aws

Guía del usuario

AWS Secrets Manager

AWS Secrets Manager: Guía del usuario

Table of Contents

¿Qué es AWS Secrets Manager?

AWS Secrets Manager le ayuda a gestionar, recuperar y rotar las credenciales de las bases de datos, las credenciales de las aplicaciones, los tokens de OAuth, las claves de API y otros datos secretos a lo largo de sus ciclos de vida. Muchos AWS servicios almacenan y utilizan secretos en Secrets Manager.

Secrets Manager ayuda a mejorar la posición de seguridad, ya que ya no necesita credenciales de codificación rígida en el código fuente de la aplicación. El almacenamiento de las credenciales en Secrets Manager ayuda a evitar una posible concesión por parte de cualquier persona que pueda inspeccionar la aplicación o sus componentes. El usuario reemplaza las credenciales de codificación rígida con una llamada de tiempo de ejecución al servicio de Secrets Manager para recuperar las credenciales de forma dinámica cuando las necesita.

Con Secrets Manager, puede configurar un programa de rotación automática para sus secretos. Esto le permite reemplazar secretos a largo plazo con secretos a corto plazo, reduciendo significativamente el riesgo de peligro. Dado que las credenciales ya no se almacenan con la aplicación, su rotación ya no requiere la actualización de las aplicaciones ni la implementación de cambios en los clientes de la aplicación.

Para otros tipos de secretos que puede tener en su organización:

- AWS credenciales: recomendamos AWS Identity and Access Management.
- Claves de cifrado: recomendamos AWS Key Management Service.
- Claves SSH: recomendamos Amazon EC2 Instance Connect.
- Claves y certificados privados: recomendamos AWS Certificate Manager.

Comience a utilizar Secrets Manager

Si es la primera vez que utiliza Secrets Manager, comience con Conceptos o uno de los siguientes tutoriales:

- the section called “Reemplazar secretos codificados ”
- the section called “Reemplazar las credenciales de base de datos codificadas ”
- the section called “Estrategia de rotación de usuarios alternativos”
- the section called “Rotación de un solo usuario”

Otras tareas que puede realizar con los secretos:

- Cree y administre secretos
- Control del acceso a sus secretos
- Recuperar secretos
- Rotar secretos de
- Monitorear secretos
- Auditoría de secretos para la conformidad
- Crea secretos en AWS CloudFormation

Conformidad con los estándares

AWS Secrets Manager se ha sometido a auditorías para comprobar los distintos estándares y puede ser parte de su solución cuando necesite obtener una certificación de conformidad. Para obtener más información, consulte Validación de conformidad.

Precios

Cuando utiliza Secrets Manager, solo paga por lo que use, sin tarifas mínimas ni tarifas de configuración. No hay ningún cargo por los secretos que se marcan para su eliminación. Para obtener la lista de precios completa, consulte Precios deAWS Secrets Manager.

Puedes usar el Clave administrada de AWS `aws/secretsmanager` que crea Secrets Manager para cifrar tus secretos de forma gratuita. Si crea sus propias claves de KMS para cifrar sus secretos, se le AWS cobrará la tarifa actual AWS KMS . Para más información, consulte Precios deAWS Key Management Service.

Al activar la rotación automática (excepto la rotación gestionada), Secrets Manager utiliza una AWS Lambda función para girar el secreto y se le cobra por la función de rotación a la tasa Lambda actual. Para obtener más información, consulte AWS Lambda Precios.

Si lo habilitas AWS CloudTrail en tu cuenta, puedes obtener los registros de las llamadas a la API que envía Secrets Manager. Secrets Manager registra todos los eventos como eventos de administración. AWS CloudTrail almacena la primera copia de todos los eventos de administración de forma gratuita. Sin embargo, puede incurrir en cargos de Amazon S3 por almacenamiento de registros y de Amazon SNS si habilita las notificaciones. Además, si configura las pistas adicionales,

las copias adicionales de los eventos de administración pueden incurrir en costos. Para más información, consulte Precios deAWS CloudTrail.

AWS servicios que utilizan AWS Secrets Manager secretos

- AWS App Runner: consulte Referencia a variables de entorno y Administración de variables de entorno en la Guía para desarrolladores deAWS App Runner .
- AWS App2Container: consulte Administrar los secretos de AWS App2Container en la guía de uso de App2Container.AWS
- AWS AppConfig: consulte Creación de un perfil de configuración de formato libre en la Guía del usuario deAWS AppConfig .
- Amazon AppFlow: consulteSecretos gestionados por otros servicios.
- AWS AppSync: consulte Tutorial: Aurora Serverless en la Guía para desarrolladores deAWS AppSync .
- Amazon Athena: consulte Uso de consulta federada de Amazon Athena en la Guía del usuario de Amazon Athena.
- Amazon Aurora: consulte, consulteSecretos gestionados por otros servicios.
- AWS CodeBuild— Consulte Registro privado con un AWS Secrets Manager ejemplo CodeBuild en la Guía delAWS CodeBuild usuario.
- AWS DataSync: consulte Secretos gestionados por otros servicios.
- Amazon DataZone: consulte Creación de una fuente de datos para una base de datos de Amazon Redshift mediante una AWS Glue conexión nueva en la Guía DataZone del usuario de Amazon.
- AWS Direct Connect: consulte Secretos gestionados por otros servicios.
- AWS Directory Service— Consulte Unir sin problemas una instancia EC2 de Linux a su directorio AD AWS administrado de Microsoft, Unir sin problemas una instancia EC2 de Linux a su directorio AD Connector y Unir sin problemas una instancia de Linux EC2 a su directorio AD simple en laAWS Direct Connect Guía del usuario.
- Amazon DocumentDB (con compatibilidad con MongoDB): consulte the section called “Creación de un secreto de base de datos” y Managing Amazon DocumentDB Users en la Guía para desarrolladores de Amazon DocumentDB.
- AWS Elastic Beanstalk: consulte Configuración de Docker en la Guía para desarrolladores deAWS Elastic Beanstalk .
- Amazon Elastic Container Registry – consulte Creación de una regla de extracción de caché en la Guía del usuario de Amazon ECR.

- Amazon Elastic Container Service: consulte Tutorial: Specifying sensitive data using Secrets Manager secrets, Retrieve secrets programmatically through your application, Retrieve secrets through environment variables, Retrieve secrets for logging configuration, Tutorial: Using FSx for Windows File Server file systems with Amazon ECS, FSx for Windows File Server volumes y Private registry authentication for tasks en la Guía para desarrolladores de Amazon Elastic Container Service.
- Amazon Elastic Container Service Service Connect: consulteSecretos gestionados por otros servicios.
- Amazon ElastiCache: consulte Rotación automática de contraseñas para los usuarios en la Guía del ElastiCache usuario de Amazon.
- AWS Elemental Live— Consulta cómo MediaConnect funciona la entrega desde y AWS Elemental Live hasta en tiempo de ejecución en la Guía del usuario de Elemental Live.
- AWS Elemental MediaConnect: consulte Cifrado por clave estática en AWS Elemental MediaConnect en la Guía del usuarioAWS Elemental MediaConnect .
- AWS Elemental MediaConvert— Consulte Uso de Kantar para marcar con agua el audio en AWS Elemental MediaConvert las salidas de la Guía del AWS Elemental MediaConvert usuario.
- AWS Elemental MediaLive— Consulte Configuración MediaLive como entidad de confianza en la Guía del MediaLive usuario.
- AWS Elemental MediaPackage: consulte Acceso a Secrets Manager para obtener información sobre la autorización de CDN en la Guía del usuarioAWS Elemental MediaPackage .
- AWS Elemental MediaTailor— Consulte Configuración de la autenticación mediante token de AWS Secrets Manager acceso en la GuíaAWS Elemental MediaTailor del usuario.
- Amazon EMR que se ejecuta en Amazon EC2: consulte Store sensitive configuration data in Secrets Manager y Add a Git-based Repository to Amazon EMR en la Guía de administración de Amazon EMR.
- EMR sin servidor: consulte Secrets Manager for data protection with EMR Serverless en la Guía del usuario de Amazon EMR sin servidor.
- Amazon EventBridge: consulteSecretos gestionados por otros servicios.
- Amazon FSx: consulte File shares y Migrating file share configurations to Amazon FSx en la Guía del usuario de Amazon FSx para Windows File Server.
- AWS Glue DataBrew: consulte Secretos gestionados por otros servicios.
- AWS Glue Studio: consulta el tutorial: Uso del conector AWS Glue para Elasticsearch en la guía para AWS Glue desarrolladores.

- AWS IoT SiteWise: consulte Configuring data source authentication en la Guía del usuario deAWS IoT SiteWise .
- Amazon Kendra: consulte Using a database data source en la Guía del usuario de Amazon Kendra.
- Amazon Kinesis Video Streams: consulte Deploy the Amazon Kinesis Video Streams Edge Agent to AWS IoT Greengrass en la Guía para desarrolladores de Amazon Kinesis Video Streams.
- AWS Launch Wizard— Consulte Configurar Active Directory en la Guía del AWS Launch Wizard usuario. AWS Launch Wizard
- Amazon Lookout for Metrics: consulte Using Amazon RDS with Lookout for Metrics y Using Amazon Redshift with Lookout for Metrics en la Guía para desarrolladores de Amazon Lookout for Metrics.
- Amazon Managed Grafana: consulte Configuring Amazon Redshift en la Guía del usuario de Amazon Managed Grafana.
- AWS Managed Services: consulte AWS Secrets Manager (AMS self-service provisioning) en la Guía del usuario de AMS Advanced.
- Amazon Managed Streaming para Apache Kafka: consulte Username and password authentication with AWS Secrets Manager en la Guía para desarrolladores de Amazon Managed Streaming para Apache Kafka.
- Amazon Managed Workflows para Apache Airflow: consulte Configuración de una conexión de Apache Airflow mediante un secreto de Secrets Manager y Uso de una clave secreta AWS Secrets Manager para una variable de Apache Airflow en la guía del usuario de Amazon Managed Workflows para Apache Airflow.
- AWS Marketplace: consulte Secretos gestionados por otros servicios.
- AWS Migration Hub— Consulte Migración de NetWeaver aplicaciones SAP AWS y realojamiento de aplicaciones en Amazon EC2 en la Guía AWS Migration Hub del usuario de Orchestrator.
- AWS OpsWorks for Chef Automate: consulte Secretos gestionados por otros servicios.
- AWS Panorama: consulte Managing camera streams in AWS Panorama en la Guía para desarrolladores deAWS Panorama .
- AWS ParallelCluster: consulte Integrating Active Directory en la Guía del usuario deAWS ParallelCluster .
- Amazon Q: consulte Conceptos sobre autenticación en la Guía para desarrolladores de Amazon Q.
- Amazon QuickSight: consulte Uso de AWS Secrets Manager secretos en lugar de credenciales de bases de datos en Amazon QuickSight en la Guía del QuickSight usuario de Amazon.

- Amazon RDS: consulte Secretos gestionados por otros servicios.
- Amazon Redshift: consulte Almacenamiento de credenciales de bases de datos en Secretos gestionados por otros serviciosthe section called “Creación de un secreto de base de datos”AWS Secrets Manager, uso de la API de datos de Amazon Redshift y Consulta de una base de datos mediante el editor de consultas en la Guía de administración de Amazon Redshift.
- Editor de consultas de Amazon Redshift v2: consulte Secretos gestionados por otros servicios.
- Amazon SageMaker: consulte Asociar repositorios de Git con instancias de Amazon SageMaker Notebook, Importar datos de Databricks (JDBC) e Importar datos de Snowflake en la Guía para desarrolladores de Amazon. SageMaker
- AWS Schema Conversion Tool— Consulte Uso de la interfaz de usuario AWS Secrets Manager en la Guía del usuario AWS SCT.AWS Schema Conversion Tool
- AWS Toolkit for JetBrains: consulte Accessing Amazon Redshift clusters en la Guía del usuario deAWS Toolkit for JetBrains .
- AWS Transfer Family: consulte Basic authentication for AS2 connectors, Working with custom identity providers y Generate and manage PGP keys en la Guía del usuario deAWS Transfer Family .
- AWS Wickr: consulte Iniciar el robot de retención de datos en la Guía de administración deAWS Wickr.

Acceso AWS Secrets Manager

Puede trabajar con Secrets Manager de cualquiera de las siguientes formas:

- Consola de Secrets Manager
- Herramientas de la línea de comandos
- AWS SDK
- API de consulta HTTPS
- AWS Secrets Manager puntos finales

Consola de Secrets Manager

Puede administrar sus secretos mediante la consola de Secrets Manager basada en navegador y llevar a cabo prácticamente cualquier tarea relacionada con sus secretos por medio de ella.

Herramientas de la línea de comandos

Las herramientas de línea de AWS comandos le permiten ejecutar comandos en la línea de comandos del sistema para realizar Secrets Manager y otras AWS tareas. Esto puede ser más rápido y práctico que usar la consola. Las herramientas de línea de comandos pueden resultar útiles si desea crear scripts para realizar AWS tareas.

Cuando utiliza ingresa comandos en un shell de comandos, existe el riesgo de que se acceda al historial de comandos o de que las utilidades tengan acceso a sus parámetros de comando. Consulte the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

Las herramientas de línea de comandos utilizan automáticamente el punto final predeterminado para el servicio en una AWS región. Puede especificar un punto de conexión diferente para las solicitudes de la API. Consulte the section called “Puntos de conexión de Secrets Manager”.

AWS proporciona dos conjuntos de herramientas de línea de comandos:

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

AWS SDK

Los AWS SDK constan de bibliotecas y códigos de muestra para varios lenguajes de programación y plataformas. Los SDK incluyen tareas como firmar solicitudes criptográficamente, administrar errores y reintentar solicitudes automáticamente. Para descargar e instalar cualquiera de los SDK, consulte Herramientas para crear en Amazon Web Services.

Los AWS SDK utilizan automáticamente el punto final predeterminado para el servicio en una AWS región. Puede especificar un punto de conexión diferente para las solicitudes de la API. Consulte the section called “Puntos de conexión de Secrets Manager”.

Para obtener la documentación relativa a los SDK, consulte:

- C++
- Go
- Java
- JavaScript
- Kotlin
- .NET
- PHP
- Python (Boto3)
- Ruby
- Rust
- SAP ABAP
- Swift

API de consulta HTTPS

La API de consulta HTTPS le brinda acceso programático a Secrets Manager y AWS. La API de consulta HTTPS le permite emitir solicitudes HTTPS directamente al servicio.

Aunque puede realizar llamadas directas a la API de consulta HTTPS de Secrets Manager, se recomienda utilizar uno de los SDK. El SDK realiza muchas tareas de gran utilidad que, de otro modo, tendría que realizar de forma manual. Por ejemplo, los SDK firman automáticamente las solicitudes y convierten la respuesta en una estructura adecuada para su idioma desde el punto de vista sintáctico.

Para realizar llamadas HTTPS a Secrets Manager, debe conectarse a ???.

