DataBrew 預設密碼來儲存加密金鑰，請使用前置詞 DataBrew 建立受管理的密碼`databrew`。存儲秘密的費用已包含在使用費用中 DataBrew。

AWS DataSync

若要收集內部部署儲存系統的相關資訊，AWS DataSyncDiscovery 會使用儲存區系統管理介面的認證。 DataSync 將這些認證儲存在具有前置詞的 Secret 管理員管理密碼中`datasync`。您需要為該秘密付費。如需詳細資訊，請參閱AWS DataSync使用指南中的將內部部署儲存系統新增至 DataSync 探索。

AWS Direct Connect

AWS Direct Connect 將連線關聯金鑰名稱和連線關聯金鑰對 (CKN/CAK 對) 儲存在字首為 `directconnect` 的受管秘密中。秘密的成本包含在 AWS Direct Connect 的費用中。若要更新秘密，必須使用 AWS Direct Connect 而不是 Secrets Manager。如需詳細資訊，請參閱《AWS Direct Connect 使用者指南》中的將 MACsec CKN/CAK 與 LAG 產生關聯。

Amazon Elastic Container Service

當您使用 Amazon ECS 服務 Connect 時，Amazon ECS 會使用 Secrets Manager 密碼來存放 AWS Private Certificate Authority TLS 憑證。Amazon ECS 的費用已包含儲存密碼的費用。要更新密碼，您必須使用 Amazon ECS 而不是 Secrets Manager。如需詳細資訊，請參閱 Amazon 彈性容器服務開發人員指南中的 TLS 與服務 Connect。

Amazon EventBridge

當您建立 Amazon EventBridge API 目的地時，會將其連線 EventBridge 存放在具有前置詞的秘密管理秘密中`events`。存放秘密的成本包含在使用 API 目的地的費用中。若要更新秘密，必須使用

EventBridge 而不是 Secrets Manager。如需詳細資訊，請參閱 Amazon EventBridge 使用者指南中的 API 目的地。

AWS Marketplace

當您使用 AWS Marketplace 快速啟動時，AWS Marketplace 會將軟體與授權金鑰一起散佈。AWS Marketplace將授權金鑰儲存在您的帳戶中，做為秘密管理員管理的密碼。秘密的成本儲存在 AWS Marketplace 的費用中。若要更新秘密，必須使用 AWS Marketplace 而不是 Secrets Manager。如需詳細資訊，請參閱AWS Marketplace《賣方指南》中的使用進行快速組態設定。

AWS OpsWorks for Chef Automate

當您在中建立新伺服器時AWS OpsWorks CM， OpsWorks CM 會將伺服器的資訊儲存在具有前置詞的 Secret Manager 管理密碼中`opsworks-cm`。秘密的成本包含在 AWS OpsWorks 的費用中。如需詳細資訊，請參閱《AWS OpsWorks 使用者指南》中的與 AWS Secrets Manager 整合。

Amazon RDS 和 Aurora

若要管理 Amazon Relational Database Service (Amazon RDS) (包括 Aurora) 的主要使用者憑證，Amazon RDS 可以為您建立受管秘密。您需要為該秘密付費。Amazon RDS 也會管理這些憑證的輪換。如需詳細資訊，請參閱《Amazon RDS 使用者指南》中的使用 Amazon RDS 和 AWS Secrets Manager 進行密碼管理和《Amazon Aurora 使用者指南》中的使用 Amazon Aurora 和 AWS Secrets Manager 進行密碼管理。

若為其他 Amazon RDS 憑證，請參閱 the section called “建立資料庫秘密”。

Amazon Redshift

若要管理 Amazon Redshift 的管理員登入資料，Amazon Redshift 可以為您建立受管密碼。您需要為該秘密付費。Amazon Redshift 也會管理這些憑證的輪換。如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的使用 AWS Secrets Manager 管理 Amazon Redshift 管理員密碼。

如需其他 Amazon Redshift 憑證的資訊，請參閱 the section called “建立資料庫秘密”。若要在呼叫資料 API 時使用憑證的秘密，請參閱《使用 Amazon Redshift 資料 API》。若要在使用 Amazon Redshift 查詢編輯器連線到資料庫時使用秘密，請參閱《Amazon Redshift 管理指南》中的使用查詢編輯器查詢資料庫和 the section called “Amazon Redshift 查詢編輯器第 2 版”。

Amazon Redshift 查詢編輯器第 2 版

當您使用 Amazon Redshift 查詢編輯器 v2 連線到資料庫時，Amazon Redshift 會將您的憑證存放在字首為 `sqlworkbench` 的 Secrets Manager 受管秘密中。存放秘密的成本包含在使用 Amazon Redshift 的費用中。若要更新秘密，必須使用 Amazon Redshift 而不是 Secrets Manager。如需詳細資訊，請參閱《Amazon Redshift 管理指南》中的使用查詢編輯器第 2 版。

使用 AWS Secrets Manager VPC 端點

我們建議您在無法從公有網際網路存取的私有網路上儘可能執行基礎設施。您可以建立介面 VPC 端點，以在您的 VPC 與 Secrets Manager 之間建立私有連線。介面端點採用 AWS PrivateLink 技術，這項技術可讓您在沒有網際網路閘道、NAT 裝置、VPN 連接或 AWS Direct Connect 連接的情況下私密地存取 Secrets Manager API。VPC 中的執行個體不需要公有 IP 地址，即能與 Secrets Manager API 通訊。您的 VPC 與 Secrets Manager 之間的流量都會保持在 AWS 網路的範圍內。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的介面 VPC 端點 (AWS PrivateLink)。

Secrets Manager 使用 Lambda 輪換函數輪換秘密時，例如包含資料庫憑證的秘密，Lambda 函數會同時向資料庫和 Secrets Manager 發出請求。在您使用主控台開啟自動輪換後，Secrets Manager 將在與資料庫相同的 VPC 中建立 Lambda 函數。我們建議您在相同的 VPC 中建立 Secrets Manager 端點，以便從 Lambda 輪換函數到 Secrets Manager 的請求保持在 Amazon 網路的範圍內。

如果您為該端點啟用私有 DNS，您可以使用其區域的預設 DNS 名稱 (例如 `secretsmanager.us-east-1.amazonaws.com`)，向 Secrets Manager 發出 API 請求。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的透過介面端點存取服務。

您可以透過在許可政策中包含條件，確保對 Secrets Manager 的請求都來自 VPC 存取。如需詳細資訊，請參閱 the section called “範例：許可和 VPC”。

您可以使用 AWS CloudTrail 日誌，藉此稽核透過 VPC 端點使用秘密的情況。

為 Secrets Manager 建立 VPC 端點

1. 請參閱 Amazon VPC 使用者指南中的建立介面端點。使用服務名稱：com.amazonaws.*region*.secretsmanager
2. 若要控制對端點的存取，請參閱使用端點策略控制對 VPC 端點的存取。

共用子網路

無法在與您共用的子網路中建立、描述、修改或刪除 VPC 端點。不過，可以在與您共用的子網路中使用 VPC 端點。如需 VPC 共享的相關資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的與其他帳戶共享 VPC。

在 AWS CloudFormation 中建立 AWS Secrets Manager 秘密

您可以使用 CloudFormation 範本中的 `AWS::SecretsManager::Secret` 資源，在 CloudFormation 堆疊中建立秘密，如 建立秘密 中所示。

若要為 Amazon RDS 或 Aurora 建立管理員密碼，建議您在 `AWS::RDS::DBCluster` 中使用 `ManageMasterUserPassword`。Amazon RDS 接著會為您建立秘密並管理輪換。如需更多詳細資訊，請參閱 受管輪換。

針對 Amazon Redshift 和 Amazon DocumentDB 憑證，首先使用 Secrets Manager 產生的密碼建立秘密，然後使用動態參考從秘密中擷取使用者名稱和密碼，當作新資料庫的憑證。接下來，使用 `AWS::SecretsManager::SecretTargetAttachment` 資源，將資料庫相關詳細資訊，新增至 Secrets Manager 必須輪換秘密的秘密。最後，若要開啟自動輪換，請使用 `AWS::SecretsManager::RotationSchedule` 資源並提供輪換函數和排程。請參閱以下範例：

- 使用 Amazon Redshift 憑證建立秘密
- 使用 Amazon DocumentDB 憑證建立秘密

使用 `AWS::SecretsManager::ResourcePolicy` 資源將資源政策連接至秘密。

如需使用 AWS CloudFormation 建立資源的資訊，請參閱《AWS CloudFormation 使用者指南》中的瞭解範本的基本知識。您也可以使用 AWS Cloud Development Kit (AWS CDK)。如需詳細資訊，請參閱 AWS Secrets Manager 建構程式庫。

使用 AWS CloudFormation 建立 AWS Secrets Manager 秘密

此範例會建立名為 **CloudFormationCreatedSecret-*a1b2c3d4e5f6*** 的秘密。秘密值是以下 JSON，是建立秘密時產生的 32 個字元的密碼。

```
{
    "password": "EXAMPLE-PASSWORD",
    "username": "saanvi"
}
```

此範例將使用以下 CloudFormation 資源：

- `AWS::SecretsManager::Secret`

如需使用 AWS CloudFormation 建立資源的資訊，請參閱《AWS CloudFormation 使用者指南》中的瞭解範本的基本知識。

JSON

```
{
    "Resources": {
        "CloudFormationCreatedSecret": {
            "Type": "AWS::SecretsManager::Secret",
            "Properties": {
                "Description": "Simple secret created by AWS CloudFormation.",
                "GenerateSecretString": {
                    "SecretStringTemplate": "{\"username\": \"saanvi\"}",
                    "GenerateStringKey": "password",
                    "PasswordLength": 32
                }
            }
        }
    }
}
```

YAML

```
Resources:
  CloudFormationCreatedSecret:
    Type: 'AWS::SecretsManager::Secret'
    Properties:
      Description: Simple secret created by AWS CloudFormation.
      GenerateSecretString:
        SecretStringTemplate: '{"username": "saanvi"}'
        GenerateStringKey: password
        PasswordLength: 32
```

使用 AWS CloudFormation 建立具有自動輪換功能的 AWS Secrets Manager 秘密及 Amazon RDS MySQL 資料庫執行個體

若要為 Amazon RDS 或 Aurora 建立管理員密碼，建議您使用 `ManageMasterUserPassword`，如 `AWS::RDS::DBCluster` 中的範例「為主要密碼建立 Secrets Manager 秘密」所示。Amazon RDS 接著會為您建立秘密並管理輪換。如需更多詳細資訊，請參閱 受管輪換。

創建一個 AWS Secrets Manager 秘密和一個 Amazon Redshift 集群 AWS CloudFormation

若要為 Amazon Redshift 建立管理員密碼，我們建議您使用AWS::Redshift::Cluster和AWS::RedshiftServerless::Namespace上的範例。

使用 AWS CloudFormation 建立 AWS Secrets Manager 秘密及 Amazon DocumentDB 執行個體

此範例會使用秘密中的憑證作為使用者和密碼，來建立秘密和 Amazon DocumentDB 執行個體。該秘密連接了資源型政策，可定義能夠存取秘密的人員。範本亦會從 輪換函數範本 建立 Lambda 輪換函數，並將秘密設定為每月第一天在上午 8:00 至 10:00 (UTC) 之間自動輪換。作為安全最佳實務，執行個體位於 Amazon VPC 中。

此範例將以下 CloudFormation 資源用於 Secrets Manager：

- AWS::SecretsManager::Secret
- AWS::SecretsManager::SecretTargetAttachment
- AWS::SecretsManager::RotationSchedule

如需使用 AWS CloudFormation 建立資源的資訊，請參閱《AWS CloudFormation 使用者指南》中的瞭解範本的基本知識。

JSON

```
{
   "AWSTemplateFormatVersion":"2010-09-09",
   "Transform":"AWS::SecretsManager-2020-07-23",
   "Resources":{
      "TestVPC":{
         "Type":"AWS::EC2::VPC",
         "Properties":{
            "CidrBlock":"10.0.0.0/16",
            "EnableDnsHostnames":true,
            "EnableDnsSupport":true
         }
      },
      "TestSubnet01":{
```

```
         "Type":"AWS::EC2::Subnet",
         "Properties":{
            "CidrBlock":"10.0.96.0/19",
            "AvailabilityZone":{
               "Fn::Select":[
                  "0",
                  {
                     "Fn::GetAZs":{
                        "Ref":"AWS::Region"
                     }
                  }
               ]
            },
            "VpcId":{
               "Ref":"TestVPC"
            }
         }
      },
      "TestSubnet02":{
         "Type":"AWS::EC2::Subnet",
         "Properties":{
            "CidrBlock":"10.0.128.0/19",
            "AvailabilityZone":{
               "Fn::Select":[
                  "1",
                  {
                     "Fn::GetAZs":{
                        "Ref":"AWS::Region"
                     }
                  }
               ]
            },
            "VpcId":{
               "Ref":"TestVPC"
            }
         }
      },
      "SecretsManagerVPCEndpoint":{
         "Type":"AWS::EC2::VPCEndpoint",
         "Properties":{
            "SubnetIds":[
               {
                  "Ref":"TestSubnet01"
               },
```

```
            {
               "Ref":"TestSubnet02"
            }
         ],
         "SecurityGroupIds":[
            {
               "Fn::GetAtt":[
                  "TestVPC",
                  "DefaultSecurityGroup"
               ]
            }
         ],
         "VpcEndpointType":"Interface",
         "ServiceName":{
            "Fn::Sub":"com.amazonaws.${AWS::Region}.secretsmanager"
         },
         "PrivateDnsEnabled":true,
         "VpcId":{
            "Ref":"TestVPC"
         }
      }
   },
   "MyDocDBClusterRotationSecret":{
      "Type":"AWS::SecretsManager::Secret",
      "Properties":{
         "GenerateSecretString":{
            "SecretStringTemplate":"{\"username\": \"someadmin\",\"ssl\": true}",
            "GenerateStringKey":"password",
            "PasswordLength":16,
            "ExcludeCharacters":"\"@/\\"
         },
         "Tags":[
            {
               "Key":"AppName",
               "Value":"MyApp"
            }
         ]
      }
   },
   "MyDocDBCluster":{
      "Type":"AWS::DocDB::DBCluster",
      "Properties":{
         "DBSubnetGroupName":{
            "Ref":"MyDBSubnetGroup"
```

```
            },
            "MasterUsername":{
               "Fn::Sub":"{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::username}}"
            },
            "MasterUserPassword":{
               "Fn::Sub":"{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::password}}"
            },
            "VpcSecurityGroupIds":[
               {
                  "Fn::GetAtt":[
                     "TestVPC",
                     "DefaultSecurityGroup"
                  ]
               }
            ]
         }
      },
      "DocDBInstance":{
         "Type":"AWS::DocDB::DBInstance",
         "Properties":{
            "DBClusterIdentifier":{
               "Ref":"MyDocDBCluster"
            },
            "DBInstanceClass":"db.r5.large"
         }
      },
      "MyDBSubnetGroup":{
         "Type":"AWS::DocDB::DBSubnetGroup",
         "Properties":{
            "DBSubnetGroupDescription":"",
            "SubnetIds":[
               {
                  "Ref":"TestSubnet01"
               },
               {
                  "Ref":"TestSubnet02"
               }
            ]
         }
      },
      "SecretDocDBClusterAttachment":{
         "Type":"AWS::SecretsManager::SecretTargetAttachment",
```

```
         "Properties":{
            "SecretId":{
               "Ref":"MyDocDBClusterRotationSecret"
            },
            "TargetId":{
               "Ref":"MyDocDBCluster"
            },
            "TargetType":"AWS::DocDB::DBCluster"
         }
      },
      "MySecretRotationSchedule":{
         "Type":"AWS::SecretsManager::RotationSchedule",
         "DependsOn":"SecretDocDBClusterAttachment",
         "Properties":{
            "SecretId":{
               "Ref":"MyDocDBClusterRotationSecret"
            },
            "HostedRotationLambda":{
               "RotationType":"MongoDBSingleUser",
               "RotationLambdaName":"MongoDBSingleUser",
               "VpcSecurityGroupIds":{
                  "Fn::GetAtt":[
                     "TestVPC",
                     "DefaultSecurityGroup"
                  ]
               },
               "VpcSubnetIds":{
                  "Fn::Join":[
                     ",",
                     [
                        {
                           "Ref":"TestSubnet01"
                        },
                        {
                           "Ref":"TestSubnet02"
                        }
                     ]
                  ]
               }
            },
            "RotationRules":{
              "Duration": "2h",
              "ScheduleExpression": "cron(0 8 1 * ? *)"
            }
```

```
         }
      }
   }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::SecretsManager-2020-07-23
Resources:
  TestVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  TestSubnet01:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.96.0/19
      AvailabilityZone:
        Fn::Select:
        - '0'
        - Fn::GetAZs:
            Ref: AWS::Region
      VpcId:
        Ref: TestVPC
  TestSubnet02:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.128.0/19
      AvailabilityZone:
        Fn::Select:
        - '1'
        - Fn::GetAZs:
            Ref: AWS::Region
      VpcId:
        Ref: TestVPC
  SecretsManagerVPCEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      SubnetIds:
      - Ref: TestSubnet01
```

```
      - Ref: TestSubnet02
      SecurityGroupIds:
      - Fn::GetAtt:
        - TestVPC
        - DefaultSecurityGroup
      VpcEndpointType: Interface
      ServiceName:
        Fn::Sub: com.amazonaws.${AWS::Region}.secretsmanager
      PrivateDnsEnabled: true
      VpcId:
        Ref: TestVPC
  MyDocDBClusterRotationSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      GenerateSecretString:
        SecretStringTemplate: '{\"username\": \"someadmin\",\"ssl\": true}'
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "\"@/\\"
      Tags:
      - Key: AppName
        Value: MyApp
  MyDocDBCluster:
    Type: AWS::DocDB::DBCluster
    Properties:
      DBSubnetGroupName:
        Ref: MyDBSubnetGroup
      MasterUsername:
        Fn::Sub: "{{resolve:secretsmanager:${MyDocDBClusterRotationSecret}::username}}"
      MasterUserPassword:
        Fn::Sub: "{{resolve:secretsmanager:${MyDocDBClusterRotationSecret}::password}}"
      VpcSecurityGroupIds:
      - Fn::GetAtt:
        - TestVPC
        - DefaultSecurityGroup
  DocDBInstance:
    Type: AWS::DocDB::DBInstance
    Properties:
      DBClusterIdentifier:
        Ref: MyDocDBCluster
      DBInstanceClass: db.r5.large
  MyDBSubnetGroup:
    Type: AWS::DocDB::DBSubnetGroup
    Properties:
```

```
      DBSubnetGroupDescription: ''
      SubnetIds:
      - Ref: TestSubnet01
      - Ref: TestSubnet02
  SecretDocDBClusterAttachment:
    Type: AWS::SecretsManager::SecretTargetAttachment
    Properties:
      SecretId:
        Ref: MyDocDBClusterRotationSecret
      TargetId:
        Ref: MyDocDBCluster
      TargetType: AWS::DocDB::DBCluster
  MySecretRotationSchedule:
    Type: AWS::SecretsManager::RotationSchedule
    DependsOn: SecretDocDBClusterAttachment
    Properties:
      SecretId:
        Ref: MyDocDBClusterRotationSecret
      HostedRotationLambda:
        RotationType: MongoDBSingleUser
        RotationLambdaName: MongoDBSingleUser
        VpcSecurityGroupIds:
          Fn::GetAtt:
          - TestVPC
          - DefaultSecurityGroup
        VpcSubnetIds:
          Fn::Join:
          - ","
          - - Ref: TestSubnet01
            - Ref: TestSubnet02
      RotationRules:
        Duration: 2h
        ScheduleExpression: 'cron(0 8 1 * ? *)'
```

Secrets Manager 使用 AWS CloudFormation 的方式

當您使用主控台開啟輪換功能，Secrets Manager 會使用 AWS CloudFormation 建立輪換所需的資源。如果您在該過程中建立新的輪換函數，AWS CloudFormation 會根據適當的 輪換函數範本 建立 AWS::Serverless::Function。接著，AWS CloudFormation 會設定 RotationSchedule，這會設定秘密的輪換函數和輪換規則。開啟自動輪換後，您可以在橫幅中選擇 View stack (檢視堆疊)，檢視 AWS CloudFormation 堆疊。

如需瞭解開啟自動輪換功能的相關資訊，請參閱輪換 秘密。

在 AWS Cloud Development Kit (AWS CDK) 中建立 AWS Secrets Manager 秘密

要在 CDK 應用程式中建立、管理和擷取秘密，可以使用 AWS Secrets Manager 建構程式庫，其中包含 `ResourcePolicy`、`RotationSchedule`、`Secret`、`SecretRotation` 和 `SecretTargetAttachment` 建構模組。

如需範例，請參閱：

- 建立秘密
- 匯入秘密
- 擷取秘密
- 授予秘密使用許可
- 輪換秘密
- 輪換資料庫秘密
- 將秘密複寫至其他區域

如需有關 CDK 的詳細資訊，請參閱《AWS Cloud Development Kit (AWS CDK) v2 開發人員指南》。

監控 AWS Secrets Manager 秘密

AWS 提供監控工具來監看 Secrets Manager 秘密，這些工具會在發生錯誤時回報，並自動適時採取動作：如果您需要針對任何未預期的使用或變更進行調查，則可使用此日誌，然後還原不想要的變更。您也可以針對不當使用秘密以及任何刪除秘密的嘗試設定自動檢查。

主題

使用 AWS CloudTrail 記錄 AWS Secrets Manager 事件

AWS CloudTrail 會將 Secrets Manager 的所有 API 呼叫記錄為事件，包括來自 Secrets Manager 主控台的呼叫，以及輪換和刪除秘密版本的其他事件。如需 Secrets Manager 記錄的日誌項目清單，請參閱 CloudTrail 條目。

您可以使用 CloudTrail 控制台查看過去 90 天的記錄事件。如需AWS帳戶中持續的事件記錄 (包括 Secrets Manager 的事件)，請建立追蹤，以便將日誌檔案 CloudTrail 傳送到 Amazon S3 儲存貯體。請參閱建立 AWS 帳戶的追蹤。您也可以設定 CloudTrail 為接收來自多個AWS 帳戶和的 CloudTrail 記錄檔AWS 區域。