AWS Secrets Manager puntos finales

Para conectarse mediante programación a Secrets Manager, se debe utilizar un punto de conexión, la URL del punto de entrada del servicio. Puntos de conexión de Secrets Manager puntos de conexión de doble pila, lo que significa que admiten IPv4 e IPv6.

Secrets Manager ofrece puntos de conexión que admiten el Estándar de procesamiento de la información federal (FIPS) 140-2.

Secrets Manager admite TLS 1.2 y 1.3. Secrets Manager admite PQTLS en todas las regiones excepto las de China.

> **Note**
>
> El AWS SDK de Python y el AWS CLI intento de llamar a IPv6 y luego a IPv4 en secuencia, por lo que si no tienes IPv6 habilitado, puede pasar algún tiempo antes de que se agote el tiempo de espera de la llamada y volver a intentarlo con IPv4. Para solucionar este problema, puedes deshabilitar IPv6 por completo o migrar a IPv6.

Los siguientes son los puntos de conexión de servicio para Secrets Manager. Tenga en cuenta que la denominación difiere de la típica convención de nomenclatura de doble pila.

Nombre de la región	Región	Punto de conexión	Protocolo	
Este de EE. UU. (Ohio)	us-east-2	secretsmanager.us-east-2.amazonaws.com secretsmanager-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
Este de EE. UU. (Norte de Virginia)	us-east-1	secretsmanager.us-east-1.amazonaws.com secretsmanager-fips.us-east-1.amazonaws.com	HTTPS HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo
Oeste de EE. UU. (Norte de California)	us-west-1	secretsmanager.us-west-1.amazonaws.com secretsmanager-fips.us-west-1.amazonaws.com	HTTPS HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	secretsmanager.us-west-2.amazonaws.com secretsmanager-fips.us-west-2.amazonaws.com	HTTPS HTTPS
África (Ciudad del Cabo)	af-south-1	secretsmanager.af-south-1.amazonaws.com	HTTPS
Asia-Pacífico (Hong Kong)	ap-east-1	secretsmanager.ap-east-1.amazonaws.com	HTTPS
Asia-Pacífico (Hyderabad)	ap-south-2	secretsmanager.ap-south-2.amazonaws.com	HTTPS
Asia-Pacífico (Yakarta)	ap-southeast-3	secretsmanager.ap-southeast-3.amazonaws.com	HTTPS
Asia-Pacífico (Melbourne)	ap-southeast-4	secretsmanager.ap-southeast-4.amazonaws.com	HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo
Asia-Pacífico (Bombay)	ap-south-1	secretsmanager.ap-south-1.amazonaws.com	HTTPS
Asia-Pacífico (Osaka)	ap-northeast-3	secretsmanager.ap-northeast-3.amazonaws.com	HTTPS
Asia-Pacífico (Seúl)	ap-northeast-2	secretsmanager.ap-northeast-2.amazonaws.com	HTTPS
Asia-Pacífico (Singapur)	ap-southeast-1	secretsmanager.ap-southeast-1.amazonaws.com	HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	secretsmanager.ap-southeast-2.amazonaws.com	HTTPS
Asia-Pacífico (Tokio)	ap-northeast-1	secretsmanager.ap-northeast-1.amazonaws.com	HTTPS
Canadá (centro)	ca-central-1	secretsmanager.ca-central-1.amazonaws.com secretsmanager-fips.ca-central-1.amazonaws.com	HTTPS HTTPS
Oeste de Canadá (Calgary)	ca-west-1	secretsmanager.ca-west-1.amazonaws.com secretsmanager-fips.ca-west-1.amazonaws.com	HTTPS HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo	
Europa (Fráncfort)	eu-central-1	secretsmanager.eu-central-1.amazonaws.com	HTTPS	
Europa (Irlanda)	eu-west-1	secretsmanager.eu-west-1.amazonaws.com	HTTPS	
Europa (Londres)	eu-west-2	secretsmanager.eu-west-2.amazonaws.com	HTTPS	
Europa (Milán)	eu-south-1	secretsmanager.eu-south-1.amazonaws.com	HTTPS	
Europa (París)	eu-west-3	secretsmanager.eu-west-3.amazonaws.com	HTTPS	
Europa (España)	eu-south-2	secretsmanager.eu-south-2.amazonaws.com	HTTPS	
Europa (Estocolmo)	eu-north-1	secretsmanager.eu-north-1.amazonaws.com	HTTPS	
Europa (Zúrich)	eu-central-2	secretsmanager.eu-central-2.amazonaws.com	HTTPS	
Israel (Tel Aviv)	il-central-1	secretsmanager.il-central-1.amazonaws.com	HTTPS	
Medio Oriente (Baréin)	me-south-1	secretsmanager.me-south-1.amazonaws.com	HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo
Medio Oriente (EAU)	me-central-1	secretsmanager.me-central-1.amazonaws.com	HTTPS
América del Sur (São Paulo)	sa-east-1	secretsmanager.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (Este de EE. UU.)	us-gov-east-1	secretsmanager.us-gov-east-1.amazonaws.com secretsmanager-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS
AWS GovCloud (EE. UU.-Oeste)	us-gov-west-1	secretsmanager.us-gov-west-1.amazonaws.com secretsmanager-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS

Conceptos de AWS Secrets Manager

Los siguientes conceptos son importantes para comprender cómo funciona Secrets Manager.

- secreta
- Versión
- Rotation
- Estrategia de rotación

secreta

En Secrets Manager, un secreto comprende la información secreta, el valor secreto, además de los metadatos sobre ese secreto. Un valor secreto puede ser de tipo cadena o binario. Para almacenar varios valores de tipo cadena en un secreto, se recomienda utilizar una cadena de texto JSON con pares clave-valor, por ejemplo:

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

Entre los metadatos de un secreto se encuentran los siguientes:

- Un Nombre de recurso de Amazon (ARN) con el siguiente formato:

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

  Secrets Manager incluye seis caracteres de asignación al azar al final del nombre del secreto para garantizar que el ARN del secreto sea único. Si se elimina el secreto original y, a continuación, se crea un secreto nuevo con el mismo nombre, ambos tendrán ARN diferentes debido a estos caracteres. Los usuarios con acceso al secreto anterior no tienen acceso automático al secreto nuevo porque los ARN son diferentes.
- El nombre del secreto, una descripción, una política de recursos y las etiquetas.

- El ARN para una clave de cifrado, una AWS KMS key que Secrets Manager utiliza para cifrar y descifrar el valor del secreto. Secrets Manager almacena texto secreto en un formato cifrado y cifra el secreto en tránsito. Consulte the section called “Cifrado y descifrado de secretos”.
- Información sobre cómo rotar el secreto, si configura la rotación. Consulte the section called “Rotation”.

Secrets Manager utiliza políticas de permisos de IAM para garantizar que solo los usuarios autorizados tengan acceso al secreto y puedan modificarlo. Consulte Autenticación y control de acceso de AWS Secrets Manager.

Un secreto tiene versiones que tienen copias del valor cifrado del secreto. Cuando se cambia el valor secreto, o el secreto es rotado, el Secrets Manager crea una nueva versión. Consulte the section called “Versión”.

Puede usar un secreto en varias Regiones de AWS por replicación. Cuando se replica un secreto, se crea una copia del secreto original o secreto principal llamada secreto réplica. El secreto réplica permanece vinculado al secreto principal. Consulte the section called “Replicar un secreto a otras regiones”.

Consulte Cree y administre secretos.

Versión

Un secreto tiene versiones que tienen copias del valor cifrado del secreto. Cuando se cambia el valor secreto, o el secreto es rotado, el Secrets Manager crea una nueva versión.

Secrets Manager no almacena ningún historial lineal de secretos junto con las versiones. En cambio, etiqueta tres versiones específicas para hacer un seguimiento de ellas:

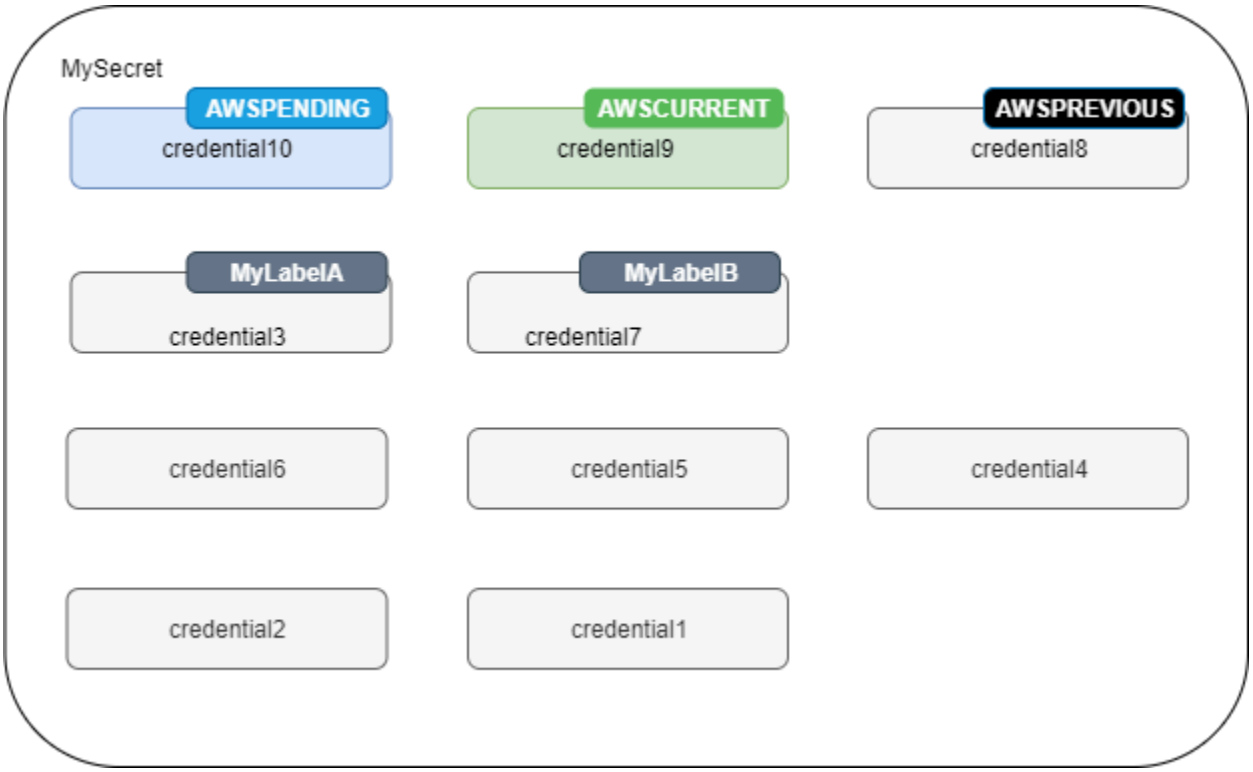
- Versión actual: `AWSCURRENT`
- Versión anterior: `AWSPREVIOUS`
- Versión pendiente (durante la rotación): `AWSPENDING`

Un secreto siempre tiene una versión con la etiqueta `AWSCURRENT` y Secrets Manager devuelve esa versión de forma predeterminada cuando se recupera el valor del secreto.

Para etiquetar versiones con sus propias etiquetas, llame a `update-secret-version-stage` en la AWS CLI. Puede adjuntar hasta 20 etiquetas a versiones en un secreto. Dos versiones de un secreto no puede tener la misma etiqueta provisional. Las versiones pueden tener varias etiquetas.

Secrets Manager nunca elimina las versiones etiquetadas, pero las versiones sin etiquetar se consideran obsoletas. Secrets Manager elimina las versiones obsoletas cuando hay más de 100. Secrets Manager no elimina versiones creadas hace menos de 24 horas.

En la siguiente ilustración, se muestra un secreto que tiene versiones etiquetadas por AWS y versiones etiquetadas por el cliente. Las versiones sin etiquetas se consideran obsoletas y Secrets Manager las eliminará en algún momento.



Rotation

La rotación es el proceso en el que periódicamente se cambia el secreto para que resulte más difícil que un atacante pueda acceder a las credenciales. En Secrets Manager, puede configurar la rotación automática de sus secretos. Cuando Secrets Manager rota un secreto, se actualizan las credenciales tanto en el secreto como en la base de datos o servicio. Consulte Rotar secretos de .

> **Tip**
>
> En algunos Secretos gestionados por otros servicios, se utiliza la rotación administrada. Para utilizar Rotación administrada, primero se debe crear el secreto a través del servicio de administración.

Estrategia de rotación

Secrets Manager ofrece dos estrategias de rotación:

- Estrategia de rotación: un solo usuario
- Estrategia de rotación: usuarios alternativos

Estrategia de rotación: un solo usuario

Esta estrategia actualiza las credenciales de un usuario en un secreto. En el caso de las instancias Db2 de Amazon RDS, dado que los usuarios no pueden cambiar sus propias contraseñas, debe proporcionar las credenciales de administrador en un secreto independiente. Esta es la estrategia de rotación más sencilla y es adecuada para la mayoría de los casos de uso. En particular, recomendamos que utilice esta estrategia para las credenciales de los usuarios interactivos o únicos (ad hoc).

Cuando el secreto rota, las conexiones de bases de datos abiertas no se eliminan. Mientras se produce la rotación, hay un breve periodo de tiempo entre el momento en que cambia la contraseña de la base de datos y el momento en que se actualiza el secreto. Durante este tiempo, existe un riesgo bajo de que la base de datos deniegue las llamadas que utilizan las credenciales rotadas. Puede mitigar este riesgo con una estrategia de reintentos apropiada. Tras la rotación, las nuevas conexiones utilizan las nuevas credenciales.

Estrategia de rotación: usuarios alternativos

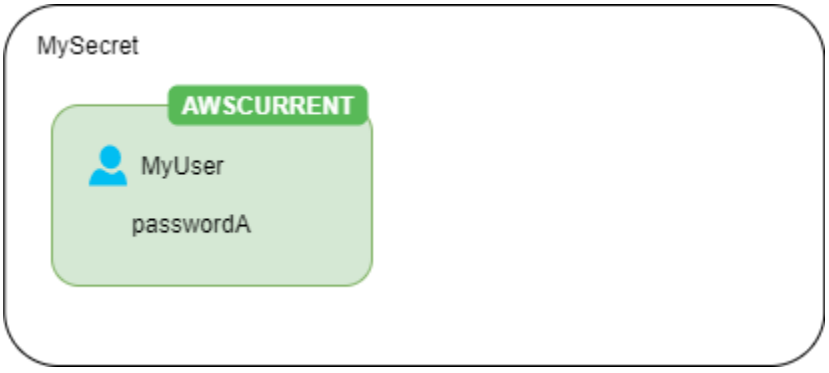
Esta estrategia actualiza las credenciales de dos usuarios en un secreto. Se crea el primer usuario y, durante la primera rotación, la función de rotación lo clona para crear el segundo usuario. Cada vez que el secreto rota, la función de rotación alterna la contraseña de usuario que actualiza. Dado que la mayoría de los usuarios no tienen permiso para clonarse a sí mismos, debe proporcionar las credenciales de un usuario de tipo `superuser` en otro secreto. Recomendamos que utilice

la estrategia de rotación de un solo usuario cuando los usuarios clonados en su base de datos no tienen los mismos permisos que el usuario original y para las credenciales de los usuarios interactivos o únicos (ad hoc).

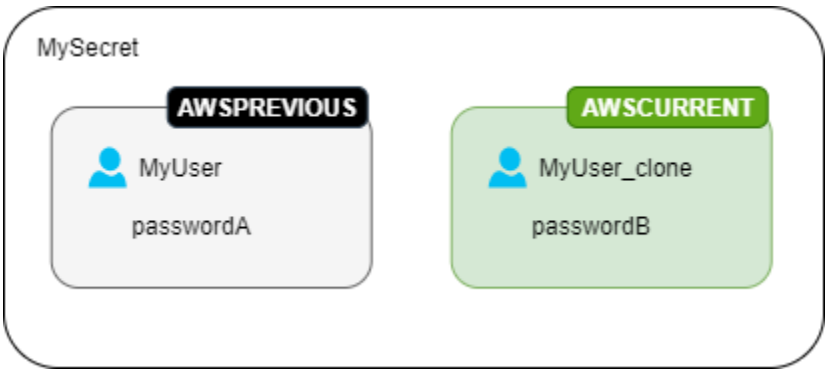
Esa estrategia es adecuada para bases de datos con modelos de permisos en los que un rol es propietario de las tablas de base de datos y un segundo rol tiene permiso para acceder a las tablas de base de datos. También es adecuada para aplicaciones que requieren alta disponibilidad. Si una aplicación recupera el secreto durante la rotación, seguirá obteniendo un conjunto de credenciales válido. Tras la rotación, las credenciales de `user` y `user_clone` son válidas. Incluso hay menos posibilidades de que las aplicaciones sufran denegaciones durante este tipo de rotación que con la rotación de un solo usuario. Si la base de datos está alojada en una granja de servidores donde el cambio de contraseña tarda tiempo en propagarse a todos los servidores, existe el riesgo de que la base de datos deniegue las llamadas que utilicen las nuevas credenciales. Puede mitigar este riesgo con una estrategia de reintentos apropiada.

Secrets Manager crea el usuario clonado con los mismos permisos que el usuario original. Si cambia los permisos del usuario original después de crear el clon, también debe cambiar los permisos del usuario clonado.

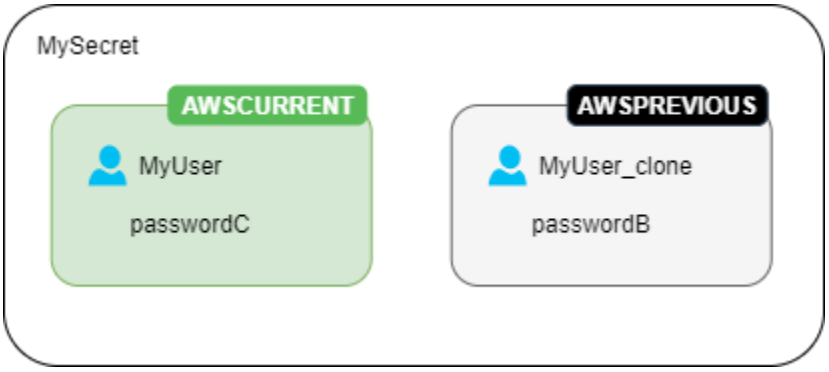
Por ejemplo, si crea un secreto con las credenciales de un usuario de base de datos, el secreto contiene una versión con esas credenciales.



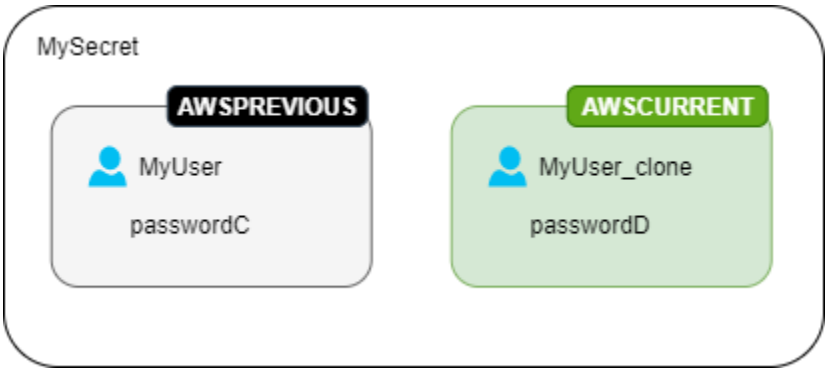
Primera rotación: la función de rotación crea un clon del usuario con una contraseña generada y esas credenciales se convierten en la versión del secreto actual.



Segunda rotación: la función de rotación actualiza la contraseña del usuario original.



Segunda rotación: la función de rotación actualiza la contraseña del usuario clonado.



Tutoriales de AWS Secrets Manager

Temas

Encuentre secretos sin protección en su código con el Revisor de Amazon CodeGuru

El Revisor de Amazon CodeGuru es un servicio que utiliza el análisis de programas y machine learning para detectar posibles defectos difíciles de encontrar para los desarrolladores y ofrece sugerencias para mejorar el código Java y Python. El Revisor CodeGuru se integra con Secrets Manager para encontrar secretos sin protección en su código. Para conocer los tipos de secretos que puede encontrar, consulte Types of secrets detected by CodeGuru Reviewer (Tipos de secretos detectados por el Revisor de CodeGuru) en la Guía del usuario del Revisor de Amazon CodeGuru..

Una vez haya encontrado secretos codificados, tome medidas para reemplazarlos:

- the section called “Reemplazar las credenciales de base de datos codificadas ”
- the section called “Reemplazar secretos codificados ”

Mover secretos codificados a AWS Secrets Manager

Si tiene secretos de texto sin formato en su código, le recomendamos que los rote y los almacene en Secrets Manager. Al mover el secreto a Secrets Manager se soluciona el problema de que sea visible para cualquiera que vea el código porque, en el futuro, el código recupera el secreto directamente de Secrets Manager. Al rotar el secreto se anula el secreto codificado actual para que ya no sea válido.

Para ver los secretos de credenciales de base de datos, consulte Mover las credenciales de base de datos codificadas a AWS Secrets Manager.

Antes de comenzar, debe determinar quién necesita acceso al secreto. Recomendamos utilizar dos roles de IAM para administrar el permiso a su secreto:

- Un rol que administra los secretos de su organización. Para obtener más información, consulte the section called “Permisos de Secrets Manager”. Creará y rotará el secreto utilizando este rol.
- Un rol que puede utilizar el secreto en tiempo de ejecución, por ejemplo, en este tutorial que utiliza `RoleToRetrieveSecretAtRuntime`. El código asume esta función para recuperar el secreto. En este tutorial, otorga al rol solamente el permiso para recuperar un valor secreto y concede el permiso mediante la política de recursos del secreto. Si desea conocer otras alternativas, consulte the section called “Pasos siguientes”.

Pasos:

Paso 1: Crear el secreto

El primer paso es copiar el secreto codificado existente en Secrets Manager. Si el secreto está relacionado con un recurso AWS, guárdelo en la misma región que el recurso. De lo contrario, guárdelo en la región que tenga la menor latencia para su caso de uso.

Para crear un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Elija Store a new secret (Almacenar un nuevo secreto).
3. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:
   a. En Secret type (Tipo de secreto), elija Other type of secret (Otro tipo de secreto).
   b. Ingrese su secreto como Key/value pairs (pares clave/valor) o en Plaintext (texto sin formato). Presentamos algunos ejemplos:

Pares clave/valor de API:

ClientID : *my_client_id*

ClientSecret : *wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

Pares clave/valor de credenciales:

Username : *saanvis*

Password : *CONTRASEÑA DE EJEMPLO*

Texto sin formato de token de OAuth:

AKIAI44QH8DHBEXAMPLE

Texto sin formato de certificado digital:

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

Clave privada de texto sin formato:

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

c. Para Clave encriptada, seleccione aws/secretsmanager para utilizar Clave administrada de AWS para Secrets Manager. No se aplica ningún cargo por el uso de esta clave. También puede utilizar su propia clave administrada por el cliente, por ejemplo, para acceder al secreto desde otro Cuenta de AWS. Para obtener información sobre los costos por usar una clave administrada por el cliente, consulte Precios.

d. Elija Next (Siguiente).

4. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:

   a. Ingrese un Nombre de secreto descriptivo y una Descripción.

b. En Permisos de recursos, seleccione Edit permissions (Editar permisos). Pegue la siguiente política, que permite a *RoleToRetrieveSecretAtRuntime* recuperar el secreto y, a continuación, seleccione Save (Guardar).

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

c. En la parte inferior de la página, elija Next.

5. En la página Configure rotation (Configurar rotación), mantenga la rotación desactivada. Elija Next (Siguiente).
6. En la página Review (Revisar), revise los detalles del secreto y, a continuación, elija Store (Almacenar).

Paso 2: Actualización del código

El código debe asumir el rol de IAM *RoleToRetrieveSecretAtRuntime* para poder recuperar el secreto. Para obtener más información, consulte Cambiar a un rol de IAM (AWS API).

A continuación, actualice el código para recuperar el secreto de Secrets Manager utilizando el código de ejemplo proporcionado por Secrets Manager.

Para encontrar el código de muestra

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la página Secrets (Secretos), elija el secreto.
3. Baje hasta Código de muestra. Elija su lenguaje de programación y, a continuación, copie el fragmento de código.

En su aplicación, elimine el secreto codificado y pegue el fragmento de código. Según el idioma del código, es posible que tenga que añadir una llamada a la función o método del fragmento.

Compruebe que su aplicación funciona según lo esperado con el secreto en lugar del secreto codificado.

Paso 3: Actualizar el secreto

El último paso consiste en revocar y actualizar el secreto codificado. Consulte la fuente del secreto para encontrar instrucciones para revocar y actualizar el secreto. Por ejemplo, es posible que tenga que desactivar el secreto actual y generar un nuevo secreto.

Para actualizar el secreto con el nuevo valor

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Seleccione Secrets (Secretos) y luego elija el secreto.
3. En la página Detalles del secreto, baje hasta Recuperar valor del secreto y seleccione Edit (Editar).
4. Actualice el secreto y, a continuación, seleccione Save (Guardar).

A continuación, compruebe que su aplicación funciona según lo esperado con el nuevo secreto.

Pasos siguientes

A continuación, algunas ideas a tener en cuenta después de eliminar un secreto codificado de su código:

- Para encontrar secretos codificados en sus aplicaciones Java y Python, le recomendamos el Revisor de Amazon CodeGuru.
- Puede mejorar el rendimiento y reducir los costos almacenando secretos en caché. Para obtener más información, consulte Recuperar secretos.
- Para los secretos a los que accede desde varias regiones, considere la posibilidad de replicar su secreto para mejorar la latencia. Para obtener más información, consulte the section called “Replicar un secreto a otras regiones”.
- En este tutorial, ha concedido a *RoleToRetrieveSecretAtRuntime* solo el permiso para recuperar el valor secreto. Para otorgar más permisos al rol, por ejemplo, para obtener metadatos

sobre el secreto o para ver una lista de secretos, consulte the section called “Ejemplos de políticas de permisos”.

- En este tutorial, ha concedido permiso a *RoleToRetrieveSecretAtRuntime* mediante la política de recursos del secreto. Para ver otras formas de conceder permiso, consulte the section called “Adjuntar una política de permisos a una identidad”.

Mover las credenciales de base de datos codificadas a AWS Secrets Manager

Si tienes credenciales de base de datos de texto sin formato en el código, te recomendamos que muevas las credenciales a Secrets Manager y luego las rotes inmediatamente. Al mover las credenciales a Secrets Manager se soluciona el problema de que sean visibles para cualquiera que vea el código porque, en el futuro, el código recupera las credenciales directamente de Secrets Manager. Al rotar el secreto se actualiza la contraseña y, a continuación, se anula la contraseña codificada actual para que ya no sea válida.

Para Amazon RDS, Amazon Redshift y Amazon DocumentDB, siga los pasos de esta página para mover credenciales codificadas a Secrets Manager. Para otro tipo de credenciales y otros secretos, consulte the section called “Reemplazar secretos codificados ”.

Antes de comenzar, debe determinar quién necesita acceso al secreto. Recomendamos utilizar dos roles de IAM para administrar el permiso a su secreto:

- Un rol que administra los secretos de su organización. Para obtener más información, consulte the section called “Permisos de Secrets Manager”. Creará y rotará el secreto utilizando este rol.
- Un rol que puede usar las credenciales en tiempo de ejecución, *RoleToRetrieveSecretAtRuntime*en este tutorial. El código asume esta función para recuperar el secreto.

Pasos:

- Paso 1: Crear el secreto
- Paso 2: Actualización del código
- Paso 3: rote el secreto
- Siguientes pasos

Paso 1: Crear el secreto

El primer paso consiste en copiar las credenciales codificadas existentes en un secreto en Secrets Manager. Para obtener la menor latencia, guarde el secreto en la misma región que la base de datos.

Para crear un secreto

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija Almacenar un secreto nuevo.
3. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:
   a. En Secret type (Tipo secreto), elija el tipo de credenciales de base de datos que desea almacenar:
      - Bases de datos de Amazon RDS
      - Base de datos de Amazon DocumentDB
      - Almacén de datos de Amazon Redshift.
      - Para otro tipo de secretos, consulte Reemplazar secretos codificados .
   b. En Credenciales, ingrese las credenciales existentes para la base de datos.
   c. Para Clave encriptada, seleccione aws/secretsmanager para utilizar Clave administrada de AWS para Secrets Manager. No se aplica ningún cargo por el uso de esta clave. También puede utilizar su propia clave administrada por el cliente, por ejemplo, para acceder al secreto desde otro Cuenta de AWS. Para obtener información sobre los costos por usar una clave administrada por el cliente, consulte Precios.
   d. En Database (Base de datos), elija la base de datos.
   e. Elija Siguiente.
4. En la página Configure secret (Configurar el secreto), haga lo siguiente:
   a. Ingrese un Nombre de secreto descriptivo y una Descripción.
   b. En Permisos de recursos, seleccione Edit permissions (Editar permisos). Pegue la siguiente política, que permite *RoleToRetrieveSecretAtRuntime*recuperar el secreto y, a continuación, seleccione Guardar.