您可以設定其他AWS服務，以進一步分析 CloudTrail 記錄中收集的資料並採取行動。請參閱與 CloudTrail 日誌的AWS服務整合。您也可以在將新的日誌檔 CloudTrail 發佈到 Amazon S3 儲存貯體時收到通知。請參閱設定的 Amazon SNS 通知 CloudTrail。

從 CloudTrail 記錄檔擷取 Secrets Manager 事件 (主控台)

1. 請在以下位置開啟 CloudTrail 主控台。 https://console.aws.amazon.com/cloudtrail/
2. 請確定主控台指向事件發生的區域。主控台只會顯示所選區域內發生的那些事件。請從主控台右上角的下拉式清單中選擇區域。
3. 在左側導覽窗格中，選擇 Event history (事件歷史記錄)。
4. 選擇 Filter (篩選) 標準和/或 Time range (時間範圍)，以協助找到您在尋找的事件。例如，若要查看所有 Secrets Manager 事件，請在 Select attribute (選取屬性) 中選擇 Event source (事件來源)。然後，針對輸入事件來源，選擇 **`secretsmanager.amazonaws.com`**。

5. 若要查看其他詳細資訊，請選擇事件旁的展開箭頭。若要查看所有可用的資訊，請選擇 View event (檢視事件)。

AWS CLI

Example 從 CloudTrail 記錄擷取 Secrets Manager 事件

下列 `lookup-events` 範例會查詢 Secrets Manager 事件。

```
aws cloudtrail lookup-events \
    --region us-east-1 \
    --lookup-attributes
 AttributeKey=EventSource,AttributeValue=secretsmanager.amazonaws.com
```

Secrets Manager 的 AWS CloudTrail 項目

AWS Secrets Manager 會將所有 Secrets Manager 操作的項目以及與輪換和刪除相關的其他事件寫入 AWS CloudTrail 日誌。如需如何對這些事件採取動作的相關資訊，請參閱 將 Secrets Manager 事件與 EventBridge。

日誌項目類型

- Secrets Manager 操作的日誌項目
- 要刪除的日誌項目
- 複寫的日誌項目
- 要輪換的日誌項目

Secrets Manager 操作的日誌項目

呼叫 Secrets Manager 操作所產生的事件具有 `"detail-type": ["AWS API Call via CloudTrail"]`。

> **Note**
>
> 在 2024 年 2 月之前，一些 Secrets Manager 操作報告了包含秘密 ARN 的「ARN」而不是「arn」的事件。如需詳細資訊，請參閱 AWS re:Post。

以下是當您或服務呼叫 Secrets Manager 透過 API、SDK 或 CLI 進行操作時產生的 CloudTrail 項目。

BatchGetSecretValue

由BatchGetSecretValue作業產生。如需瞭解擷取秘密的相關資訊，請參閱 擷取秘密。

CancelRotateSecret

由CancelRotateSecret作業產生。如需輪換的相關資訊，請參閱輪換 秘密。

CreateSecret

由CreateSecret作業產生。如需瞭解建立秘密的相關資訊，請參閱 建立和管理秘密。

DeleteResourcePolicy

由DeleteResourcePolicy作業產生。如需許可的相關資訊，請參閱身分驗證與存取控制。

DeleteSecret

由DeleteSecret作業產生。如需瞭解刪除秘密的相關資訊，請參閱 the section called “刪除秘密”。

DescribeSecret

由DescribeSecret作業產生。

GetRandomPassword

由GetRandomPassword作業產生。

GetResourcePolicy

由GetResourcePolicy作業產生。如需許可的相關資訊，請參閱身分驗證與存取控制。

GetSecretValue

由GetSecretValue和BatchGetSecretValue作業產生。如需瞭解擷取秘密的相關資訊，請參閱 擷取秘密。

ListSecrets

由ListSecrets作業產生。如需瞭解列出秘密的相關資訊，請參閱 the section called “查找秘密”。

ListSecretVersionIds

由ListSecretVersionIds作業產生。

PutResourcePolicy

由PutResourcePolicy作業產生。如需許可的相關資訊，請參閱身分驗證與存取控制。

PutSecretValue

由PutSecretValue作業產生。如需瞭解更新秘密的相關資訊，請參閱 the section called “修改秘密”。

RemoveRegionsFromReplication

由RemoveRegionsFromReplication作業產生。如需複寫秘密的相關資訊，請參閱the section called “將機密複寫到其他區域”。

ReplicateSecretToRegions

由ReplicateSecretToRegions作業產生。如需複寫秘密的相關資訊，請參閱the section called “將機密複寫到其他區域”。

RestoreSecret

由RestoreSecret作業產生。如需還原的相關資訊，請參閱the section called “還原秘密”。

RotateSecret

由RotateSecret作業產生。如需輪換的相關資訊，請參閱輪換 秘密。

StopReplicationToReplica

由StopReplicationToReplica作業產生。如需複寫秘密的相關資訊，請參閱the section called “將機密複寫到其他區域”。

TagResource

由TagResource作業產生。如需瞭解標記秘密的相關資訊，請參閱 the section called “標籤 秘密”。

UntagResource

由UntagResource作業產生。如需瞭解取消秘密標記的相關資訊，請參閱 the section called “標籤秘密”。

UpdateSecret

由UpdateSecret作業產生。如需瞭解更新秘密的相關資訊，請參閱 the section called “修改秘密”。

UpdateSecretVersionStage

由UpdateSecretVersionStage作業產生。如需版本階段的相關資訊，請參閱the section called “版本”。

ValidateResourcePolicy

由ValidateResourcePolicy作業產生。如需許可的相關資訊，請參閱身分驗證與存取控制。

要刪除的日誌項目

除了 Secrets Manager 操作的事件之外，Secrets Manager 還會產生下列與刪除相關的事件。這些事件具有 `"detail-type": ["AWS Service Event via CloudTrail"]`。

CancelSecretVersionDelete

由 Secrets Manager 服務產生。如果您在具有多個版本的秘密上呼叫 `DeleteSecret`，然後呼叫 `RestoreSecret`，Secrets Manager 會針對所還原的各個密碼版本記錄這個事件。如需還原的相關資訊，請參閱the section called “還原秘密”。

EndSecretVersionDelete

刪除機密版本時由 Secrets Manager 服務產生。如需詳細資訊，請參閱 the section called “刪除秘密”。

StartSecretVersionDelete

Secrets Manager 開始刪除機密版本時由 Secrets Manager 服務產生。如需瞭解刪除秘密的相關資訊，請參閱 the section called “刪除秘密”。

SecretVersionDeletion

Secrets Manager 刪除淘汰的秘密版本時由 Secrets Manager 服務所產生。如需詳細資訊，請參閱秘密版本。

複寫的日誌項目

除了 Secrets Manager 操作的事件之外，Secrets Manager 還會產生下列與複寫相關的事件。這些事件具有 `"detail-type": ["AWS Service Event via CloudTrail"]`。

ReplicationFailed

複寫失敗時由 Secrets Manager 服務產生。如需複寫秘密的相關資訊，請參閱the section called “將機密複寫到其他區域”。

ReplicationStarted

Secrets Manager 開始複寫秘密時由 Secrets Manager 服務產生。如需複寫秘密的相關資訊，請參閱the section called “將機密複寫到其他區域”。

ReplicationSucceeded

成功複寫秘密時由 Secrets Manager 服務產生。如需複寫秘密的相關資訊，請參閱the section called “將機密複寫到其他區域”。

要輪換的日誌項目

除了 Secrets Manager 操作的事件之外，Secrets Manager 還會產生下列與輪換相關的事件。這些事件具有 `"detail-type": ["AWS Service Event via CloudTrail"]`。

RotationStarted

Secrets Manager 開始輪換機密時由 Secrets Manager 服務產生。如需輪換的相關資訊，請參閱輪換 秘密。

RotationAbandoned

Secrets Manager 放棄嘗試輪換，並從現有的機密版本移除 `AWSPENDING` 標籤時由 Secrets Manager 服務產生。如果您在輪換期間建立新的秘密版本，Secrets Manager 便會放棄輪換。如需輪換的相關資訊，請參閱輪換 秘密。

RotationFailed

輪換失敗時由 Secrets Manager 服務產生。如需輪換的相關資訊，請參閱the section called “ 輪換疑難排解”。

RotationSucceeded

已成功輪換機密時由 Secrets Manager 服務產生。如需輪換的相關資訊，請參閱輪換 秘密。

TestRotationStarted

Secrets Manager 針對未排程立即輪換的機密開始測試輪換時由 Secrets Manager 服務產生。如需輪換的相關資訊，請參閱輪換 秘密。

TestRotationSucceeded

Secrets Manager 針對未排程立即輪換的機密成功測試輪換時由 Secrets Manager 服務產生。如需輪換的相關資訊，請參閱輪換 秘密。

TestRotationFailed

Secrets Manager 針對未排程立即輪換的機密測試輪換，且輪換失敗時由 Secrets Manager 服務產生。如需輪換的相關資訊，請參閱the section called “ 輪換疑難排解”。

與 Amazon 匹配 AWS Secrets Manager 活動 EventBridge

在 Amazon 中 EventBridge，您可以比對 CloudTrail 日誌項目中的 Secrets Manager 事件。您可以設定尋找這些事件的 EventBridge 規則，然後將新產生的事件傳送至目標以採取動作。如需 Secrets Manager 記 CloudTrail 錄的項目清單，請參閱CloudTrail 條目。如需設定的指示 EventBridge，請參閱《EventBridge 使用者指南》 EventBridge中的〈入門〉。

比對指定機密的所有變更

下列範例顯示符合密碼變更之記錄項目的 EventBridge 事件模式。

```
{
    "source": ["aws.secretsmanager"],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail": {
        "eventSource": ["secretsmanager.amazonaws.com"],
        "eventName": ["DeleteResourcePolicy", "PutResourcePolicy", "RotateSecret",
 "TagResource", "UntagResource", "UpdateSecret"],
        "responseElements": {
            "arn": ["arn:aws:secretsmanager:us-west-2:012345678901:secret:mySecret-
a1b2c3"]
        }
    }
}
```

機密值輪換時比對事件

下列範例顯示的 EventBridge 事件模式會與手動更新或自動輪換所發生之密碼值變更的 CloudTrail 記錄項目相符。由於其中一些事件來自 Secrets Manager 操作，一些由 Secrets Manager 服務產生，因此您必須包含兩者的 `detail-type`。

```
{
    "source": ["aws.secretsmanager"],
    "$or": [
        { "detail-type": ["AWS API Call via CloudTrail"] },
        { "detail-type": ["AWS Service Event via CloudTrail"] }
    ],
    "detail": {
        "eventSource": ["secretsmanager.amazonaws.com"],
        "eventName": ["PutSecretValue", "UpdateSecret", "RotationSucceeded"]
```

```
    }
}
```

AWS Secrets Manager使用 Amazon 監控 CloudWatch

您可以AWS Secrets Manager使用 Amazon 進行監控 CloudWatch，Amazon 會收集原始資料並將其處理為可讀且接近即時的指標。這些統計資料會保留 15 個月，以便您存取歷史資訊，並更清楚 Web 應用程式或服務的執行效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 Amazon CloudWatch 使用者指南。

對於 Secrets Manager，您可 CloudWatch 以使用在 API 的請求率或帳戶中的密碼數量達到特定閾值時提醒您。您也可以使用 CloudWatch 來監控 Secrets Manager 的估計費用。如需詳細資訊，請參閱建立帳單警示來監控預估的 AWS 費用。