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

```
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:role/RoleToRetrieveSecretAtRuntime"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

c. En la parte inferior de la página, elija Siguiente.

5. En la página Configure rotation (Configurar rotación), mantenga la rotación desactivada por ahora. La activará más tarde. Elija Siguiente.
6. En la página Review (Revisar), revise los detalles del secreto y, a continuación, elija Store (Almacenar).

Paso 2: Actualización del código

El código debe asumir la función de IAM *RoleToRetrieveSecretAtRuntime*para poder recuperar el secreto. Para obtener más información, consulte Cambiar a un rol de IAM (AWS API).

A continuación, actualice el código para recuperar el secreto de Secrets Manager utilizando el código de ejemplo proporcionado por Secrets Manager.

Para encontrar el código de muestra

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la página Secrets (Secretos), elija el secreto.
3. Baje hasta Código de muestra. Elija su idioma y, a continuación, copie el fragmento de código.

En la aplicación, elimine las credenciales codificadas y pegue el fragmento de código. Según el idioma del código, es posible que tenga que añadir una llamada a la función o método del fragmento.

Compruebe que su aplicación funciona según lo esperado con el secreto en lugar de las credenciales codificadas.

Paso 3: rote el secreto

El último paso es anular las credenciales codificadas rotando el secreto. La rotación es el proceso de actualización periódica de un secreto. Cuando se rota un secreto, se actualizan las credenciales tanto en el secreto como en la base de datos. Secrets Manager puede configurar la rotación de un secreto automáticamente en el horario que usted establezca.

Parte de la configuración de la rotación consiste en garantizar que la función de rotación de Lambda pueda acceder tanto a Secrets Manager como a su base de datos. Cuando activa la rotación automática, Secrets Manager crea la función de rotación Lambda en la misma VPC que la base de datos para que tenga acceso en red a la base de datos. La función de rotación de Lambda también debe poder realizar llamadas a Secrets Manager para actualizar el secreto. Le recomendamos que cree un punto final de Secrets Manager en la VPC para que las llamadas de Lambda a Secrets Manager no salgan de la infraestructura. AWS Para ver instrucciones, consulte Punto de conexión VPC.

Activar la rotación

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la página Secrets (Secretos), elija el secreto.
3. En la página Secret details (Detalles del secreto), en la sección Rotation configuration (Configuración de la rotación), elija Edit rotation (Editar rotación).
4. En el cuadro de diálogo Edit rotation configuration (Configuración para editar la rotación), haga lo siguiente:

    a. Active Automatic rotation (Rotación automática).

    b. En la sección Programación de rotación, ingrese su horario en la zona horaria UTC.

    c. Elija Rotate immediately when the secret is stored (Rotar inmediatamente cuando se almacene el secreto) a fin de rotar su secreto en cuanto guarde los cambios.

    d. En la sección Función de rotación, seleccione Create a new Lambda function (Crear una nueva función de Lambda) e ingrese un nombre para la nueva función. Secrets Manager agrega “SecretsManager” al principio del nombre de la función.

    e. Para la estrategia de rotación, elija un solo usuario.

    f. Seleccione Guardar.

Para comprobar que el secreto ha rotado

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Seleccione Secrets (Secretos) y luego elija el secreto.
3. En la página Secret details (Detalles del secreto), deslícese hacia abajo y elija Retrieve secret value (Recuperar valor del secreto).

   Si el valor secreto ha cambiado, la rotación se realizó correctamente. Si el valor secreto no ha cambiado, consulte Solución de problemas de rotación de los CloudWatch registros para ver la función de rotación.

Compruebe que su aplicación funciona según lo esperado con el secreto rotado.

Siguientes pasos

A continuación, algunas ideas a tener en cuenta después de eliminar un secreto codificado de su código:

- Puede mejorar el rendimiento y reducir los costos almacenando secretos en caché. Para obtener más información, consulte Recuperar secretos.
- Puede elegir un programa de rotación diferente. Para obtener más información, consulte the section called “Programación de expresiones”.
- Para encontrar secretos codificados en sus aplicaciones de Java y Python, le recomendamos Amazon CodeGuru Reviewer.

Configuración de rotación de usuarios alternativos para AWS Secrets Manager

En este tutorial, aprenderá a configurar la rotación de usuarios alternativos para un secreto que contiene credenciales de bases de datos. La rotación de usuarios alternativos es una estrategia de rotación en la que Secrets Manager clona al usuario y, luego, alterna las credenciales del usuario que se actualizan. Esta estrategia es una buena opción si necesita disponibilidad alta para su secreto, ya que uno de los usuarios alternativos tiene credenciales actuales para la base de datos mientras que el otro se actualiza. Para obtener más información, consulte the section called “Usuarios alternativos”.

Para configurar la rotación de usuarios alternativos, necesita dos secretos:

- Un secreto con las credenciales que desea rotar.
- Un segundo secreto que tiene credenciales de administrador.

  Este usuario tiene permisos para clonar al primer usuario y cambiar la contraseña del primer usuario. En este tutorial, debe hacer que Amazon RDS cree este secreto para un usuario administrador. Amazon RDS también administra la rotación de contraseñas de administrador. Para obtener más información, consulte the section called “Rotación administrada”.

La primera parte de este tutorial consiste en configurar un entorno realista. Para mostrar cómo funciona la rotación, este tutorial incluye un ejemplo de base de datos MySQL en Amazon RDS. Por seguridad, la base de datos se encuentra en una VPC que limita el acceso entrante desde Internet. Para conectarse a la base de datos desde su computadora local a través de Internet, utilice un host bastión, un servidor de la VPC que se puede conectar a la base de datos y que también permite conexiones SSH desde Internet. El host bastión de este tutorial es una instancia de Amazon EC2, y los grupos de seguridad de la instancia impiden otros tipos de conexiones.

Una vez terminado el tutorial, le recomendamos que limpie los recursos del tutorial. No los utilice en un entorno de producción.

La rotación de Secrets Manager utiliza una función de AWS Lambda para actualizar el secreto y la base de datos. Para obtener información sobre los costos por usar una función de Lambda, consulte Precios.

Tutorial:

Permisos

Para los requisitos previos del tutorial, necesita permisos administrativos para su Cuenta de AWS. En una configuración de producción, una práctica recomendada es utilizar diferentes roles para cada

uno de los pasos. Por ejemplo, un rol con permisos de administrador de bases de datos creará la base de datos de Amazon RDS, y un rol con permisos de administrador de red configurará la VPC y los grupos de seguridad. Para los pasos del tutorial, le recomendamos que siga utilizando la misma identidad.

Para obtener más información sobre cómo configurar permisos en un entorno de producción, consulte Autenticación y control de acceso.

Requisitos previos

Para este tutorial, necesita lo siguiente:

- Requisito previo A: Amazon VPC
- Requisito previo B: instancia de Amazon EC2
- Requisito previo C: base de datos de Amazon RDS y un secreto de Secrets Manager para las credenciales de administrador
- Requisito previo D: permita que su equipo local se conecte a la instancia de EC2

Requisito previo A: Amazon VPC

En este paso, cree una VPC en la que pueda lanzar una base de datos de Amazon RDS y una instancia de Amazon EC2. En un paso posterior, utilizará su computadora para conectarse a través de Internet al bastión y, después, a la base de datos, por lo que tendrá que permitir que el tráfico salga de la VPC. Para ello, Amazon VPC adjunta una puerta de enlace de Internet a la VPC y agrega una ruta en la tabla de enrutamiento de manera que el tráfico destinado fuera de la VPC se envíe a la puerta de enlace de Internet.

Dentro de la VPC, se crean un punto de conexión de Secrets Manager y otro de Amazon RDS. Cuando configure la rotación automática en un paso posterior, Secrets Manager creará la función de rotación de Lambda en la VPC para que tenga acceso a la base de datos. La función de rotación de Lambda también llama a Secrets Manager para actualizar el secreto y a Amazon RDS para obtener la información de conexión a la base de datos. Al crear puntos de conexión en la VPC, se asegura de que las llamadas de la función de Lambda a Secrets Manager y Amazon RDS no salgan de la infraestructura de AWS. En su lugar, se dirigen a puntos de conexión dentro de la VPC.

Para crear una VPC

1. Abra la consola de Amazon VPC en https://console.aws.amazon.com/vpc/.
2. Seleccione Create VPC (Crear VPC).

3. En la página Create VPC (Crear VPC), seleccione VPC and more (VPC y más).
4. En Name tag auto-generation (Generación automática de etiquetas de nombre), ingrese **`SecretsManagerTutorial`** en Auto-generate (Generar automáticamente).
5. En DNS options (Opciones de DNS), elija **`Enable DNS hostnames`** y **`Enable DNS resolution`**.
6. Seleccione Create VPC (Crear VPC).

Para crear un punto de conexión de Secrets Manager dentro de la VPC

1. En la consola de Amazon VPC, en Endpoints (Puntos de conexión), elija Create Endpoint (Crear punto de conexión).
2. En Endpoint settings (Configuración de punto de conexión), ingrese **`SecretsManagerTutorialEndpoint`** en Name (Nombre).
3. En Services (Servicios), ingrese **`secretsmanager`** para filtrar la lista y, luego, seleccione el punto de conexión de Secrets Manager en su Región de AWS. Por ejemplo, en Este de EE. UU. (Norte de Virginia), elija `com.amazonaws.us-east-1.secretsmanager`.
4. En VPC, elija **`vpc**** (SecretsManagerTutorial)`**.
5. En Subnets (Subredes), seleccione todas las Availability Zones (Zonas de disponibilidad) y, luego, para cada una, elija un Subnet ID (ID de subred) para incluir.
6. En IP address type ((Tipo de dirección IP), elija **`IPv4`**.
7. En Security groups (Grupos de seguridad), elija el grupo de seguridad predeterminado.
8. En Policy (Política), elija **`Full access`**.
9. Elija Crear punto de conexión.

Para crear un punto de conexión de Amazon RDS dentro de la VPC

1. En la consola de Amazon VPC, en Endpoints (Puntos de conexión), elija Create Endpoint (Crear punto de conexión).
2. En Endpoint settings (Configuración de punto de conexión), ingrese **`RDSTutorialEndpoint`** en Name (Nombre).
3. En Services (Servicios), ingrese **`rds`** para filtrar la lista y, luego, seleccione el punto de conexión de Amazon RDS en su Región de AWS. Por ejemplo, en Este de EE. UU. (Norte de Virginia), elija `com.amazonaws.us-east-1.rds`.
4. En VPC, elija **`vpc**** (SecretsManagerTutorial)`**.

5. En Subnets (Subredes), seleccione todas las Availability Zones (Zonas de disponibilidad) y, luego, para cada una, elija un Subnet ID (ID de subred) para incluir.
6. En IP address type ((Tipo de dirección IP), elija **`IPv4`**.
7. En Security groups (Grupos de seguridad), elija el grupo de seguridad predeterminado.
8. En Policy (Política), elija **`Full access`**.
9. Elija Crear punto de conexión.

Requisito previo B: instancia de Amazon EC2

La base de datos de Amazon RDS que cree en un paso posterior estará en la VPC, por lo que para acceder a ella necesitará un host bastión. El host bastión también está en la VPC, pero en un paso posterior, configurará un grupo de seguridad para permitir que su equipo local se conecte al host bastión con SSH.

Para crear una instancia de EC2 para un host bastión

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. Elija Instances (Instancias) y, luego, elija Launch Instances (Lanzar instancias).
3. En Name and tags (Nombre y etiquetas), en Name (Nombre), introduzca **`SecretsManagerTutorialInstance`**.
4. En Application and OS Images (Imágenes de aplicaciones y sistemas operativos), mantenga el valor predeterminado **`Amazon Linux 2 AMI (HMV) Kernel 5.10`**.
5. En Instance type (Tipo de instancia), mantenga el valor predeterminado **`t2.micro`**.
6. En Key pair (Par de claves), seleccione Create key pair (Crear par de claves).

   En el cuadro de diálogo Create key pair (Crear par de claves), en Key pair name (Nombre del par de claves), ingrese **`SecretsManagerTutorialKeyPair`** y haga clic en Create (Crear).

   La clave privada se descarga automáticamente.
7. En Network settings (Configuración de red), elija Edit (Editar) y realice lo siguiente:

   a. En VPC, elija **`vpc-**** SecretsManagerTutorial`**.

   b. En Auto-assign Public IP (Asignar IP pública automáticamente), elija **`Enable`**.

   c. En Firewall, seleccione Select existing security group (Seleccionar grupo de seguridad existente).

d. En Common security groups (Grupos de seguridad comunes), elija **`default`**.

8. Seleccione Launch Instance (Lanzar instancia).

Requisito previo C: base de datos de Amazon RDS y un secreto de Secrets Manager para las credenciales de administrador

En este paso, cree una base de datos MySQL de Amazon RDS y configúrela de manera que Amazon RDS cree un secreto que contenga las credenciales de administrador. A continuación, Amazon RDS gestionará automáticamente la rotación del secreto de administrador por usted. Para obtener más información, consulte Rotación administrada.

Como parte de la creación de la base de datos, debe especificar el host bastión que creó en el paso anterior. A continuación, Amazon RDS configura grupos de seguridad para que la base de datos y la instancia puedan acceder entre sí. Agregue una regla al grupo de seguridad adjunto a la instancia para permitir que su equipo local también se conecte a ella.

Para crear una base de datos de Amazon RDS con un secreto de Secrets Manager que contenga las credenciales de administrador

1. En la consola de Amazon RDS, seleccione Create database (Crear base de datos).
2. En la sección Engine options (Opciones del motor), en Engine type (Tipo de motor) elija **`MySQL`**.
3. En la sección Templates (Plantillas), elija **`Free tier`**.
4. En la sección Settings (Configuración), realice lo siguiente:

   a. En DB instance identifier (Identificador de instancia de base de datos), ingrese **`SecretsManagerTutorial`**.

   b. En Configuración de credenciales, seleccione Administrar credenciales maestras en AWS Secrets Manager.

5. En la sección Connectivity (Conectividad), para Computer resource (Recurso de equipo), elija Connect to an EC2 computer resource (Conectarse a un recurso de equipo de EC2) y, a continuación, para EC2 Instance (Instancia de EC2), elija **`SecretsManagerTutorialInstance`**.
6. Elija Create database (Crear base de datos).

Requisito previo D: permita que su equipo local se conecte a la instancia de EC2

En este paso, configurará la instancia de EC2 que creó en el requisito previo B para permitir que su equipo local se conecte a ella. Para ello, edite el grupo de seguridad que Amazon RDS agregó al requisito previo C para incluir una regla que permita que la dirección IP de su equipo se conecte con SSH. La regla permite que su equipo local (identificado por su dirección IP actual) se conecte al host bastión mediante SSH a través de Internet.

Para permitir que su equipo local se conecte a la instancia de EC2

1. Abra la consola de Amazon EC2 en https://console.aws.amazon.com/ec2/.
2. En la instancia de EC2 SecretsManagerTutorialInstance, en la pestaña Security (Seguridad), en Security groups (Grupos de seguridad), elija **`sg-*** (ec2-rds-X)`**.
3. En la pestaña Inbound rules (Reglas de entrada), seleccione Edit inbound rules (Editar reglas de entrada).
4. Elija Add Rule (Agregar regla) y, a continuación para la regla, haga lo siguiente:
   a. En Type (Tipo), elija **`SSH`**.
   b. En Tipo de origen, elija **`My IP`**.

Paso 1: cree un usuario de base de datos de Amazon RDS

Primero, necesita un usuario cuyas credenciales se almacenarán en el secreto. Para crear el usuario, inicie sesión en la base de datos de Amazon RDS con las credenciales de administrador. Para simplificar, en el tutorial se crea un usuario con todos los permisos para acceder a una base de datos. En un entorno de producción, esto no es habitual y le recomendamos que siga el principio de privilegio mínimo.

Para conectarse a la base de datos, utilizará una herramienta de cliente de MySQL. En este tutorial, utilizará MySQL Workbench, una aplicación basada en la interfaz gráfica de usuario (GUI). Para instalar MySQL Workbench, consulte Download MySQL Workbench (Descargar MySQL Workbench).

Para conectarse a la base de datos, cree una configuración de conexión en MySQL Workbench. Para la configuración, necesita información de Amazon EC2 y Amazon RDS.

Para crear una conexión de base de datos en MySQL Workbench

1. En MySQL Workbench, junto a MySQL Connections (Conexiones de MySQL), elija el botón (+).