主題

- Secrets Manager 指標和維度
- 建立警示以監視 Secrets Manager 指標
- Amazon CloudWatch Synthetics 金絲雀

Secrets Manager 指標和維度

AWS/SecretsManager 命名空間包含下列指標。

指標	描述
ResourceCount	您帳戶中的秘密數量，包括標記為刪除的秘密。指標每小時發佈一次。 單位：計數

Secrets Manager 指標的維度。

維度	描述
Service	包含該資源的 AWS 服務的名稱。針對 Secrets Manager，此維度的值為 Secrets Manager。

維度	描述
Type	正在報告的實體類型。針對 Secrets Manager，此維度的值為 Resource。
Resource	正在執行的資源類型。針對 Secrets Manager，此維度的值為 SecretCount 。
Class	無。

您可以使用 CloudWatch 指標監控的 Secrets Manager API 請求包括GetSecretValue、DescribeSecretListSecrets、和其他指標。若要尋找指標，請在 CloudWatch 主控台中選擇 [所有指標]，然後在搜尋方塊中輸入您的搜尋字詞**secrets**。

建立警示以監視 Secrets Manager 指標

您可以建立 CloudWatch 警示，在指標值變更時傳送 Amazon SNS 訊息，並導致警示狀態變更。警示會監看指定時段內的指標，並根據與多個時段內指定閾值相對的指標值來執行動作。警示只會呼叫持續狀態變更的動作。 CloudWatch 警示不會僅因為處於特定狀態而叫用動作；狀態必須已變更並維持指定數目的期間。

如需詳細資訊，請參閱使用 Amazon CloudWatch 警示和根據異常偵測建立警示。 CloudWatch

Amazon CloudWatch Synthetics 金絲雀

Amazon CloudWatch Synthetics 金絲雀是可設定的指令碼，可按排程執行，以監控您的端點和 API。Canary 遵循相同的路由並執行與客戶相同的動作，即使您的應用程式沒有任何客戶流量，也能持續驗證您的客戶體驗。

如需如何整合 Secrets Manager 的範例，請參閱將您的 Canary 與其他 AWS 服務整合。

使用 Amazon CloudWatch 監控排定刪除的 AWS Secrets Manager 秘密

您可以使用 AWS CloudTrail、Amazon CloudWatch Logs 及 Amazon Simple Notification Service (Amazon SNS) 的組合來建立警示，只要任何人嘗試存取待刪除的秘密，就會通知您。如果您收到警示的通知，您可以取消秘密的刪除，讓您有更多時間決定是否真的想要刪除該秘密。您調查到最後可能會還原秘密，因為您仍需要此秘密。或者，您可能需要為使用新秘密的使用者，提供最新的詳細資訊。

下列程序說明在對 `GetSecretValue` 操作提出請求而導致特定的錯誤訊息寫入 CloudTrail 日誌檔時，如何收到通知。可對秘密執行其他 API 操作，無需觸發警示。此 CloudWatch 警示會偵測可能指出使用過期登入資料之人員或應用程式的使用情況。

在開始這些程序前，您必須在想要監控 AWS Secrets Manager API 請求的 AWS 區域 和帳戶中開啟 CloudTrail。如需說明，請前往 AWS CloudTrail 使用者指南中的首次建立追蹤。

步驟 1：設定 CloudTrail 日誌檔案交付至 CloudWatch Logs

您必須將 CloudTrail 日誌檔設定為交付到 CloudWatch Logs。這樣做可讓 CloudWatch Logs 監控它們，以使 Secrets Manager API 請求擷取待刪除的秘密。

設定將 CloudTrail 日誌檔交付至 CloudWatch Logs

1. 透過 https://console.aws.amazon.com/cloudtrail/ 開啟 CloudTrail 主控台。
2. 在頂端導覽列中，選擇 AWS 區域來監控秘密。
3. 在左側導覽窗格中，選擇 Trails (追蹤)，然後選擇要為 CloudWatch 設定的追蹤名稱。
4. 在 Trails Configuration (追蹤組態) 頁面，向下捲動到 CloudWatch Logs 部分，然後選擇編輯圖示 (✎)。
5. 在 New or existing log group (新建或現有的日誌群組) 中，輸入日誌群組的名稱，例如 **`CloudTrail/MyCloudWatchLogGroup`**。
6. 對於 IAM role (IAM 角色)，您可以使用名為 CloudTrail_CloudWatchLogs_Role 的預設角色。此角色有預設的角色政策，內含將 CloudTrail 事件交付至日誌群組所需的許可。
7. 選擇 Continue (繼續) 來儲存您的組態。
8. 在 AWS CloudTrail 將與帳戶中 API 活動相關的 CloudTrail 事件交付至 CloudWatch Logs 日誌群組頁面，選擇 Allow (允許)。

步驟 2：建立 CloudWatch 警示

若要在 Secrets Manager `GetSecretValue` API 操作請求存取待刪除的秘密時收到通知，您必須建立 CloudWatch 警示並設定通知。

建立 CloudWatch 警示

1. 前往 https://console.aws.amazon.com/cloudwatch/ 登入 CloudWatch 主控台。
2. 在頂端導覽列中，選擇您要監控秘密的 AWS 區域。

3. 在左側導覽窗格中，選擇 Logs (日誌)。
4. 在 Log Groups (日誌群組) 中，選取您在之前程序中建立的日誌群組旁的核取方塊，例如 CloudTrail/MyCloudWatchLogGroup。然後，選擇 Create Metric Filter (建立指標篩選條件)。
5. 在 Filter Pattern (篩選條件模式) 中，請輸入或貼上下列內容：

   ```
   { $.eventName = "GetSecretValue" && $.errorMessage = "*secret because it was marked
    for deletion*" }
   ```

   選擇 Assign Metric (指派指標)。
6. 在 Create Metric Filter and Assign a Metric (建立指標篩選條件並指定指標) 頁面上，請執行下列動作：

   a. 針對 Metric Namespace (指標命名空間)，輸入 **`CloudTrailLogMetrics`**。
   b. 針對 Metric Name (指標名稱)，輸入 **`AttemptsToAccessDeletedSecrets`**。
   c. 選擇 Show advanced metric settings (顯示進階指標設定)，然後在必要時於 Metric Value (指標值) 中輸入 **1**。
   d. 選擇 Create Filter (建立篩選條件)。
7. 在篩選條件方塊中，選擇 Create Alarm (建立警示)。
8. 在 Create Alarm (建立警示) 視窗中，請執行下列動作：

   a. 在 Name (名稱) 輸入 **`AttemptsToAccessDeletedSecretsAlarm`**。
   b. 在 Whenever: (無論何時：) 的 is: (是) 中，選擇 **>=**，然後輸入 **1**。
   c. 在 Send notification to: (傳送通知給：) 欄位旁，執行以下其中一項：

      - 若要建立和使用新的 Amazon SNS 主題，請選擇 New list (新清單)，然後輸入新的主題名稱。對於 Email list: (電子郵件清單：) 欄位，請輸入至少一個電子郵件地址。您可以利用逗號分隔來輸入多個電子郵件地址。
      - 若要使用現有的 Amazon SNS 主題，請選擇要使用的主題名稱。如果清單不存在，請選擇 Select list (選取清單)。

   d. 選擇 Create Alarm (建立警示)。

步驟 3：測試 CloudWatch 警示

若要測試警示，請建立秘密，然後將其排定刪除。接著嘗試擷取秘密值。您很快就會在警示中設定的地址收到電子郵件。它會提醒您使用的秘密已排定要刪除。

AWS Secrets Manager 的合規驗證

您使用 Secrets Manager 時的合規責任，取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。AWS 會提供以下資源協助您處理合規事宜：

- 安全與合規快速入門指南：這些部署指南討論架構考量，並提供在 AWS 上部署以安全及合規為重心之基準環境的步驟。
- HIPAA 安全與合規架構白皮書：本白皮書說明公司可如何運用 AWS 來建立 HIPAA 合規的應用程式。
- AWS 合規資源：這組手冊和指南可能適用於您的產業和位置。
- AWS Config 可評定資源組態與內部實務、業界準則和法規的合規狀態。如需更多詳細資訊，請參閱 the section called “稽核秘密以合規”。
- AWS Security Hub 可供您檢視 AWS 中的安全狀態，可助您檢查是否符合安全產業標準和最佳實務。如需有關使用 Security Hub 評估 Secrets Manager 資源的資訊，請參閱《AWS Security Hub 使用者指南》中的 AWS Secrets Manager 控制項。
- IAM Access Analyzer 會分析允許外部實體存取秘密的政策 (包括政策中的條件陳述式)。如需詳細資訊，請參閱使用 Access Analyzer 預覽存取。
- AWS Systems Manager 為 Secrets Manager 提供了預先定義的 Runbook。如需詳細資訊，請參閱《適用於 Secrets Manager 的 Systems Manager Automation Runbook 參考》。

AWS Secrets Manager 已針對下列標準進行稽核，可作為需要取得合規認證時的一部分解決方案。

	AWS 已擴大其健康保險流通與責任法案 (HIPAA) 合規計劃，並加入 AWS Secrets Manager 作為 HIPAA 合格服務。如果您與 AWS 具有已執行的商業夥伴協議 (BAA)，您可以使用 Secrets Manager 來協助建置 HIPAA 合規的應用程式。AWS 會將著重 HIPAA 白皮書提供給有興趣進一步瞭解的客戶，讓他們能利用 AWS 來處理和儲存運作狀態資訊。如需詳細資訊，請參閱 HIPAA 合規。
	AWS Secrets Manager 具備服務供應商第 1 級之支付卡產業 (PCI) 資料安全標準 (DSS) 3.2 版的合規聲明文件。使用 AWS 產品和服務存放、處理或傳輸持卡人資料的客戶，可以使用 AWS Secrets Manager 來管理自己的 PCI DSS 合規認證。如需 PCI DSS 的詳細資訊，包括如何索取 AWS PCI 合規套裝服務的副本，請參閱 PCI DSS 第 1 級。

	AWS Secrets Manager 已成功完成 ISO/IEC 27001、ISO/IEC 27017、ISO/IEC 27018 及 ISO 9001 的合規認證。如需詳細資訊，請參閱 ISO 27001、ISO 27017、ISO 27018 及 ISO 9001。
	系統與組織控制 (SOC) 報告是獨立的第三方檢驗報告，我們可根據其中內容瞭解 Secrets Manager 如何達成關鍵合規控制與目標。這些報告的用途是協助您和稽核人員瞭解為了支援操作與法規遵循所建立的 AWS 控制。如需詳細資訊，請參閱 SOC 合規。
	聯邦風險與授權管理計畫 (FedRAMP) 是一項政府整體計畫，提供標準化的方法，為雲端產品和服務進行安全評估、授權和持續監控。FedRAMP 計畫還針對東部/西部和 GovCloud 的服務和區域提供臨時授權，以使用政府或監管資料。如需詳細資訊，請參閱 FedRAMP 合規。
	國防部 (DoD) 雲端運算安全要求指南 (SRG) 提供標準化的評定和授權程序，讓雲端服務提供者 (CSP) 取得 DoD 臨時授權，以便為 DoD 客戶提供服務。如需詳細資訊，請參閱 DoD SRG 資源
	資訊安全註冊評估人計畫 (IRAP) 可讓澳洲政府客戶驗證是否有適當的控制措施，並決定適當的責任模式，以便滿足由澳洲網路安全中心 (ACSC) 所制定的澳洲政府資訊安全手冊 (ISM) 要求。如需詳細資訊，請參閱 IRAP 資源
	Amazon Web Services (AWS) 取得委外服務供應商的稽核報告 (OSPAR) 認證。AWS 符合新加坡銀行公會 (ABS)《委外服務供應商控制目標與程序準則》(ABS 準則)，向客戶展示 AWS 致力於滿足新加坡金融服務業對雲端服務供應商的高度期望。如需詳細資訊，請參閱 OSPAR 資源

您可使用 AWS Artifact 下載第三方稽核報告。如需詳細資訊，請參閱 AWS Artifact 中的下載報告。

使用 AWS Config 來稽核 AWS Secrets Manager 秘密以合規

您可以使用 AWS Config 評估您的秘密，並評定其與內部實務、業界準則和法規的合規狀態。您可以使用 AWS Config 規則來定義秘密的內部安全與合規要求。然後 AWS Config 可以識別不符合規則的秘密。您也可以追蹤秘密中繼資料、輪換組態、用於秘密加密的 KMS 金鑰、Lambda 輪換功能以及與秘密相關聯之標籤的變更。

您可以從 Amazon SNS 接收有關秘密組態的通知。例如，您可以收到關於未設定輪換之秘密清單的 Amazon SNS 通知，這些通知可讓您推動輪換秘密的安全最佳實務。

如果您在組織的多個 AWS 帳戶 和 AWS 區域 中擁有秘密，就可以彙總組態和合規資料。

若要為秘密新增規則

- 遵循使用 AWS Config 受管規則中的指示，選擇以下其中一項規則：
  - secretsmanager-rotation-enabled-check – 檢查是否為 Secrets Manager 中存放的秘密設定了輪換。
  - secretsmanager-scheduled-rotation-success-check – 檢查上次成功輪換是否在設定的輪換頻率範圍內。檢查的最低頻率為每日。
  - secretsmanager-secret-periodic-rotation – 檢查是否在指定的天數內輪換秘密。
  - secretsmanager-secret-unused – 檢查是否在指定的天數內存取秘密。
  - secretsmanager-using-cmk – 檢查是否使用 AWS 受管金鑰 aws/secretsmanager 或您在 AWS KMS 中建立的客戶受管金鑰來加密秘密。

儲存規則之後，每次秘密的中繼資料變更時，AWS Config 都會評估您的秘密。您可以設定 AWS Config 以為您通知變更。如需詳細資訊，請參閱 AWS Config 傳送至 Amazon SNS 主題的通知。

彙總 AWS 帳戶 和 AWS 區域 中的秘密

您可以設定「AWS Config 多帳戶多區域資料彙整工具」以檢閱貴組織中所有帳戶和區域之間的秘密組態，然後檢閱您的秘密組態，並且與秘密管理最佳實務進行比較。

在建立彙整工具之前，您必須啟用所有帳戶和區域之間的秘密專屬的 AWS Config 和 AWS Config 受管規則。如需詳細資訊，請參閱使用 CloudFormation StackSets 在多個 AWS 帳戶 和區域中佈建資源。

如需 AWS Config 彙整工具的詳細資訊，請參閱 《AWS Config 開發人員指南》中的多帳戶多區域資料彙整和使用主控台設定彙整工具。

AWS Secrets Manager 中的安全

安全是 AWS 最重視的一環。身為 AWS 的客戶，您將能從資料中心和網路架構中獲益，這些都是專為最重視安全的組織而設計的。

你和 AWS 要一同承擔安全方面的責任。共同責任模型將此描述為雲端本身的安全和雲端內部的安全：

- 雲端本身的安全 – AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎設施。AWS 也提供您可安全使用的服務。在 AWS 合規計劃中，第三方稽核員會定期測試並驗證我們的安全功效。若要進一步瞭解適用於 AWS Secrets Manager 的合規計劃，請參閱 合規計劃範圍內的 AWS 服務。
- 雲端內部的安全 — 您的 AWS 服務會決定您的責任。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

如需更多資源，請參閱安全性支柱 – AWS Well-Architected 架構。

主題

降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險

若要使用 AWS Command Line Interface (AWS CLI) 來呼叫 AWS 操作，則可在命令 shell 中輸入命令。例如，您可以使用 Windows 命令提示字元或 Windows PowerShell，或 Bash 或 Z shell 等等。其中許多命令 Shell 包含旨在提高生產力的功能。但是，此功能可以用來洩露您的秘密。例如，在大多數 Shell 中，您可以使用向上鍵來查看最後輸入的命令。此「命令歷程記錄」功能可能被存取不安全工作階段的任何人濫用。此外，其他在背景執行的公用程式可能具有存取您命令參數的權限 (旨在協助您更有效率地執行任務)。為了降低這類風險，請確認您採取以下步驟：

- 當您離開主控台時，請務必鎖上電腦。
- 解除安裝或停用您不需要或不再使用的主控台公用程式。

- 確保 Shell 和遠端存取程式 (若您有使用) 不會記錄輸入的命令。
- 使用技術來傳遞不會被 Shell 命令歷程記錄擷取的參數。下列範例會說明如何在文字檔案中輸入秘密文字，然後將此檔案傳遞至 AWS Secrets Manager 命令後，立即銷毀。這表示典型 shell 歷史記錄不會擷取秘密文字。

  以下範例顯示典型 Linux 命令 (但您的 Shell 可能需要有些許不同的命令)：

```
$ touch secret.txt
        # Creates an empty text file
$ chmod go-rx secret.txt
        # Restricts access to the file to only the user
$ cat > secret.txt
        # Redirects standard input (STDIN) to the text file
ThisIsMyTopSecretPassword^D
        # Everything the user types from this point up to the CTRL-D (^D) is saved in
 the file
$ aws secretsmanager create-secret --name TestSecret --secret-string file://
secret.txt       # The Secrets Manager command takes the --secret-string parameter
 from the contents of the file
$ shred -u secret.txt
        # The file is destroyed so it can no longer be accessed.
```

在您執行這些命令後，應要能夠使用上下箭號來捲動命令歷程記錄，並可看到秘密文字未顯示於任何列。

> **Important**
>
> 在預設情況下，您必須先將命令歷史記錄緩衝區的大小降低至 1，否則無法在 Windows 中執行相同技術。

若要設定 Windows 命令提示字元為只有 1 個命令的 1 個歷程記錄緩衝區

1. 開啟管理員命令提示字元，或選擇 Run as administrator (以系統管理員身分執行)。
2. 選擇上方左側的圖示，然後選擇 Properties (屬性)。
3. 在 Options (選項) 索引標籤上，將 Buffer Size (緩衝區大小) 及 Number of Buffers (緩衝區數量) 皆設定為 **1**，然後選擇 OK (確定)。
4. 當您需要輸入不想留存於歷史記錄中的命令時，請立即接續輸入另一個命令，例如：

```
echo.
```

這可確保您刷新敏感命令。

對於 Windows 命令提示字元 Shell，您可以下載 SysInternals SDelete 工具，然後使用類似以下的命令：

```
C:\> echo. 2> secret.txt
        # Creates an empty file
C:\> icacls secret.txt /remove "BUILTIN\Administrators" "NT AUTHORITY/SYSTEM" /
inheritance:r   # Restricts access to the file to only the owner
C:\> copy con secret.txt /y
        # Redirects the keyboard to text file, suppressing prompt to overwrite
THIS IS MY TOP SECRET PASSWORD^Z
      # Everything the user types from this point up to the CTRL-Z (^Z) is saved in the
 file
C:\> aws secretsmanager create-secret --name TestSecret --secret-string file://
secret.txt      # The Secrets Manager command takes the --secret-string parameter from
 the contents of the file
C:\> sdelete secret.txt
        # The file is destroyed so it can no longer be accessed.
```

AWS Secrets Manager 中的資料保護

AWS 共同的責任模型適用於 AWS Secrets Manager 中的資料保護。如此模型所述，AWS 負責保護執行所有 AWS 雲端 的全球基礎設施。您必須負責維護在此基礎設施上託管之內容的控制權。此內容包括您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，建議您使用 AWS 帳戶 (IAM) 保護 AWS Identity and Access Management 憑證，並設定個別使用者帳戶。如此一來，每個使用者都只會獲得授予完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。Secrets Manager 支援所有區域中的 TLS 1.2 和 1.3。Secrets Manager 也支援混合型適用於 TLS (PQTLS) 的後量子金鑰交換選項網路加密通訊協定。

- 使用存取金鑰 ID 和與 IAM 主體關聯的私密存取金鑰來簽署您對 Secrets Manager 的程式設計請求。或者，您可以使用 AWS Security Token Service (AWS STS) 來產生暫時性安全憑證以簽署請求。
- 使用 AWS CloudTrail 設定 API 和使用者活動記錄。請參閱 the section called “使用 AWS CloudTrail 記錄 ”。
- 如果您在透過命令列介面或 API 存取 AWS 時，需要 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。請參閱 the section called “Secrets Manager 端點”。
- 如果您使用 AWS CLI 來存取 Secrets Manager，the section called “降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險”。

靜態加密

Secrets Manager 透過 AWS Key Management Service (AWS KMS) 使用加密功能來保護靜態資料的機密性。AWS KMS 提供許多 AWS 服務使用的金鑰存放和加密服務。Secrets Manager 中的每個秘密，都使用唯一資料金鑰加密。每個資料金鑰都由 KMS 金鑰保護。您可以選擇為該帳戶搭配 Secrets Manager AWS 受管金鑰 使用預設加密，或者可以在 AWS KMS 中建立自己的客戶管理金鑰。使用客戶管理金鑰，可讓您更精細進行對 KMS 金鑰活動的授權控制。如需更多詳細資訊，請參閱 the section called “秘密加密和解密”。

傳輸中加密

Secrets Manager 會提供安全且私有的端點，以供您加密傳輸中的資料。安全且私有的端點可讓 AWS 保護對於 Secrets Manager 之 API 請求的完整性。AWS 需要呼叫者使用 X.509 憑證和 / 或 Secrets Manager 私密存取金鑰簽署 API 呼叫。Signature 第 4 版簽署程序 (Sigv4) 規定了這項要求。

如果使用 AWS Command Line Interface (AWS CLI) 或任何 AWS 開發套件來呼叫 AWS，您要設定要使用的存取金鑰。然後這些工具會自動使用存取金鑰來為您簽署請求。請參閱 the section called “降低使用 AWS CLI 來存放 AWS Secrets Manager 秘密的風險”。

網際網路流量隱私權

AWS 提供了在透過已知和私人網路路由來路由傳送流量時，可供維護隱私權的選項。

服務和內部部署用戶端與應用程式之間的流量。

在您的私有網路和 AWS Secrets Manager 之間，您有兩個連線選項：

- AWS Site-to-Site VPN 連接。如需詳細資訊，請參閱什麼是 AWS Site-to-Site VPN？
- AWS Direct Connect 連線。如需詳細資訊，請參閱什麼是 AWS Direct Connect？

相同區域中 AWS 資源間的流量

如果您想要保護 Secrets Manager 與 AWS 中 API 用戶端之間的流量，請設定 AWS PrivateLink，以便以私有方式存取 Secrets Manager API 端點。