2. En el cuadro de diálogo Setup New Connection (Configurar una conexión), haga lo siguiente:

   a. En Connection Name (Nombre de conexión), ingrese **`SecretsManagerTutorial`**.

   b. En Connection Method (Método de conexión), elija **`Standard TCP/IP over SSH`**.

   c. En la pestaña Parameters (Parámetros), haga lo siguiente:

      i. En SSH Hostname (Nombre de host SSH), ingrese la dirección IP pública de la instancia de Amazon EC2.

         Podrá encontrar la dirección IP en la consola de Amazon EC2 si elige la instancia SecretsManagerTutorialInstance. Copie la dirección IP en Public IPv4 DNS (DNS IPv4 público).

      ii. En SSH Username (Nombre de usuario SSH), ingrese **`ec2-user`**.

      iii. En SSH Keyfile (Archivo de claves SSH), elija el archivo de par de claves SecretsManagerTutorialKeyPair.pem que descargó en el requisito previo anterior.

      iv. En MySQL Hostname (Nombre de host de MySQL), ingrese la dirección del punto de conexión de Amazon RDS.

         Podrá encontrar la dirección del punto de conexión en la consola de Amazon RDS si elige la instancia de base de datos secretsmanagertutorialdb. Copie la dirección en Endpoint (Punto de conexión).

      v. En Username (Nombre de usuario), ingrese **`admin`**.

   d. Seleccione OK (Aceptar).

Para recuperar la contraseña de administrador

1. En la consola de Amazon RDS, acceda a su base de datos.
2. En la pestaña Configuration (Configuración), en Master Credentials ARN (ARN de credenciales maestras), seleccione Manage in Secrets Manager (Administrar en Secrets Manager).

   Se abrirá la consola de Secrets Manager.
3. En la página de detalles del secreto, elija Retrieve secret value (Recuperar valor del secreto).
4. La contraseña aparece en la sección Secret value (Valor secreto).

Para crear un usuario de base de datos

1. En MySQL Workbench, elija la conexión SecretsManagerTutorial.
2. Ingrese la contraseña de administrador que recuperó del secreto.
3. En MySQL Workbench, en la ventana Query (Consulta), ingrese los siguientes comandos (incluida una contraseña segura) y, luego, elija Execute (Ejecutar).

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

   En la ventana Output (Salida), observará que los comandos se ejecutaron correctamente.

Paso 2: cree un secreto para las credenciales del usuario

A continuación, crea un secreto para almacenar las credenciales del usuario que acaba de crear. Este es el secreto que rotará. Activa la rotación automática y, para indicar la estrategia de usuarios alternativos, elige un secreto de superusuario independiente que tenga permiso para cambiar la contraseña del primer usuario.

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija Store a new secret (Almacenar un nuevo secreto).
3. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:
   a. En Secret type (Tipo de secreto), elija Credentials for Amazon RDS database (Credenciales para base de datos de Amazon RDS).
   b. En Credentials (Credenciales), ingrese el nombre de usuario **`appuser`** y la contraseña que ingresó para el usuario de base de datos que creó mediante MySQL Workbench.
   c. En Database (Base de datos), elija secretsmanagertutorialdb.
   d. Elija Siguiente.
4. En la página Configure secrect (Configurar el secreto), en Secret name (Nombre del secreto), ingrese **`SecretsManagerTutorialAppuser`** y, luego, elija Next (Siguiente).
5. En la página Configure rotation (Configurar la rotación), haga lo siguiente:
   a. Active Automatic rotation (Rotación automática).

b. En Rotation schedule (Programación de rotación), configure una programación de Days (Días): **2** días con Duration (Duración): **2h**. Mantenga seleccionada la opción Rotate immediately (Rotar inmediatamente).

c. En Rotation function (Función de rotación), elija Create a rotation function (Crear una función de rotación) y, luego, para el nombre de la función, ingrese **`tutorial-alternating-users-rotation`**.

d. En Utilizar credenciales individuales, elija Sí, y luego en Secretos, elija el secreto llamado rds!cluster… que tiene una Descripción que incluye el nombre de la base de datos que creó en este tutorial **`secretsmanagertutorial`**, como `Secret associated with primary RDS DB instance: arn:aws:rds:`*`Region`*`:`*`AccountId`*`:db:secretsmanagertutorial`.

e. Elija Siguiente.

6. En la página Review (Revisar), elija Store (Almacenar).

   Secrets Manager vuelve a la página de detalles del secreto. En la parte superior de la página, puede observar el estado de la configuración de la rotación. Secrets Manager utiliza CloudFormation para crear recursos como la función de rotación de Lambda y un rol de ejecución que ejecuta la función Lambda. Cuando CloudFormation termina, el banner cambia a Secret scheduled for rotation (Secreto programado para rotación). Se completó la primera rotación.

Paso 3: pruebe el secreto rotado

Una vez que el secreto se ha rotado, puede comprobar que contenga nuevas credenciales válidas. La contraseña del secreto cambió con respecto a las credenciales originales.

Para recuperar la contraseña nueva del secreto

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija Secrets (Secretos) y, luego, elija el secreto **`SecretsManagerTutorialAppuser`**.
3. En la página Secret details (Detalles del secreto), deslícese hacia abajo y elija Retrieve secret value (Recuperar valor del secreto).
4. En la tabla Key/value (Clave/valor), copie el Secret value (Valor del secreto) en **`password`**.

Para probar las credenciales

1. En MySQL Workbench, haga clic con el botón derecho en la conexión SecretsManagerTutorial y, luego, elija Edit Connection (Editar conexión).
2. En el cuadro de diálogo Manage Server Connections (Administrar conexiones de servidores), en Username (Nombre de usuario), ingrese **`appuser`** y, luego, elija Close (Cerrar).
3. De vuelta en MySQL Workbench, elija la conexión SecretsManagerTutorial.
4. En el cuadro de diálogo Open SSH Connection (Conexión SSH abierta), en Password (Contraseña), pegue la contraseña que recuperó del secreto y, luego, elija OK (Aceptar).

   Si las credenciales son válidas, MySQL Workbench abrirá la página de diseño de la base de datos.

Esto indica que la rotación del secreto se realizó correctamente. Las credenciales del secreto se actualizaron y es una contraseña válida para conectarse a la base de datos.

Paso 4: limpie los recursos

Si desea probar otra estrategia de rotación, la rotación de un solo usuario, omita la eliminación de recursos y diríjase a the section called “Rotación de un solo usuario”.

De lo contrario, para evitar posibles cargos y eliminar la instancia de EC2 que tiene acceso a Internet, elimine los siguientes recursos que creó en este tutorial y los requisitos previos:

- Instancia de base de datos de Amazon RDS. Para obtener instrucciones, consulte Deleting a DB instance (Eliminar una instancia de base de datos) en la Guía del usuario de Amazon RDS.
- Instancia de Amazon EC2. Para obtener más información, consulte Terminate an instance (Terminar una instancia) en la Guía del usuario de Amazon EC2 para instancias de Linux.
- Secreto `SecretsManagerTutorialAppuser` de Secrets Manager. Para obtener instrucciones, consulte the section called “Eliminar un secreto”.
- Punto de conexión de Secrets Manager. Para obtener instrucciones, consulte Delete a VPC endpoint (Eliminar un punto de conexión de VPC) en la Guía de AWS PrivateLink.
- Punto de conexión de VPC. Para obtener instrucciones, consulte Delete your VPC (Eliminar su VPC) en la Guía de AWS PrivateLink.

Pasos siguientes

- Obtenga información sobre cómo recuperar secretos en sus aplicaciones.
- Obtenga más información sobre otras programaciones de rotación.

Configuración de la rotación de un solo usuario para AWS Secrets Manager

En este tutorial, aprenderá a configurar la rotación de un solo usuario para un secreto que contiene credenciales de bases de datos. La rotación de un solo usuario es una estrategia de rotación en la que Secrets Manager actualiza las credenciales de un usuario tanto en el secreto como en la base de datos. Para obtener más información, consulte the section called “Un solo usuario”.

Una vez terminado el tutorial, le recomendamos que limpie los recursos del tutorial. No los utilice en un entorno de producción.

La rotación de Secrets Manager utiliza una función de AWS Lambda para actualizar el secreto y la base de datos. Para obtener información sobre los costos por usar una función de Lambda, consulte Precios.

Contenido

Permisos

Para los requisitos previos del tutorial, necesita permisos administrativos para su Cuenta de AWS. En una configuración de producción, una práctica recomendada es utilizar diferentes roles para cada uno de los pasos. Por ejemplo, un rol con permisos de administrador de bases de datos creará la base de datos de Amazon RDS, y un rol con permisos de administrador de red configurará la VPC y

los grupos de seguridad. Para los pasos del tutorial, le recomendamos que siga utilizando la misma identidad.

Para obtener más información sobre cómo configurar permisos en un entorno de producción, consulte Autenticación y control de acceso.

Requisitos previos

El requisito previo para este tutorial es the section called “Estrategia de rotación de usuarios alternativos”. No limpie los recursos al final del primer tutorial. Después de ese tutorial, tendrá un entorno realista con una base de datos de Amazon RDS y un secreto en Secrets Manager que contiene las credenciales de administrador para la base de datos. También tiene un segundo secreto que contiene las credenciales de un usuario de base de datos, pero no utilizará ese secreto en este tutorial.

También cuenta con una conexión configurada en MySQL Workbench para conectarse a la base de datos con las credenciales de administrador.

Paso 1: cree un usuario de base de datos de Amazon RDS

Primero, necesita un usuario cuyas credenciales se almacenarán en el secreto. Para crear el usuario, inicie sesión en la base de datos de Amazon RDS con las credenciales de administrador almacenadas en un secreto. Para simplificar, en el tutorial se crea un usuario con todos los permisos para acceder a una base de datos. En un entorno de producción, esto no es habitual y le recomendamos que siga el principio de privilegio mínimo.

Para recuperar la contraseña de administrador

1. En la consola de Amazon RDS, acceda a su base de datos.
2. En la pestaña Configuration (Configuración), en Master Credentials ARN (ARN de credenciales maestras), seleccione Manage in Secrets Manager (Administrar en Secrets Manager).

   Se abrirá la consola de Secrets Manager.
3. En la página de detalles del secreto, elija Retrieve secret value (Recuperar valor del secreto).
4. La contraseña aparece en la sección Secret value (Valor secreto).

Para crear un usuario de base de datos

1. En MySQL Workbench, haga clic con el botón derecho en la conexión SecretsManagerTutorial y, luego, elija Edit Connection (Editar conexión).
2. En el cuadro de diálogo Manage Server Connections (Administrar conexiones de servidores), en Username (Nombre de usuario), ingrese **admin** y, luego, elija Close (Cerrar).
3. De vuelta en MySQL Workbench, elija la conexión SecretsManagerTutorial.
4. Ingrese la contraseña de administrador que recuperó del secreto.
5. En MySQL Workbench, en la ventana Query (Consulta), ingrese los siguientes comandos (incluida una contraseña segura) y, luego, elija Execute (Ejecutar).

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

   En la ventana Output (Salida), observará que los comandos se ejecutaron correctamente.

Paso 2: cree un secreto para las credenciales del usuario de base de datos

A continuación, cree un secreto para almacenar las credenciales del usuario que acaba de crear y active la rotación automática, incluida la rotación inmediata. Secrets Manager rotará el secreto, lo que significa que la contraseña se genera mediante programación; ninguna persona ha visto esta nueva contraseña. Hacer que la rotación comience inmediatamente también puede ayudarlo a determinar si la rotación está configurada de manera correcta.

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija Store a new secret (Almacenar un nuevo secreto).
3. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:
   a. En Secret type (Tipo de secreto), elija Credentials for Amazon RDS database (Credenciales para base de datos de Amazon RDS).
   b. En Credentials (Credenciales), ingrese el nombre de usuario **dbuser** y la contraseña que ingresó para el usuario de base de datos que creó mediante MySQL Workbench.
   c. En Database (Base de datos), elija secretsmanagertutorialdb.
   d. Elija Siguiente.

4. En la página Configure secrect (Configurar el secreto), en Secret name (Nombre del secreto), ingrese **`SecretsManagerTutorialDbuser`** y, luego, elija Next (Siguiente).
5. En la página Configure rotation (Configurar la rotación), haga lo siguiente:
   a. Active Automatic rotation (Rotación automática).
   b. En Rotation schedule (Programación de rotación), configure una programación de Days (Días): **2** días con Duration (Duración): **2h**. Mantenga seleccionada la opción Rotate immediately (Rotar inmediatamente).
   c. En Rotation function (Función de rotación), elija Create a rotation function (Crear una función de rotación) y, luego, para el nombre de la función, ingrese **`tutorial-single-user-rotation`**.
   d. Para la estrategia de rotación, elija un solo usuario.
   e. Elija Siguiente.
6. En la página Review (Revisar), elija Store (Almacenar).

   Secrets Manager vuelve a la página de detalles del secreto. En la parte superior de la página, puede observar el estado de la configuración de la rotación. Secrets Manager utiliza CloudFormation para crear recursos como la función de rotación de Lambda y un rol de ejecución que ejecuta la función Lambda. Cuando CloudFormation termina, el banner cambia a Secret scheduled for rotation (Secreto programado para rotación). Se completó la primera rotación.

Paso 3: pruebe la contraseña rotada

Después de la primera rotación del secreto, que puede tardar unos segundos, puede comprobar que el secreto siga conteniendo credenciales válidas. La contraseña del secreto cambió con respecto a las credenciales originales.

Para recuperar la contraseña nueva del secreto

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija Secrets (Secretos) y, luego, elija el secreto **`SecretsManagerTutorialDbuser`**.
3. En la página Secret details (Detalles del secreto), deslícese hacia abajo y elija Retrieve secret value (Recuperar valor del secreto).
4. En la tabla Key/value (Clave/valor), copie el Secret value (Valor del secreto) en **`password`**.

Para probar las credenciales

1. En MySQL Workbench, haga clic con el botón derecho en la conexión SecretsManagerTutorial y, luego, elija Edit Connection (Editar conexión).
2. En el cuadro de diálogo Manage Server Connections (Administrar conexiones de servidores), en Username (Nombre de usuario), ingrese **`dbuser`** y, luego, elija Close (Cerrar).
3. De vuelta en MySQL Workbench, elija la conexión SecretsManagerTutorial.
4. En el cuadro de diálogo Open SSH Connection (Conexión SSH abierta), en Password (Contraseña), pegue la contraseña que recuperó del secreto y, luego, elija OK (Aceptar).

   Si las credenciales son válidas, MySQL Workbench abrirá la página de diseño de la base de datos.

Paso 4: limpie los recursos

Para evitar posibles cargos, elimine el secreto que creó en este tutorial. Para obtener instrucciones, consulte the section called “Eliminar un secreto”.

Para limpiar los recursos creados en el tutorial anterior, consulte the section called “Paso 4: limpie los recursos”.

Pasos siguientes

- Obtenga información sobre cómo recuperar secretos en sus aplicaciones. Consulte Recuperar secretos.
- Obtenga más información sobre otras programaciones de rotación. Consulte the section called “Programación de expresiones”.

Autenticación y control de acceso de AWS Secrets Manager

Secrets Manager utiliza AWS Identity and Access Management (IAM) para asegurar el acceso a los secretos. IAM proporciona autenticación y control de acceso. La autenticación verifica la identidad de las personas que realizan solicitudes. Secrets Manager utiliza un proceso de inicio de sesión con contraseñas, claves de acceso y token de autenticación multifactor (MFA) para verificar la identidad de los usuarios. Consulte Inicio de sesión en AWS. El control de acceso garantiza que solo las personas autorizadas puedan realizar operaciones en los recursos de AWS tales como los secretos. Secrets Manager utiliza políticas para definir quién tiene acceso a qué recursos y qué acciones puede realizar la identidad sobre esos recursos. Consulte Políticas y permisos en IAM.

Puede utilizar Funciones de AWS Identity and Access Management en cualquier lugar para obtener credenciales de seguridad temporales en IAM para cargas de trabajo, como servidores, contenedores y aplicaciones que se ejecutan fuera de AWS. Sus cargas de trabajo pueden usar las mismas políticas de IAM y los mismos roles de IAM que utiliza con las aplicaciones de AWS para acceder a los recursos de AWS. Con Funciones de IAM en cualquier lugar, puede usar Secrets Manager para almacenar y administrar las credenciales a las que pueden acceder los recursos en AWS, así como los dispositivos en las instalaciones, como los servidores de aplicaciones. Para obtener más información, consulte la Guía del usuario de Funciones de IAM en cualquier lugar.