加密金鑰管理

當 Secrets Manager 需要加密新版本的受保護秘密資料時，Secrets Manager 會將請求傳送給 AWS KMS，以便從 KMS 金鑰產生新的資料金鑰。Secrets Manager 使用此資料金鑰進行信封加密。Secrets Manager 會將加密的資料金鑰與加密的秘密一起存放。當秘密需要解密時，Secrets Manager 會要求 AWS KMS 解密資料金鑰。Secrets Manager 接著會使用已解密的資料金鑰來解密加密的秘密。Secrets Manager 絕不會以未加密的形式存放資料金鑰，而且會盡快將其從記憶體中移除。如需更多詳細資訊，請參閱 the section called “秘密加密和解密”。

秘密加密和解密 AWS Secrets Manager

Secrets Manager 使用包絡加密與 AWS KMS 金鑰和資料金鑰來保護每個密碼值。每當密碼中的密碼值發生變更時，Secrets Manager 就會要求新的資料金鑰 AWS KMS 來保護它。資料金鑰將在 KMS 金鑰下加密，並會存放在秘密的中繼資料中。若要解密密碼，秘密管理員會先使用 KMS 金鑰解密加密的資料金鑰。 AWS KMS

Secrets Manager 不會直接使用 KMS 金鑰來加密秘密值。相反地，它會使用 KMS 金鑰來產生和加密 256 位元的進階加密標準 (AES) 對稱資料金鑰，再使用資料金鑰來加密秘密值。Secret Manager 會使用純文字資料金鑰來加密外部的密碼值 AWS KMS，然後將其從記憶體中移除。它將加密的資料金鑰副本存放在秘密的中繼資料中。

當您建立密碼時，您可以選擇 AWS 帳戶 和區域中的任何對稱加密客戶管理金鑰，也可以使用用 AWS 受管金鑰 於 Secrets Manager (`aws/secretsmanager`)。如果您選擇 AWS 受管金鑰 `aws/secretsmanager`，但它還不存在，Secrets Manager 會建立它並將其與密碼產生關聯。對於帳戶中的每個秘密，您可以使用相同的 KMS 金鑰或不同的 KMS 金鑰。建議您使用其他 KMS 金鑰，為一組秘密設定金鑰的自訂許可，或者如果您想要稽核這些金鑰的特定作業。Secrets Manager 只支援對稱加密 KMS 金鑰。如果您在外部金鑰存放區中使用 KMS 金鑰，KMS 金鑰上的密碼編譯作業可能需要更長的時間，且較不可靠和耐用，因為請求必須在 AWS外傳輸。

如需變更秘密的加密金鑰的資訊，請參閱 the section called “變更秘密的加密金鑰”。

當您變更加密金鑰時，Secrets Manager 會使用新金鑰重新加密`AWSCURRENT`和`AWSPREVIOUS`版本。`AWSPENDING`為了避免將您鎖定在密碼之外，Secrets Manager 會使用先前的金鑰加密所有現有

版本。這表示您可以使用先前的金鑰或新金鑰來解密AWSCURRENTAWSPENDING、和AWSPREVIOUS版本。

為了使其只AWSCURRENT能通過新的加密密鑰進行解密，請使用新密鑰創建一個新版本的密鑰。然後，為了能夠解密密鑰版本，您必須具有新密鑰的權限。AWSCURRENT

若要尋找與密碼相關聯的 KMS 金鑰，請在主控台中檢視密碼，或呼叫ListSecrets或DescribeSecret。當密碼與秘 Secrets Manager (aws/secretsmanager) 相關聯時，這些作業不會傳回 KMS 金鑰識別碼。 AWS 受管金鑰

主題

會加密哪些資料？

Secrets Manager 會加密機密值，但不會加密下列項目：

- 秘密名稱和說明
- 輪換設定
- 與秘密相關聯的 KMS 金鑰 ARN
- 任何附加的 AWS 標籤

加密和解密程序

若要加密秘密中的秘密值，Secrets Manager 使用以下程序。

1. Secrets Manager 會使用 KMS 金鑰的識別碼來呼叫 AWS KMS GenerateDataKey作業，以及 256 位元 AES 對稱金鑰的要求。 AWS KMS 傳回純文字資料金鑰，以及在 KMS 金鑰下加密的該資料金鑰副本。

2. Secrets Manager 會使用純文字資料金鑰和進階加密標準 (AES) 演算法來加密之外的機密值。 AWS KMS使用完畢後，它會盡快從記憶體中移除這些純文字金鑰。
3. Secrets Manager 將加密的資料金鑰存放在秘密的中繼資料，以便可以用來解密秘密值。不過，沒有 Secrets Manager API 會傳回加密的秘密或加密的資料金鑰。

若要解密加密秘密值：

1. Secrets Manager 會呼叫「 AWS KMS 解密」作業，並傳入加密的資料金鑰。
2. AWS KMS 使用 KMS 金鑰做為密碼來解密資料金鑰。它會傳回純文字資料金鑰。
3. Secrets Manager 使用純文字資料金鑰來解密秘密值。接著，它會盡快從記憶體中移除資料金鑰。

KMS 金鑰的許可

Secrets Manager 在密碼編譯操作中使用 KMS 金鑰時，等於是代表正在存取或更新秘密值的使用者執行動作。您可以在 IAM 政策或金鑰政策中授予許可。下列 Secrets Manager 作業需要 AWS KMS 權限。

- CreateSecret
- GetSecretValue
- PutSecretValue
- UpdateSecret
- ReplicateSecretToRegions

若要允許 KMS 金鑰僅用於來自 Secrets Manager 的要求，您可以在權限原則中使用公里：ViaService條件金鑰與`secretsmanager.<Region>.amazonaws.com`值。

您也可以使用加密內容中的金鑰或值作為條件金鑰，以便將 KMS 金鑰用於密碼編譯操作。例如，您可以在 IAM 或金鑰政策文件中使用字串條件運算子，或在授權中使用授權限制。KMS 金鑰授予傳播可能需時 5 分鐘。如需詳細資訊，請參閱CreateGrant。

Secrets Manager 如何使用您的 KMS 金鑰

Secrets Manager 會使用您的 KMS 金鑰呼叫下列 AWS KMS 作業。

GenerateDataKey

Secrets Manager 會呼叫 AWS KMS GenerateDataKey作業以回應下列 Secrets Manager 作業。

- CreateSecret— 如果新密碼包含秘密值，Secrets Manager 會要求新的資料金鑰加密。
- PutSecretValue— Secrets Manager 要求新的資料金鑰來加密指定的密碼值。
- ReplicateSecretToRegions— 若要加密複寫的密碼，Secrets Manager 會要求複本區域中 KMS 金鑰的資料金鑰。
- UpdateSecret— 如果您變更密碼值或 KMS 金鑰，Secret Manager 會要求新的資料金鑰來加密新的密碼值。

作RotateSecret業不會呼叫GenerateDataKey，因為它不會變更密碼值。不過，如果 RotateSecret 調用的 Lambda 函數變更了秘密值，其對 PutSecretValue 操作的呼叫會觸發 GenerateDataKey 請求。

解密

Secrets Manager 會呼叫解密操作來回應以下 Secrets Manager 操作。

- GetSecretValue和 BatchGetSecretValue— 秘密管理器將秘密值返回給調用者之前解密. 若要解密加密的密碼值，Secrets Manager 會呼叫「 AWS KMS 解密」作業來解密中的加密資料金鑰。接著，使用純文字資料金鑰來解密加密的秘密值。對於批次命令，Secrets Manager 可以重複使用解密的金鑰，因此並非所有呼叫都會產生 Decrypt 要求。
- PutSecretValue和 UpdateSecret-大多數PutSecretValue和UpdateSecret請求不會觸發Decrypt操作。不過，當 PutSecretValue 或 UpdateSecret 請求嘗試變更現有秘密版本中的秘密值時，Secrets Manager 會解密現有的秘密值並將其與請求中的秘密值進行比較，確認它們相同。這個動作可確保 Secrets Manager 操作為等冪操作。若要解密加密的密碼值，Secrets Manager 會呼叫「 AWS KMS 解密」作業來解密中的加密資料金鑰。接著，使用純文字資料金鑰來解密加密的秘密值。
- ReplicateSecretToRegions— Secrets Manager 先解密主要區域中的秘密值，然後再使用複本區域中的 KMS 金鑰重新加密密密碼值。

加密

Secrets Manager 會呼叫 Encrypt 操作，回應以下 Secrets Manager 操作：

- UpdateSecret— 如果您變更 KMS 金鑰，Secrets Manager 會使用新金鑰重新加密保護AWSCURRENTAWSPREVIOUS、和AWSPENDING秘密版本的資料金鑰。
- ReplicateSecretToRegions— 秘密管理員會在複寫期間使用複本區域中的 KMS 金鑰重新加密資料金鑰。

DescribeKey

Secrets Manager 會呼叫DescribeKey作業，以決定當您在秘密管理員主控台中建立或編輯密碼時是否列出 KMS 金鑰。

驗證對 KMS 金鑰的存取

在建立或變更與秘密關聯的 KMS 金鑰後，Secrets Manager 會使用指定的 CMK 來呼叫 `GenerateDataKey` 和 `Decrypt` 操作。這些呼叫會確認呼叫者擁有使用 KMS 金鑰進行這些操作的許可。Secrets Manager 會捨棄這些操作的結果，不會在任何密碼編譯操作中使用它們。

您可以識別這些驗證呼叫，因為這些請求中 `SecretVersionId` 金鑰加密內容的值是 `RequestToValidateKeyAccess`。

> **Note**
>
> 在過去，Secrets Manager 驗證呼叫不包含加密內容。您可能會在較舊的 AWS CloudTrail 記錄檔中找到沒有加密內容的通話。

AWS 受管金鑰 (`aws/secretsmanager`) 的金鑰政策

Secrets Manager (`aws/secretsmanager`) 的金鑰原則授予使用者權限，只有當 Secrets Manager 代表使用者提出要求時，才能將 KMS 金鑰用於指定的作業。 AWS 受管金鑰 金鑰政策不允許任何使用者直接使用 KMS 金鑰。

此金鑰政策與所有 AWS 受管金鑰 的政策一樣，都是由服務建立。您無法變更金鑰政策，但可以隨時進行檢視。如需詳細資訊，請參閱檢視金鑰政策。

金鑰政策中的政策陳述式具有下列效果：

- 只在請求來自代表使用者的 Secrets Manager 時，才允許帳戶中的使用者將 KMS 金鑰用於密碼編譯操作。`kms:ViaService` 條件金鑰會強制實施此限制。
- 允許 AWS 帳戶建立 IAM 政策，以允許使用者檢視 KMS 金鑰屬性和撤銷授權。
- 雖然 Secrets Manager 不會使用授權來取得 KMS 金鑰的存取權，但政策也允許 Secrets Manager 代表使用者為 KMS 金鑰建立授權，並允許帳戶撤銷任何授權 (該授權允許 Secrets Manager 使用 KMS 金鑰)。這些是 AWS 受管金鑰之政策文件的標準元素。

以下是 Secrets Manager 範 AWS 受管金鑰 例的金鑰原則。

```
{
  "Id": "auto-secretsmanager-2",
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "Allow access through AWS Secrets Manager for all principals in the
account that are authorized to use AWS Secrets Manager",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:CallerAccount": "111122223333",
          "kms:ViaService": "secretsmanager.us-west-2.amazonaws.com"
        }
      }
    },
    {
      "Sid": "Allow access through AWS Secrets Manager for all principals in the
account that are authorized to use AWS Secrets Manager",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": "kms:GenerateDataKey*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:CallerAccount": "111122223333"
        },
        "StringLike": {
          "kms:ViaService": "secretsmanager.us-west-2.amazonaws.com"
        }
      }
    },
```

```
    {
      "Sid": "Allow direct access to key metadata to the account",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Action": [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
      ],
      "Resource": "*"
    }
  ]
}
```

Secrets Manager 加密內容

加密內容是一組金鑰/值對，其中包含任意非私密資料。當您在加密資料的要求中包含加密內容時，AWS KMS 密碼編譯會將加密內容繫結至加密的資料。若要解密資料，您必須傳遞相同的加密內容。

在其GenerateDataKey和「解密」要求中 AWS KMS，Secrets Manager 會使用具有兩個名稱的加密內容 — 值配對來識別密碼及其版本，如下列範例所示。名稱不會改變，但組合的加密內容值對於每個秘密值都是不同的。

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
a1b2c3",
    "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

您可以使用加密內容在稽核記錄和日誌 (例如和 Amazon CloudWatch Logs) 中識別這些加密操作，並做為政策和授權中授權的條件。AWS CloudTrail

Secrets Manager 加密內容包含兩個名稱值對。

- SecretARN – 第一個名稱值對會識別秘密。金鑰為 `SecretARN`。值是秘密的 Amazon Resource Name (ARN)。

```
"SecretARN": "ARN of an Secrets Manager secret"
```

例如，如果秘密的 ARN 是 arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3，加密內容會包含下列對組。

```
"SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3"
```

- SecretVersionId— 第二個名稱 — 值配對可識別密碼的版本。金鑰為 SecretVersionId。值為版本 ID。

```
"SecretVersionId": "<version-id>"
```

例如，如果秘密的版本 ID 是 EXAMPLE1-90ab-cdef-fedc-ba987SECRET1，加密內容會包含下列對組。

```
"SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
```

當您建立或變更密碼的 KMS 金鑰時，Secrets Manager 會傳送GenerateDataKey和解密 AWS KMS 要求，以驗證呼叫者是否有權使用 KMS 金鑰進行這些作業。它會捨棄回應；它不會將回應用於秘密值。

在這些驗證請求中，SecretARN 的值是秘密的實際 ARN，但 SecretVersionId 值是 RequestToValidateKeyAccess，如以下範例加密內容所示。這個特殊值可協助您識別日誌和稽核線索中的驗證請求。

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3",
    "SecretVersionId": "RequestToValidateKeyAccess"
}
```

> **Note**
>
> 在過去，Secrets Manager 驗證請求不包含加密內容。您可能會在較舊的 AWS CloudTrail 記錄檔中找到沒有加密內容的通話。

監控 Secrets Manager 與之互動 AWS KMS

您可以使用 AWS CloudTrail 和 Amazon CloudWatch 日誌來追蹤 Secrets Manager 代表您傳送 AWS KMS 的請求。如需監控秘密使用情況的詳細資訊，請參閱 監控秘密。

GenerateDataKey

當您在密碼中建立或變更密碼值時，Secrets Manager 會傳送指 AWS KMS 定密碼的 KMS 金鑰的GenerateDataKey要求。

記錄 GenerateDataKey 操作的事件類似於以下範例事件。請求由 secretsmanager.amazonaws.com 呼叫。參數包括秘密 KMS 金鑰的 Amazon Resource Name (ARN)、需要 256 位元金鑰的金鑰指標，以及識別秘密和版本的加密內容。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AROAIGDTESTANDEXAMPLE:user01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/user01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-05-31T23:23:41Z"
            }
        },
        "invokedBy": "secretsmanager.amazonaws.com"
    },
    "eventTime": "2018-05-31T23:23:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "secretsmanager.amazonaws.com",
    "userAgent": "secretsmanager.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-
secret-a1b2c3",
```

```
            "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
        }
    },
    "responseElements": null,
    "requestID": "a7d4dd6f-6529-11e8-9881-67744a270888",
    "eventID": "af7476b6-62d7-42c2-bc02-5ce86c21ed36",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

解密

當您取得或變更密碼的密碼值時，Secrets Manager 會傳送解密要求 AWS KMS 來解密加密的資料金鑰。對於批次命令，Secrets Manager 可以重複使用解密的金鑰，因此並非所有呼叫都會產生 `Decrypt` 要求。

記錄 `Decrypt` 操作的事件類似於以下範例事件。使用者是您 AWS 帳戶中存取資料表的主體。這些參數包括加密的資料表金鑰 (做為密文 Blob)，以及識別資料表和帳戶的加密內容。 AWS AWS KMS 從加密文字衍生出 KMS 金鑰的識別碼。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AROAIGDTESTANDEXAMPLE:user01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/user01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-05-31T23:36:09Z"
            }
        },
```

```
        "invokedBy": "secretsmanager.amazonaws.com"
    },
    "eventTime": "2018-05-31T23:36:09Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "secretsmanager.amazonaws.com",
    "userAgent": "secretsmanager.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-
secret-a1b2c3",
            "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
        }
    },
    "responseElements": null,
    "requestID": "658c6a08-652b-11e8-a6d4-ffee2046048a",
    "eventID": "f333ec5c-7fc1-46b1-b985-cbda13719611",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

加密

當您變更與密碼相關聯的 KMS 金鑰時，Secrets Manager 會傳送「加密」 AWS KMS 要求AWSCURRENT，以使用新金鑰重新加AWSPENDING密AWSPREVIOUS、和密碼版本。當您將秘密複寫到其他區域時，Secrets Manager 也會傳送 Encrypt 請求給 AWS KMS。

記錄 Encrypt 操作的事件類似於以下範例事件。使用者是您 AWS 帳戶中存取資料表的主體。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "AROAIGDTESTANDEXAMPLE:user01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/user01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "creationDate": "2023-06-09T18:11:34Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "secretsmanager.amazonaws.com"
    },
    "eventTime": "2023-06-09T18:11:34Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "secretsmanager.amazonaws.com",
    "userAgent": "secretsmanager.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-east-2:111122223333:key/EXAMPLE1-f1c8-4dce-8777-
aa071ddefdcc",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "SecretARN": "arn:aws:secretsmanager:us-
east-2:111122223333:secret:ChangeKeyTest-5yKnKS",
            "SecretVersionId": "EXAMPLE1-5c55-4d7c-9277-1b79a5e8bc50"
        }
    },
    "responseElements": null,
    "requestID": "129bd54c-1975-4c00-9b03-f79f90e61d60",
    "eventID": "f7d9ff39-15ab-47d8-b94c-56586de4ab68",
    "readOnly": true,
    "resources": [
        {
            "accountId": "AWS Internal",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/EXAMPLE1-f1c8-4dce-8777-
aa071ddefdcc"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
```

```
}
```

AWS Secrets Manager 中的基礎設施安全

作為一種受管服務，AWS Secrets Manager 受 AWS 全球網路安全保護。如需有關 AWS 安全服務以及 AWS 如何保護基礎設施的詳細資訊，請參閱 AWS 雲端安全。若要使用基礎設施安全性的最佳實務來設計您的 AWS 環境，請參閱安全性支柱 AWS 架構良好的框架中的基礎設施保護。

透過網路存取 Secrets Manager 是透過使用 TLS 的 AWS 已發佈 API 來進行。您可以從任何網路位置來呼叫 Secrets Manager API。但是，Secrets Manager 所支援的以資源為基礎存取政策可能包含與來源 IP 地址相關的限制。您也可以使用 Secrets Manager 資源政策來控制從特定虛擬私有雲端 (VPC) 端點或特定 VPC 存取機密。實際上，這只會隔離 AWS 網路內特定 VPC 對指定機密的網路存取。如需更多詳細資訊，請參閱 VPC 端點。

AWS Secrets Manager 中的彈性

AWS 會以 AWS 區域與可用區域為中心建置全球基礎設施。AWS區域提供多個分開且隔離的可用區域，並以低延遲、高輸送量和高度備援網路連線相互連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域可讓您更有可用性、容錯能力和擴充能力，較單一或多個資料中心的傳統基礎設施還高。

如需復原和災難復原的詳細資訊，請參閱可靠性支柱 – AWS Well-Architected 架構。

如需 AWS 區域與可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

後量子 TLS

Secrets Manager 支援適用於 Transport Layer Security (TLS) 網路加密通訊協定的混合式後量子金鑰交換選項。連線至 Secrets Manager API 端點時，您可使用此 TLS 選項。在後量子演算法標準化前，我們會提供此功能，以便您可以開始測試這些金鑰交換通訊協定對於 Secrets Manager 呼叫的影響。這些選用的混合式後量子金鑰交換功能至少與現今使用的 TLS 加密功能同樣安全，且還能提供其他安全優勢。不過，與現今使用的傳統金鑰交換通訊協定相較之下，這些功能會影響延遲和輸送量。

若要保護現今加密的資料防範潛在的未來攻擊，AWS 正在與密碼編譯社群攜手合作開發抵禦量子或後量子演算法。我們已在 Secrets Manager 端點中實作混合式後量子金鑰交換密碼套件。這些混合式密碼套件結合傳統與後量子元素，能夠確保您的 TLS 連線至少與使用傳統密碼套件一樣堅強。然而，由於混合式密碼套件的效能特性和頻寬要求不同於傳統金鑰交換機制，我們建議您對 API 呼叫測試這些套件。

Secrets Manager 支援所有區域中的 PQTLS，唯中國地區除外。

設定混合式後量子 TLS

1. 將 AWS 通用執行時間用戶端新增至 Maven 相依性。我們建議使用最新的可用版本。例如，此陳述式將新增 2.20.0 版。

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>aws-crt-client</artifactId>
  <version>2.20.0</version>
</dependency>
```

2. 將 AWS SDK for Java 2.x 新增至專案並將其初始化。在 HTTP 用戶端上啟用混合式後量子密碼套件。

```
SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
            .postQuantumTlsEnabled(true)
            .build();
```

3. 建立 Secrets Manager 非同步用戶端。