Permisos de Secrets Manager

Para conceder permisos de administrador a Secrets Manager, siga las instrucciones en Agregar y eliminar permisos de identidad de IAM y adjunte las siguientes políticas:

- SecretsManagerReadWrite
- IAMFullAccess

Le recomendamos que no otorgue permisos de administrador a los usuarios finales. Aunque esto le permite a sus usuarios crear y administrar sus secretos, el permiso necesario para habilitar la rotación (IAMFullAccess) otorga permisos significativos que no son adecuados para los usuarios finales.

Permisos para acceder a secretos

Mediante la utilización las políticas de permisos de IAM, puede controlar qué usuarios o servicios obtienen acceso a los secretos. Una política de permisos describe quién puede realizar qué acciones en qué recursos. Puede hacer lo siguiente:

- the section called “Adjuntar una política de permisos a una identidad”
- the section called “Adjuntar una política de permisos a un secreto”

Permisos para las funciones de rotación de Lambda

Secrets Manager utiliza funciones de AWS Lambda para rotar los secretos. La función de Lambda debe tener acceso al secreto, así como también a la base de datos o servicio para el que el secreto contiene las credenciales. Consulte Permisos para rotación.

Permisos para claves de cifrado

Secrets Manager utiliza las claves de AWS Key Management Service (AWS KMS) para cifrar los secretos. La Clave administrada de AWS `aws/secretsmanager` tiene los permisos correctos automáticamente. Si utiliza una clave KMS diferente, el Secrets Manager necesita permisos para esa clave. Consulte the section called “Permisos para la clave KMS”.

Adjuntar una política de permisos a una identidad

Puede adjuntar políticas de permisos a las identidades, usuarios, grupos, roles, servicios y recursos de IAM. En una política basada en la identidad, se especifica a qué secretos tiene acceso la identidad y las acciones que la identidad puede realizar en los secretos. Para obtener más información, consulte Añadir y eliminar permisos de identidad de IAM.

Puede conceder permisos a un rol que representa a una aplicación o usuario en otro servicio. Por ejemplo, una aplicación que se ejecuta en una instancia de Amazon EC2 puede necesitar acceso a una base de datos. Puede crear un rol de IAM asociado al perfil de instancia de EC2 y, a continuación, utilizar una política de permisos para conceder al rol acceso al secreto que contiene las credenciales para la base de datos. Para obtener más información, consulte Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias de Amazon EC2. Otros servicios a los que puede adjuntar roles para incluir Amazon Redshift, AWS Lambda, y Amazon ECS.

Puede conceder permisos a usuarios autenticados por un sistema de identidad distinto de IAM. Por ejemplo, puede asociar roles de IAM a usuarios de aplicaciones móviles que inician sesión con Amazon Cognito. El rol concede credenciales temporales a la aplicación con los permisos en la política de permisos del rol. A continuación, puede utilizar una política de permisos para conceder al rol acceso al secreto. Para obtener más información, consulte Proveedores de identidad y federación.

Puede utilizar políticas basadas en identidad para:

- Conceder acceso por identidad a varios secretos.
- Controlar quién puede crear nuevos secretos y quién puede acceder a secretos que aún no se han creado.
- Conceder a un grupo de IAM acceso a secretos.

Para obtener más información, consulte the section called “Ejemplos de políticas de permisos”.

Adición de una política de permisos a un secreto de AWS Secrets Manager

En una política basada en recursos, usted especifica quién puede obtener acceso al secreto y las acciones que puede realizar en él. Puede utilizar políticas basadas en recursos para:

- Conceder acceso a un solo secreto a varios usuarios o roles.
- Conceda acceso a usuarios o roles en otras AWS cuentas.

Consulte the section called “Ejemplos de políticas de permisos”.

Al adjuntar una política basada en recursos a un secreto en la consola, Secrets Manager utiliza el motor de razonamiento automatizado Zelkova y la API `ValidateResourcePolicy` para evitar que pueda conceder a una amplia gama de principales de IAM acceso a sus secretos. También puede llamar a la API de `PutResourcePolicy` con el parámetro `BlockPublicPolicy` desde la CLI o el SDK.

> **Important**
>
> La validación de la política de recursos y el `BlockPublicPolicy` parámetro ayudan a proteger tus recursos al impedir que se conceda el acceso público a través de las políticas

de recursos que están directamente asociadas a tus secretos. Además de utilizar estas funciones, inspeccione detenidamente las siguientes políticas para confirmar que no conceden acceso público:

- Políticas basadas en la identidad asociadas a los AWS directores asociados (por ejemplo, las funciones de IAM)
- Políticas basadas en recursos asociadas a los AWS recursos asociados (por ejemplo, claves ()) AWS Key Management Service AWS KMS

Para revisar los permisos de sus datos secretos, consulte. Determinación de quién tiene permisos para los secretos de

Ver, cambiar o eliminar la política de recursos de un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto.
3. En la página de detalles secretos, en la pestaña Descripción general, en la sección Permisos de recursos, seleccione Editar permisos.
4. En el campo de código, realice una de las siguientes operaciones y, a continuación, elija Save (Guardar):
   - Para adjuntar o modificar una política de recursos, ingrese la política.
   - Para eliminar la política, limpie el campo de código.

AWS CLI

Example Recuperar una política de recursos

En el siguiente ejemplo de `get-resource-policy` se recupera la política basada en recursos asociada a un secreto.

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

Example Eliminar una política de recursos

En el siguiente ejemplo de `delete-resource-policy` se elimina la política basada en recursos asociada a un secreto.

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

Example Agregar una política de recursos

En el siguiente ejemplo de `put-resource-policy` se agrega una política de permisos a un secreto, pero primero se comprueba que la política no proporciona un acceso amplio al secreto. La política se lee desde un archivo. Para obtener más información, consulte Carga de AWS CLI parámetros desde un archivo en la Guía del AWS CLI usuario.

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

Contenido de `mypolicy.json`:

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

AWS SDK

Para recuperar la política adjunta a un secreto, utilice `GetResourcePolicy`.

Para eliminar una política asociada a un secreto, utilice `DeleteResourcePolicy`.

Para adjuntar una política a un secreto, utilice `PutResourcePolicy`. Si ya hay una política adjunta, el comando la reemplaza por la nueva política. La política deben tener un formato como texto estructurado JSON. Consulte Estructura del documento de política JSON. Usar the section called "Ejemplos de políticas de permisos" para empezar a escribir su política.

Para obtener más información, consulte the section called "AWS SDKs".

AWS política gestionada para AWS Secrets Manager

Una política AWS administrada es una política independiente creada y administrada por AWS. AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte Políticas administradas por AWS en la Guía del usuario de IAM.

AWS política gestionada: SecretsManagerReadWrite

Esta política proporciona acceso de lectura y escritura a los recursos de Amazon RDS AWS Secrets Manager, Amazon Redshift y Amazon DocumentDB, incluido el permiso para describirlos, así como el permiso para usarlos para cifrar y descifrar secretos. AWS KMS Esta política también permite crear conjuntos de AWS CloudFormation cambios, obtener plantillas de rotación de un bucket de Amazon S3 gestionado por AWS, enumerar AWS Lambda funciones y describir las VPC de Amazon EC2. La consola necesita estos permisos para configurar la rotación con las funciones de rotación existentes.

Para crear nuevas funciones de rotación, también debe tener permiso para crear AWS CloudFormation pilas y AWS Lambda funciones de ejecución. Puede asignar la política FullAccess gestionada de IAM. Consulte Permisos para rotación.

Detalles de los permisos

Esta política incluye los siguientes permisos.

- `secretsmanager`: permite a las entidades principales realizar todas las acciones de Secrets Manager.
- `cloudformation`— Permite a los directores crear AWS CloudFormation pilas. Esto es necesario para que los directores que utilizan la consola para activar la rotación puedan crear funciones AWS CloudFormation de rotación Lambda a través de pilas. Para obtener más información, consulte the section called “Cómo Secrets Manager utiliza AWS CloudFormation”.
- `ec2`: permite a las entidades principales describir VPC de Amazon EC2. Esto es necesario para que las entidades principales que utilicen la consola puedan crear funciones de rotación en la misma VPC que la base de datos de las credenciales que están almacenando en un secreto.
- `kms`— Permite a los directores utilizar AWS KMS claves para las operaciones criptográficas. Esto es necesario para que Secrets Manager pueda cifrar y descifrar secretos. Para obtener más información, consulte the section called “Cifrado y descifrado de secretos”.
- `lambda`: permite a las entidades principales enumerar funciones de rotación de Lambda. Esto es necesario para que las entidades principales que utilicen la consola puedan seleccionar funciones de rotación existentes.
- `rds`: permite a las entidades principales describir clústeres e instancias de Amazon RDS. Esto es necesario para que las entidades principales que utilicen la consola puedan seleccionar clústeres o instancias de Amazon RDS.
- `redshift`: permite a las entidades principales describir clústeres de Amazon Redshift. Esto es necesario para que las entidades principales que utilicen la consola puedan seleccionar clústeres de Amazon Redshift.
- `redshift-serverless`— Permite a los directores describir los espacios de nombres en Amazon Redshift Serverless. Esto es necesario para que los directores que utilizan la consola puedan elegir los espacios de nombres de Amazon Redshift Serverless.
- `docdb-elastic`: permite a las entidades principales describir clústeres elásticos de Amazon DocumentDB. Esto es necesario para que las entidades principales que utilicen la consola puedan seleccionar clústeres elásticos de Amazon DocumentDB.
- `tag`: permite a las entidades principales obtener todos los recursos de la cuenta que estén etiquetados.

- `serverlessrepo`— Permite a los directores crear conjuntos de cambios. AWS CloudFormation Esto es necesario para que las entidades principales que utilicen la consola puedan crear funciones de rotación de Lambda. Para obtener más información, consulte the section called "Cómo Secrets Manager utiliza AWS CloudFormation".
- `s3`— Permite a los directores obtener objetos de un bucket de Amazon S3 gestionado por AWS. Este bucket contiene Plantillas de función de rotación de Lambda. Este permiso es necesario para que las entidades principales que utilicen la consola puedan crear funciones de rotación de Lambda basadas en las plantillas del bucket. Para obtener más información, consulte the section called "Cómo Secrets Manager utiliza AWS CloudFormation".

Para ver la política, consulte el documento de política de SecretsManagerReadWrite JSON.

Secrets Manager actualiza las políticas AWS gestionadas

Consulta los detalles sobre las actualizaciones de las políticas AWS gestionadas de Secrets Manager.

Cambio	Descripción	Fecha
SecretsManagerReadWrite: actualización de una política actual	Esta política se actualizó para permitir describir el acceso a Amazon Redshift Serverles s, de modo que los usuarios de consolas puedan elegir un espacio de nombres de Amazon Redshift Serverless al crear un secreto de Amazon Redshift.	12 de marzo de 2024
SecretsManagerReadWrite: actualización de una política actual	Esta política se actualizó para permitir describir el acceso a clústeres elásticos de Amazon DocumentDB de modo que los usuarios de la consola puedan seleccionar un clúster elástico al crear un secreto de Amazon DocumentDB.	12 de septiembre de 2023

Cambio	Descripción	Fecha
SecretsManagerReadWrite: actualización de una política actual	Esta política se actualizó para permitir describir el acceso a Amazon Redshift de modo que los usuarios de la consola puedan seleccionar un clúster de Amazon Redshift al crear un secreto de Amazon Redshift. La actualización también agregó nuevos permisos para permitir el acceso de lectura a un bucket de Amazon S3 administrado por el AWS que se almacenan las plantillas de funciones de rotación de Lambda.	24 de junio de 2020
SecretsManagerReadWrite: actualización de una política actual	Esta política se actualizó para permitir describir el acceso a clústeres de Amazon RDS de modo que los usuarios de la consola puedan seleccionar un clúster al crear un secreto de Amazon RDS.	3 de mayo de 2018
SecretsManagerReadWrite: política nueva	Secrets Manager creó una política para conceder los permisos que sean necesarios para utilizar la consola con todos los accesos de lectura/escritura a Secrets Manager.	04 de abril de 2018
Secrets Manager comenzó a realizar un seguimiento de los cambios	Secrets Manager comenzó a realizar un seguimiento de los cambios en sus políticas AWS gestionadas.	04 de abril de 2018

Determinación de quién tiene permisos para los secretos de AWS Secrets Manager

De forma predeterminada, las identidades de IAM no tienen permiso para acceder a los secretos. Al autorizar el acceso a un secreto, Secrets Manager evalúa la política basada en los recursos adjunta al secreto y todas las políticas basadas en la identidad adjuntas al usuario o rol de IAM que hace la solicitud. Para ello, Secrets Manager utiliza un proceso similar al descrito en Cómo determinar si una solicitud se permite o se deniega en la Guía del usuario de IAM.

Cuando varias políticas son aplicables a una solicitud, Secrets Manager utiliza una jerarquía para controlar los permisos:

1. Si una instrucción en cualquier política con un `deny` explícito coincide con la acción de solicitud y el recurso:

   El `deny` explícito anula todo lo demás y bloquea la acción.
2. Si no hay `deny` explícito, sino una declaración con un `allow` explícito coincide con la acción de solicitud y el recurso:

   El `allow` explícito otorga a la acción en la solicitud acceso a los recursos de la instrucción.

   Si la identidad y el secreto están en dos cuentas diferentes, debe haber un `allow` tanto en la política de recursos para el secreto como en la política adjunta a la identidad, de lo contrario AWS deniega la solicitud. Para obtener más información, consulte Acceso entre cuentas.
3. Si no hay ninguna instrucción con un `allow` explícito que coincida con la acción de solicitud y el recurso:

   AWS deniega la solicitud de forma predeterminada, que se denomina una negación implícita.

Ver la política basada en recursos de un secreto

- Haga una de las siguientes acciones:
  - Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/. En la página de detalles del secreto del suyo, en la sección Resource permissions (Permisos de recursos), elija Edit permissions (Editar los permisos).
  - Utilice la AWS CLI para llamar a `get-resource-policy`, o bien AWS SDK para llamar a `GetResourcePolicy`.

Determinar quién tiene acceso a través de políticas basadas en identidades

- Utilice el simulador de políticas de IAM. Consulte Probar las políticas de IAM con el simulador de políticas de IAM.

Permisos para secretos de AWS Secrets Manager para usuarios en una cuenta diferente

Para permitir que los usuarios de una cuenta de obtengan acceso a otra cuenta (acceso entre cuentas), debe permitir el acceso tanto en una política de recursos como en una política de identidad. Esto es diferente de conceder acceso a identidades en la misma cuenta que el secreto.

También debe permitir que la identidad utilice la clave de KMS con la que está cifrado el secreto. Esto se debe a que no puede usar la Clave administrada de AWS (`aws/secretsmanager`) para obtener acceso entre cuentas. En su lugar, debe cifrar su secreto con una clave de KMS que cree y, a continuación, adjuntarle una política de clave. Existe un cargo por la creación de claves de KMS. Para cambiar la clave de cifrado de un secreto, consulte the section called “Modificar un secreto”.

Las siguientes políticas de ejemplo suponen que tiene un secreto y una clave de cifrado en la Account1, y una identidad en la Account2 a la que desea permitir acceder al valor secreto.

Paso 1: adjunte una política de recursos al secreto de Account1

- La siguiente política permite a *`ApplicationRole`* in la *`Account2`* acceer al secreto de la *`Account1`*. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a un secreto”.

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