```
SecretsManagerAsyncClient SecretsManagerAsync = SecretsManagerAsyncClient.builder()
            .httpClient(awsCrtHttpClient)
            .build();
```

   現在，當您呼叫 Secrets Manager API 操作時，系統會使用混合式後量子 TLS 將您的呼叫傳輸至 Secrets Manager 端點。

如需有關使用混合式後量子 TLS 的詳細資訊，請參閱：

- AWS SDK for Java 2.x 開發人員指南與 AWS SDK for Java 2.x 發布的部落格文章。
- s2n-tls 簡介 (全新開放原始碼 TLS 實作) 和使用 s2n-tls。
- 國家標準技術研究所 (NIST) 的後量子加密法。
- 用於 Transport Layer Security 1.2 (TLS) 的混合式後量子金鑰封裝法 (PQ KEM)。

Secrets Manager 的後量子 TLS 可在所有 AWS 區域 區域使用，中國除外。

AWS Secrets Manager 疑難排解

使用此處的資訊，來協助您針對在使用 Secrets Manager 時可能遇到的問題進行診斷與修復。

如需有關輪換的問題，請參閱 the section called “ 輪換疑難排解”。

主題

向 Secrets Manager 發送請求時收到「存取遭拒」訊息

確認您擁有呼叫所請求之操作和資源的許可。管理員必須將 IAM 政策連接到您的 IAM 使用者或您所屬的群組，以授予許可。如果授予這些許可的政策內容中包含了任何條件，例如時刻或 IP 地址的限制，則當您傳送請求時，也必須滿足這些要求。關於檢視或修改 IAM 使用者、群組或角色的政策方面的相關資訊，請參閱 IAM 使用者指南中的政策的使用。如需有關 Secrets Manager 所需許可的詳細資訊，請參閱 身分驗證與存取控制。

如果您是手動簽署 API 請求，而沒有使用 AWS 開發套件，請確認已正確簽署請求。

暫時安全憑證的「存取遭拒」

確認用來提出請求的 IAM 使用者或角色擁有正確許可。臨時安全登入資料的許可衍生自 IAM 使用者或角色。這表示能提供的許可僅限於授予 IAM 使用者或角色的許可。關於臨時安全登入資料許可的決定方式，詳細資訊請參閱 IAM 使用者指南中的控管臨時安全登入資料的許可。

確認您的請求已正確簽署且請求的格式也正確。如需詳細資訊，請參閱已選開發套件的工具組文件，或參閱 IAM 使用者指南 中的使用臨時安全登入資料來請求存取 AWS 資源。

確認您的臨時安全憑證並未過期。如需詳細資訊，請參閱 IAM 使用者指南中的請求臨時安全憑證。

如需有關 Secrets Manager 所需許可的詳細資訊，請參閱 身分驗證與存取控制。

我所做的變更不一定都會立即顯示。

Secrets Manager 使用稱為最終一致性的分散式運算模型。您在 Secrets Manager (或其他 AWS 服務) 中所進行的任何變更，均需要一段時間才能出現在所有可能的端點中。部分延遲是由在伺服器之間、複寫區域之間和全球不同地區之間傳送資料所花費的時間造成。Secrets Manager 也會使用快取以提升效能，但在某些情況下，這可能會增加時間。直到先前快取的資料逾時後，才能看到變更。

設計您的全球應用程式以說明這些潛在的延遲。此外，確保它們如預期般運作，即使在某個位置所做的變更也不會立即顯示在另一個位置。

如需某些其他 AWS 服務如何受到最終一致性影響的詳細資訊，請參閱：

- Amazon Redshift 資料庫開發人員指南中的管理資料一致性
- Amazon Simple Storage Service 使用者指南中的 Amazon S3 資料一致性模式
- AWS 大數據部落格中的在使用 Amazon S3 和適用於 ETL 工作流程的 Amazon EMR 時確保一致性
- Amazon EC2 API 參考中的 Amazon EC2 最終一致性

在建立秘密時收到「無法使用非對稱 KMS 金鑰產生資料金鑰」的訊息

Secrets Manager 使用與秘密相關聯的對稱加密 KMS 金鑰，來為每個秘密值產生資料金鑰。您無法使用非對稱 KMS 金鑰。請確認您使用的是對稱加密 KMS 金鑰，而不是非對稱 KMS 金鑰。如需說明，請參閱標識非對稱 KMS 金鑰。

AWS CLI 或 AWS SDK 操作無法從部分 ARN 中找到我的秘密

在許多情況下，Secrets Manager 可以從 ARN 的一部分而非完整的 ARN 中找到您的秘密。但是，如果秘密名稱以連字號結尾，且後面接著六個字元，Secrets Manager 可能無法僅從部分 ARN 中找到秘密。建議您改為使用完整的 ARN 或機密名稱。

更多詳細資訊

Secrets Manager 會在秘密名稱末尾包含六個隨機字元，協助確保秘密 ARN 是唯一。如果刪除原始秘密，然後使用相同的名稱建立新秘密，則這兩個秘密會因為這些字元而具有不同的 ARN。具有舊秘密存取權的使用者不會自動取得新秘密的存取權，因為 ARN 不同。

Secrets Manager 會為秘密建構一個 ARN，它包含區域、帳戶、秘密名稱、一個連字符和六個字元，如下所示：

```
arn:aws:secretsmanager:us-east-2:111122223333:secret:SecretName-abcdef
```

如果您的秘密名稱以連字符和六個字元結尾，則只有部分 ARN 會出現在 Secrets Manager 中，就好像您指定了一個完整的 ARN 一樣。例如，您可能擁有 ARN 為 MySecret-abcdef 的秘密。

```
arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret-abcdef-nutBrk
```

如果您呼叫以下操作，它只使用秘密 ARN 的一部分，則 Secrets Manager 可能找不到秘密。

```
$ aws secretsmanager describe-secret --secret-id arn:aws:secretsmanager:us-
east-2:111122223333:secret:MySecret-abcdef
```

此秘密由 AWS 服務管理，您必須使用該服務才能更新它。

如果您在嘗試修改秘密時遇到此訊息，則只能使用訊息中列出的管理服務來更新秘密。如需更多詳細資訊，請參閱 由其他服務管理的秘密。

若要決定管理秘密的人員，您可以檢閱秘密名稱。由其他服務管理的秘密以該服務的 ID 為字首。或者，在 AWS CLI 中，呼叫 describe-secret，然後檢閱 OwningService 欄位。

AWS Secrets Manager 配額

Secrets Manager 已讀取的 API 擁有高 TPS 配額，而較少調用的控制平面 API 則擁有較低的 TPS 配額。建議您避免以超過每 10 分鐘一次的持續速率呼叫 `PutSecretValue` 或 `UpdateSecret`。在呼叫 `PutSecretValue` 和 `UpdateSecret` 更新機密值時，機密管理員會建立機密的新版本。當版本超過 100 個時，Secrets Manager 會移除未標記的版本，但不會移除建立時間不到 24 小時的版本。如果以超過每 10 分鐘一次的速率更新機密值，您建立的版本比機密管理員移除的版本更多，且您將達到機密版本的配額。

您可以在帳戶中經營多個區域，而且每個區域都有專屬的特定配額。

當一個 AWS 帳戶 中的應用程式使用不同帳戶擁有的 KMS 金鑰時，稱為跨帳戶請求。對於跨帳戶請求，機密管理員會調節發出請求的帳戶身分，而不是擁有機密的帳戶。例如，如果帳戶 A 中的身分使用帳戶 B 中的機密，使用該機密只會套用帳戶 A 中的配額。

Secrets Manager 配額

名稱	預設	可調整	說明
DeleteResourcePolicy、GetResourcePolicy、PutResourcePolicy 和 ValidateResourcePolicy 請求的合併速率	每個支援的區域：每秒 50 個	否	DeleteResourcePolicy、GetResourcePolicy、PutResourcePolicy 和 ValidateResourcePolicy API 合併請求的每秒交易數量上限。
DescribeSecret 和 GetSecretValue API 請求的合併速率	每個支援的區域：每秒 1 萬個	否	DescribeSecret 和 GetSecretValue API 合併請求的每秒交易數量上限。
PutSecretValue、RemoveRegionsFromReplication、ReplicateSecretToRegion、StopReplicationToReplica、UpdateSecret 和	每個支援的區域：每秒 50 個	否	PutSecretValue、RemoveRegionsFromReplication、ReplicateSecretToRegion、Stop

名稱	預設	可調整	說明
UpdateSecretVersionStage API 請求的合併速率			ReplicationToReplica、UpdateSecret 和 UpdateSecretVersionStage API 合併請求的每秒交易數量上限。
RestoreSecret API 請求的合併速率	每個支援的區域：每秒 50 個	否	RestoreSecret API 請求的每秒交易數量上限。
RotateSecret 和 CancelRotateSecret API 請求的合併速率	每個支援的區域：每秒 50 個	否	RotateSecret 和 CancelRotateSecret API 合併請求的每秒交易數量上限。
TagResource 和 UntagResource API 請求的合併速率	每個支援的區域：每秒 50 個	否	TagResource 和 UntagResource API 合併請求的每秒交易數量上限。
BatchGetSecretValue API 請求的速率	每個支援的區域：每秒 100	否	BatchGetSecretValue API 請求的每秒交易數量上限。
CreateSecret API 請求的速率	每個支援的區域：每秒 50 個	否	CreateSecret API 請求的每秒交易數量上限。
DeleteSecret API 請求的速率	每個支援的區域：每秒 50 個	否	DeleteSecret API 請求的每秒交易數量上限。
GetRandomPassword API 請求的速率	每個支援的區域：每秒 50 個	否	GetRandomPassword API 請求的每秒交易數量上限。

名稱	預設	可調整	說明
ListSecretVersionIds API 請求的速率	每個支援的區域：每秒 50 個	否	ListSecretVersionIds API 請求的每秒交易數量上限。
ListSecrets API 請求的速率	每個支援的區域：每秒 100	否	ListSecrets API 請求的每秒交易數量上限。
以資源為基礎的政策長度	每個受支援的區域：20,480 個	否	連接至機密之以資源為基礎的許可政策中每秒交易數量上限。
機密值大小	每個受支援的區域：65,536 個位元組	否	加密之機密值的大小上限。如果機密值是字串，則這是機密值中允許的字元數。
秘密	每個受支援的區域：50 萬個	否	此 AWS 帳戶的每個 AWS 區域中的機密數量上限。
機密的所有版本附加的預備標籤	每個受支援的區域：20	否	機密的所有版本連接之預備標籤的數量上限。
每個機密的版本數	每個受支援的區域：100	否	機密的版本數上限。

將重試新增至應用程式

AWS 用戶端可能會看到機密管理員的呼叫失敗，因為用戶端發生未預期的問題。或者呼叫可能因機密管理員的速率限制而失敗。當您超過 API 請求配額時，機密管理員會調節請求。它會拒絕其他有效的請求，並傳回 throttling 錯誤。對於這兩種類型的失敗，我們建議您經過短暫的等待時間後重試呼叫。這就是所謂的退避和重試策略。

如果遇到下列錯誤，您可能會想要將重試新增到應用程式碼：

暫時性錯誤和例外狀況

- RequestTimeout
- RequestTimeoutException
- PriorRequestNotComplete
- ConnectionError
- HTTPClientError

服務端調節和限制錯誤與例外狀況

- Throttling
- ThrottlingException
- ThrottledException
- RequestThrottledException
- TooManyRequestsException
- ProvisionedThroughputExceededException
- TransactionInProgressException
- RequestLimitExceeded
- BandwidthLimitExceeded
- LimitExceededException
- RequestThrottled
- SlowDown

如需重試、指數退避和抖動的詳細資訊以及範例程式碼，請參閱下列資源：

- 指數退避和抖動
- 逾時、重試和退避 (具有抖動)
- AWS 中的錯誤重試與指數退避。

文件歷史紀錄

下表說明自上次發行版本以來文件的重要變更 AWS Secrets Manager。如需有關此文件更新的通知，您可以訂閱 RSS 訂閱源。

變更	描述	日期
Secrets Manager 變更為 AWS 受管理原則	受`SecretsManagerReadWrite` 管理的原則現在包含`redshift-serverless` 權限。如需詳細資訊，請參閱的AWS 受管理政策 AWS Secrets Manager	2024年3月12日

舊版更新

下表描述了 2024 年 2 月之前，《 AWS Secrets Manager 用戶指南》每個發行版本的重要變更。

變更	描述	日期
一般可用性	這是 Secrets Manager 的初始公開發行版本。	二零一八年四月四日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。