```
}
```

Paso 2: agregue una instrucción a la política clave de la clave de KMS de Account1

- La siguiente instrucción de política clave permite que *ApplicationRole* in la *Account2* use la clave de KMS en la *Account1* para descifrar el secreto en la *Account1*. Para utilizar esta instrucción, agréguela a la política de claves de la clave de KMS. Para obtener más información, consulte Cambiar una política de claves.

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

Paso 3: adjunte una política de identidad a la identidad de Account2

- La siguiente política permite que *ApplicationRole* en la *Account2* acceda al secreto de la *Account1* y descifre el valor secreto utilizando la clave de cifrado que también está en la *Account1*. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a una identidad”. Puede encontrar el ARN para su secreto en la consola de Secrets Manager en la página de detalles secretos en ARN del secreto. También puede llamar a `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"
    }
  ]
}
```

Permisos del rol de ejecución de la función de rotación de Lambda para AWS Secrets Manager

Secrets Manager utiliza una función de Lambda para rotar secretos. Para que la función de Lambda se ejecute, Lambda asume un rol de ejecución de IAM y proporciona esas credenciales al código de la función de Lambda. Consulte las instrucciones sobre cómo configurar la rotación automática en los siguientes recursos:

- Rotación automática de secretos de bases de datos (consola)
- Rotación automática (consola)
- Rotación automática (AWS CLI)

En los ejemplos siguientes se muestran políticas insertadas para roles de ejecución de la función de rotación de Lambda. Para crear un rol de ejecución y adjuntar una política de permisos, consulte Rol de ejecución de AWS Lambda.

Ejemplos:

- Política para el rol de ejecución de una función de rotación de Lambda
- Instrucción de política para una clave administrada por el cliente
- Instrucción de política para la estrategia de usuarios alternativos

Política para el rol de ejecución de una función de rotación de Lambda

La siguiente política de ejemplo permite a la función de rotación lo siguiente:

- Ejecutar operaciones de Secrets Manager para *SecretARN*.
- Crear una contraseña.

- Establecer la configuración requerida si la base de datos o el servicio se ejecutan en una VPC. Consulte Configuración de una función de Lambda para acceder a los recursos de una VPC.

```
{
    "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"
        }
    ]
}
```

Instrucción de política para una clave administrada por el cliente

Si el secreto está cifrado con una clave KMS distinta de la Clave administrada de AWS `aws/secretsmanager`, tiene que conceder permiso al rol de ejecución de Lambda para utilizar la clave. Puede utilizar el contexto de cifrado SecretARN para limitar el uso de la función de descifrado,

de modo que el rol de la función de rotación solo tenga acceso para descifrar el secreto que es responsable de rotar. En el ejemplo siguiente, se muestra una instrucción que se debe agregar a la política del rol de ejecución para descifrar el secreto con una clave de KMS.

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

Si desea utilizar la función de rotación para varios secretos cifrados con una clave administrada por el cliente, agregue una sentencia como la del siguiente ejemplo para permitir que el rol de ejecución descifre el secreto.

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

Instrucción de política para la estrategia de usuarios alternativos

Para obtener información sobre la estrategia de rotación de usuarios alternativos, consulte the section called “Estrategia de rotación”.

Para un secreto que contenga credenciales de Amazon RDS, si utiliza la estrategia de usuarios alternativos y Amazon RDS administra el secreto del superusuario, entonces también debe permitir que la función de rotación llame a las API de solo lectura de Amazon RDS para que pueda obtener la información de conexión de la base de datos. Recomendamos que adjunte la política administrada por AWS AmazonRDSReadOnlyAccess.

La siguiente política de ejemplo permite a la función:

- Ejecutar operaciones de Secrets Manager para *SecretARN*.
- Recuperar las credenciales del secreto de superusuario. Secrets Manager utiliza las credenciales del secreto de superusuario para actualizar las credenciales en el secreto rotado.
- Crear una contraseña.
- Establecer la configuración requerida si la base de datos o el servicio se ejecutan en una VPC. Para obtener más información, consulte Configuración de una función de Lambda para obtener acceso a los recursos en una VPC.

```
{
    "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"
        }
    ]
}
```

Ejemplos de políticas de permisos para AWS Secrets Manager

Una política de permisos es texto estructurado JSON. Consulte Estructura del documento de política JSON.

Las políticas de permisos que se adjuntan a los recursos e identidades son muy similares. Algunos de los elementos que se incluyen en una política de acceso a los secretos incluyen:

- `Principal`: a quién conceder acceso. Consulte Especificación de una entidad principal en la Guía del usuario de IAM. Cuando adjunta una política a una identidad, no incluye un elemento `Principal` en la política.
- `Action`: lo que pueden hacer. Consulte the section called “Acciones de Secrets Manager”.
- `Resource`: a qué secretos pueden acceder. Consulte the section called “Recursos de Secrets Manager”.

  El carácter comodín (*) tiene un significado diferente dependiendo de a qué adjunte la política:

  - En una política adjunta a un secreto, `*` significa que la política se aplica a este secreto.
  - En una política adjunta a una identidad, `*` significa que la política se aplica a todos los recursos, incluidos los secretos, de la cuenta.

Para adjuntar una política a un secreto, consulte the section called “Adjuntar una política de permisos a un secreto”.

Para adjuntar una política a una identidad, consulte the section called “Adjuntar una política de permisos a una identidad”.

Temas

Ejemplo: permiso para recuperar valores secretos

Para conceder permiso para recuperar valores secretos, puede adjuntar políticas a secretos o identidades. Para obtener ayuda para determinar el tipo de política que se va a utilizar, consulte Políticas basadas en identidad y políticas basadas en recursos. Para obtener información sobre cómo adjuntar una política a una identidad, consulte the section called “Adjuntar una política de permisos a un secreto” y the section called “Adjuntar una política de permisos a una identidad”.

En los siguientes ejemplos, se muestran dos maneras diferentes de conceder acceso a un secreto. El primer ejemplo es una política basada en recursos que puede adjuntar a un secreto. Este ejemplo es útil cuando desea conceder acceso a un secreto único a varios usuarios o roles. El segundo ejemplo es una política basada en identidades que puede adjuntar a un usuario o rol en IAM. Este ejemplo es útil cuando desea conceder acceso a un grupo de IAM. Para conceder permiso para recuperar un grupo de secretos en una llamada a la API por lotes, consulte the section called “Permiso para recuperar un grupo de valores secretos en un lote”.

Example Leer un secreto (adjuntar a un secreto)

Puede conceder acceso a un secreto adjuntando la siguiente política al secreto. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a un secreto”.

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

Example Leer un secreto (adjuntar a una identidad)

Puede conceder acceso a un secreto adjuntando la siguiente política una identidad. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a una identidad”. Si adjunta esta política al rol de *EC2RoleToAccessSecrets*, concede los mismos permisos que la directiva anterior.

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

Example Lea un secreto cifrado mediante una clave administrada por el cliente (adjunte a la identidad)

Si un secreto se cifra con una clave administrada por el cliente, puede conceder acceso para leer el secreto si adjunta la siguiente política a una identidad. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a una identidad”.

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "SecretARN"
    },
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "KMSKeyARN"
    }
  ]
}
```

Permiso para recuperar un grupo de valores secretos en un lote

Example Leer un grupo de secretos en un lote (adjuntar a la identidad)

Puedes otorgar acceso para recuperar un grupo de secretos en una llamada a la API por lotes al adjuntar la siguiente política a una identidad La política restringe a la persona que llama para que solo pueda recuperar los secretos especificados por *SecretARN1*, *SecretARN2* y *SecretARN3*incluso si la llamada por lotes incluye otros secretos. Si la persona que llama también solicita otros secretos en la llamada a la API por lotes, Secrets Manager no los devolverá. Para obtener más información, consulte the section called “Recupere los secretos en un lote”. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a una identidad”.

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

```
      "Resource": [
        "SecretARN1",
        "SecretARN2",
        "SecretARN3"
        ]
    }
  ]
}
```

Ejemplo: comodines

Puede utilizar comodines para incluir un conjunto de valores en un elemento de política.

Example Acceder a todos los secretos de una ruta (adjuntar a la identidad)

La siguiente política concede acceso para recuperar todos los secretos con un nombre que comience con "*TestEnv/*". Para utilizar esta política, visite the section called "Adjuntar una política de permisos a una identidad".

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

Example Acceder a metadatos en todos los secretos (adjuntar a la identidad)

Las siguientes políticas conceden `DescribeSecret` y permisos comenzando con `List`: `ListSecrets` y `ListSecretVersionIds`. Para utilizar esta política, visite the section called "Adjuntar una política de permisos a una identidad".

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "secretsmanager:DescribeSecret",
      "secretsmanager:List*"
    ],
```

```
    "Resource": "*"
  }
}
```

Example Coincidir el nombre secreto (adjuntar a la identidad)

La siguiente política concede permisos de Secrets Manager para un secreto por su nombre. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a una identidad”.

Para que coincida con un nombre secreto, cree el ARN para el secreto juntando la región, el ID de cuenta, el nombre secreto y el comodín (?) para que coincida con caracteres aleatorios individuales. Secrets Manager agrega seis caracteres aleatorios a nombres secretos como parte de su ARN, por lo que puede usar este comodín para hacer coincidir esos caracteres. Si utiliza la sintaxis `"another_secret_name-*"`, Secrets Manager coincide con no solo el secreto previsto con los 6 caracteres aleatorios, sino que también coincide con `"another_secret_name-<anything-here>a1b2c3"`.

Debido a que puede predecir todas las partes del ARN de un secreto, excepto por los 6 caracteres aleatorios, utilizando el carácter comodín `'??????'` le permite conceder permisos de forma segura a un secreto que no existe todavía. Tenga en cuenta, no obstante, que si elimina el secreto y vuelve a crearlo con el mismo nombre, el usuario recibe automáticamente permiso para el nuevo secreto, incluso aunque los seis caracteres han cambiado.

```
{
  "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-??????"
      ]
    }
  ]
}
```

Ejemplo: permiso para crear secretos

Para conceder permisos a un usuario para crear un secreto, recomendamos adjuntar una política de permisos a un grupo de IAM al que pertenezca el usuario. Consulte Grupos de usuarios de IAM.

Example Crear secretos (adjuntar a la identidad)

La siguiente política concede permiso para crear secretos y ver una lista de secretos. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a una identidad”.

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

Ejemplo: permisos y VPC

Si necesita acceder a Secrets Manager desde una VPC, puede asegurarse de que las solicitudes a Secrets Manager provengan de la VPC mediante la inclusión de una condición en las políticas de permisos. Para obtener más información, consulte Condiciones del punto de enlace de la VPC y Punto de conexión VPC.

Asegúrese de que las solicitudes para acceder al secreto desde otros servicios de AWSTambién provienen de la VPC, de lo contrario esta política les denegará el acceso.

Example Requerir que las solicitudes lleguen a través de un punto de enlace de la VPC (adjuntar a secreto)

La siguiente política permite a un usuario realizar operaciones de Secrets Manager solo cuando la solicitud llega a través del punto de enlace de la VPC *vpce-1234a5678b9012c*. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a un secreto”.

```
{
  "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 Requerir que las solicitudes provengan de una VPC (adjuntar al secreto)

La siguiente política permite utilizar comandos para crear y administrar secretos sólo cuando proceden de *vpc-12345678*. Además, la política permite operaciones que utilizan el acceso al valor cifrado del secreto solo cuando las solicitudes proceden de vpc-2b2b2b2b. Podría utilizar una política como esta en caso de que ejecute una aplicación en una VPC, pero utiliza una segunda VPC aislada para funciones de administración. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a un secreto”.

```
{
  "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"
        }
      }
    }
  ]
}
```

Ejemplo: controlar el acceso a los secretos mediante etiquetas

Puede usar etiquetas para controlar el acceso a sus secretos. Usar etiquetas para controlar los permisos es útil en entornos que están creciendo rápidamente y ayuda con situaciones en las que la administración de políticas resulta engorrosa. Una estrategia consiste en adjuntar etiquetas a los secretos y luego conceder permisos a una identidad cuando un secreto tiene una etiqueta específica.

Example Permitir el acceso a los secretos con una etiqueta específica (adjuntar a una identidad)

La siguiente política permite `DescribeSecret` en secretos con una etiqueta con la clave "*ServerName (Nombre de servidor)*" y el valor "*ServerABC*". Para utilizar esta política, visite the section called "Adjuntar una política de permisos a una identidad".

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "*",
```

```
    "Condition": {
      "StringEquals": {
        "secretsmanager:ResourceTag/ServerName": "ServerABC"
      }
    }
  }
}
```

Ejemplo: limitar el acceso a identidades con etiquetas que coincidan con las etiquetas de los secretos

Una estrategia es adjuntar etiquetas tanto a los secretos como a las identidades de IAM. A continuación, creará políticas de permisos para permitir operaciones cuando la etiqueta de la identidad coincida con la etiqueta del secreto. Para ver un tutorial completo, consulte Defina permisos para acceder a los secretos basados en las etiquetas.

Usar etiquetas para controlar los permisos es útil en entornos que están creciendo rápidamente y ayuda con situaciones en las que la administración de políticas resulta engorrosa. Para obtener más información, consulte ¿Qué es ABAC para AWS?

Example Permitir el acceso a roles que tienen las mismas etiquetas que los secretos (adjuntar a un secreto)

La siguiente política concede `GetSecretValue` a la cuenta *123456789012* solo si la etiqueta *AccessProject* tiene el mismo valor para el secreto y el rol. Para utilizar esta política, visite the section called “Adjuntar una política de permisos a un secreto”.

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

```
  }
}
```

Ejemplo: Entidad principal de servicio

Si la política de recursos adjunta al secreto incluye una entidad principal de servicio de AWS, se recomienda utilizar las claves de condición globales aws:SourceArn y aws:SourceAccount. Los valores del ARN y de la cuenta se incluyen en el contexto de la autorización solo cuando Secrets Manager recibe una solicitud procedente de otro servicio de AWS. Esta combinación de condiciones evita un potencial escenario de suplente confuso.

Si un ARN de recurso incluye caracteres que no están permitidos en una política de recursos, no puede utilizar ese ARN de recurso en el valor de la `aws:SourceArn` clave de condición. En cambio, utilice la clave de condición `aws:SourceAccount`. Para obtener más información, consulte los requisitos IAM.

Las entidades principales de servicio no se utilizan normalmente como entidades principales en una política adjunta a un secreto, pero algunos servicios de AWS lo requieren. Para obtener información sobre las políticas de recursos que un servicio requiere que se adjunten a un secreto, consulte la documentación del servicio.

Example Permitir que un servicio acceda a un secreto mediante una entidad principal de servicio (adjuntar a un secreto)

```
{
  "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"
        }
      }

    }
  ]
}
```

Referencia de permisos para AWS Secrets Manager

Para ver los elementos que componen una política de permisos, consulte Estructura del documento de política JSON y Referencia a los elementos de política de IAM JSON.

Para empezar a escribir su propia política de permisos, consulte the section called “Ejemplos de políticas de permisos”.

Acciones de Secrets Manager

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
CancelRotateSecret	Otorga permiso para cancelar una rotación secreta en curso.	Escritura	Secret*	secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:Res	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				ourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
CreateSecret	Otorga permiso para crear un secreto que almacena datos cifrados que se puede consultar y rotar.	Escritura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:Name secretsmanager:Description secretsmanager:KmsKeyId aws:RequestTag/${TagKey} aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:AddReplicaRegions	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:ForceOverwriteReplicaSecret	
DeleteResourcePolicy	Otorga permiso para eliminar la política de recursos adjunta a un secreto.	Administración de permisos	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
DeleteSecret	Otorga permiso para eliminar un secreto.	Escritura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:RecoveryWindowInDays secretsmanager:ForceDeleteWithoutRecovery secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:Sec	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				retPrimaryRegion	
DescribeSecret	Otorga permiso para recuperar los metadatos acerca de un secreto, pero no los datos cifrados.	Lectura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetRandomPassword	Otorga permiso para generar una cadena aleatoria para su uso en la creación de contraseñas.	Lectura			

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
GetResourcePolicy	Otorga permiso para obtener la política de recursos asociada a un secreto.	Lectura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetSecretValue	Otorga permiso para recuperar y descifrar los datos cifrados.	Lectura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:VersionId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
ListSecretVersionIds	Otorga permiso para enumerar las versiones disponibles de un secreto.	Lectura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ListSecrets	Otorga permiso para enumerar los secretos disponibles.	Enumeración			
PutResourcePolicy	Otorga permiso para asociar una política de recursos a un secreto.	Administración de permisos	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:BlockPublicPolicy secretsmanager:SecretPrimaryRegion	
PutSecretValue	Otorga permiso para crear una nueva versión del secreto con nuevos datos cifrados.	Escritura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RemoveRegionsFromReplication	Otorga permiso para eliminar regiones de replicación.	Escritura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
ReplicateSecretToRegions	Otorga permiso para convertir un secreto existente en un secreto de varias regiones y comienza a replicar el secreto en una lista de regiones nuevas.	Escritura	Secret*	secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:AddReplicaRegions secretsmanager:For	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				ceOverwriteReplicaSecret	
RestoreSecret	Otorga permiso para cancelar la eliminación de un secreto.	Escritura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RotateSecret	Otorga permiso para iniciar la rotación de un secreto.	Escritura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:RotationLambdaARN secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:Mod	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				ifyRotationRules secretsmanager:RotateImmediately	
StopReplicationToReplica	Otorga permiso para eliminar el secreto de la reproducción y promueve el secreto a un secreto regional en la región de la réplica.	Escritura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
TagResource	Otorga permiso para agregar etiquetas a un secreto.	Etiquetado	Secret*	secretsmanager:SecretId aws:RequestTag/${TagKey} aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
UntagResource	Otorga permiso para eliminar etiquetas de un secreto.	Etiquetado	Secret*		
				secretsmanager:SecretId aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
UpdateSecret	Otorga permiso para actualizar un secreto con metadatos nuevos o con una nueva versión de los datos cifrados.	Escritura	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:Description secretsmanager:KmsKeyId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
UpdateSecretVersionStage	Otorga permiso para mover una fase de un secreto a otro.	Escritura	Secret*		
				secretsmanager:SecretId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ValidateResourcePolicy	Otorga permiso para validar una política de recursos antes de adjuntar la política.	Administración de permisos	Secret*		

Acciones	Descripción	Nivel de acceso	Tipos de recursos (*necesarios)	Claves de condición	Acciones dependientes
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Recursos de Secrets Manager

Tipos de recurso	ARN	Claves de condición
Secret	`arn:${Partition}:secretsmanager:${Region}:${Account}:secret:${SecretId}`	aws:RequestTag/${TagKey}

Tipos de recurso	ARN	Claves de condición
		aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:resource/AllowRotationLambdaArn

Secrets Manager crea la última parte del ARN del secreto anexando un guion y seis caracteres alfanuméricos aleatorios al final del nombre del secreto. Si elimina un secreto y después crea otro con el mismo nombre, este formato ayuda a garantizar que las personas con permisos para el secreto original no obtengan automáticamente acceso al nuevo secreto porque Secrets Manager genera seis caracteres aleatorios diferentes.

Puede encontrar el ARN para un secreto en la consola del Secrets Manager en la página de detalles de los secretos o llamando a `DescribeSecret`.

Claves de condición

Si incluye las condiciones de la cadena que figuran en la siguiente tabla en su política de permisos, los intermediarios de Secrets Manager deberán indicar el parámetro correspondiente o, de lo contrario, se les denegará el acceso. Para evitar la denegación de acceso a los intermediarios debido a la ausencia de un parámetro, agregue `IfExists` al final del nombre del operador de condición, por ejemplo `StringLikeIfExists`. Para obtener más información, consulte Elementos de la política de JSON de IAM: Operadores de condición.

Claves de condición	Descripción	Tipo
aws:RequestTag/${TagKey}	Filtra el acceso mediante una clave que está presente en la solicitud que el usuario realiza al servicio Secrets Manager.	Cadena
aws:ResourceTag/${TagKey}	Filtra el acceso por las etiquetas asociadas al recurso	Cadena
aws:TagKeys	Filtra el acceso mediante la lista de todos los nombres de clave de etiqueta presentes en la solicitud que el usuario realiza al servicio Secrets Manager.	ArrayOfString
secretsmanager:AddReplicaRegions	Filtra el acceso mediante la lista de las regiones en las que se va a replicar el secreto.	ArrayOfString
secretsmanager:BlockPublicPolicy	Filtra el acceso en función de si la política de recursos bloquea el acceso amplio a la Cuenta de AWS	Bool
secretsmanager:Description	Filtra el acceso mediante el texto de descripción de la solicitud.	Cadena
secretsmanager:ForceDeleteWithoutRecovery	Filtra el acceso en función de si el secreto se debe eliminar de inmediato sin ventana de recuperación.	Bool
secretsmanager:ForceOverwriteReplicaSecret	Filtra el acceso en función de si se debe sobrescribir un secreto con el mismo nombre en la región de destino.	Bool

Claves de condición	Descripción	Tipo
secretsmanager:KmsKeyId	Filtra el acceso mediante el ARN de la clave de KMS de la solicitud.	Cadena
secretsmanager:ModifyRotationRules	Filtra el acceso en función de si las reglas de rotación del secreto tienen que modificarse.	Bool
secretsmanager:Name	Filtra el acceso mediante el nombre fácil de recordar del secreto de la solicitud.	Cadena
secretsmanager:RecoveryWindowInDays	Filtra el acceso mediante el número de días que Secrets Manager espera antes de poder eliminar el secreto.	Numérico
secretsmanager:ResourceTag/tag-key	Filtra el acceso por par de clave y valor de etiqueta.	Cadena
secretsmanager:RotateImmediately	Filtra el acceso en función de si el secreto debe rotarse inmediatamente.	Bool
secretsmanager:RotationLambdaARN	Filtra el acceso mediante el ARN de la función de Lambda de rotación de la solicitud.	ARN
secretsmanager:SecretId	Filtra el acceso mediante el valor de SecretID de la solicitud.	ARN
secretsmanager:SecretPrimaryRegion	Filtra el acceso mediante la región principal en la que se crea el secreto.	Cadena

Claves de condición	Descripción	Tipo
secretsmanager:VersionId	Filtra el acceso mediante el identificador único de la versión del secreto de la solicitud.	Cadena
secretsmanager:VersionStage	Filtra el acceso mediante la lista de las fases de versión de la solicitud.	Cadena
secretsmanager:resource/AllowRotationLambdaArn	Filtra el acceso mediante el ARN de la función de Lambda de rotación asociada al secreto.	ARN

Bloquear el acceso amplio a los secretos con la condición `BlockPublicPolicy`

En las políticas de identidad que permiten la acción `PutResourcePolicy`, le recomendamos que utilice `BlockPublicPolicy: true`. Esta condición significa que los usuarios solo pueden adjuntar una política de recursos a un secreto si la política no permite un acceso amplio.

Secrets Manager utiliza el razonamiento automatizado de Zelkova para analizar las políticas de recursos para un acceso amplio. Para obtener más información acerca de Zelkova, consulte De qué manera AWS utiliza el razonamiento automatizado para ayudarlo a alcanzar la la seguridad a escala en el Blog de seguridad de AWS.

En el siguiente ejemplo se muestra cómo utilizar `BlockPublicPolicy`.

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

```
        }
    }
}
```

Condiciones de dirección IP

Pero tenga cuidado al especificar los operadores de condición de dirección IP o la clave de condición `aws:SourceIp` en la misma declaración de política que permite o deniega el acceso a Secrets Manager. Por ejemplo, si adjunta una política que restrinja las acciones de AWS para realizar solicitudes desde rango de direcciones IP de la red corporativa a un secreto, las solicitudes que realice como un usuario de IAM que invoca la solicitud desde la red corporativa funcionarán de la manera prevista. Sin embargo, si habilita otros servicios para obtener acceso al secreto en su nombre, como, por ejemplo, cuando se habilita la rotación con una función de Lambda, dicha función llama a las operaciones de Secrets Manager desde un espacio de direcciones internas de AWS. Las solicitudes afectadas por la política con el filtro de dirección IP generarán un error.

Además, la clave de condición `aws:sourceIP` es menos efectiva si la solicitud procede de un punto de conexión de VPC de Amazon VPC. Para restringir las solicitudes a un punto de enlace de la VPC específica, utilice the section called “Condiciones del punto de enlace de la VPC”.

Condiciones del punto de enlace de la VPC

Para permitir o denegar el acceso a solicitudes procedentes de una VPC o punto de enlace de la VPC particular, utilice `aws:SourceVpc` para limitar el acceso a las solicitudes procedentes de la VPC especificada o `aws:SourceVpce` para limitar el acceso a las solicitudes procedentes del punto de enlace de la VPC especificado. Consulte the section called “Ejemplo: permisos y VPC”.

- `aws:SourceVpc` limita el acceso a las solicitudes procedentes de la VPC especificada.
- `aws:SourceVpce` limita el acceso a las solicitudes procedentes del punto de conexión de VPC especificado.

Si utiliza estas claves de condición en una declaración de política de recurso que permite o deniega el acceso a los secretos de Secrets Manager, puede denegar el acceso de forma accidental a los servicios que Secrets Manager utiliza para obtener acceso a los secretos en su nombre. Solo algunos servicios de AWS pueden ejecutarse con un punto de enlace dentro de la VPC. Si restringe las solicitudes de un secreto a una VPC o un punto de enlace de la VPC, pueden producirse errores si las llamadas a Secrets Manager se realizan desde un servicio que no esté configurado.

Consulte Punto de conexión VPC.

Cree y administre secretos con AWS Secrets Manager

Un secreto puede ser una contraseña, un conjunto de credenciales, como un nombre de usuario y una contraseña, un token de OAuth u otra información secreta que se almacene de forma cifrada en Secrets Manager.

Temas

Crear un secreto de AWS Secrets Manager base de datos

Después de crear un usuario en Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB, se pueden almacenar sus credenciales en Secrets Manager siguiendo estos pasos. Cuando utilices el AWS CLI o uno de los SDK para almacenar el secreto, debes proporcionarlo en la estructura JSON correcta. Al utilizar la consola para almacenar un secreto de base de datos, Secrets Manager lo crea automáticamente en la estructura JSON correcta.

> **Tip**
>
> Para las credenciales de usuario administrador de Amazon RDS y Amazon Redshift, le recomendamos que utilice secretos gestionados. El secreto administrado se crea a través del servicio de administración y, a continuación, se puede utilizar la rotación administrada.

Cuando se almacenan las credenciales de una base de datos de origen que se replica a otras regiones, el secreto contiene información de conexión para la base de datos de origen. Si luego replica el secreto, las réplicas son copias del secreto de origen y contienen la misma información de conexión. Puede agregar pares clave/valor adicionales al secreto para obtener información de conexión regional.

Para crear un secreto, necesita los permisos otorgados por. SecretsManagerReadWriteAWS políticas gestionadas

Secrets Manager genera una entrada de CloudTrail registro al crear un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para crear un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Elija Almacenar un secreto nuevo.
3. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:

    a. En Secret type (Tipo secreto), elija el tipo de credenciales de base de datos que desea almacenar:

        - Base de datos de Amazon RDS (incluye Aurora)
        - Base de datos de Amazon DocumentDB
        - Almacén de datos de Amazon Redshift

    b. En Credentials (Credenciales), ingrese las credenciales de la base de datos.

    c. En Clave de cifrado, elija la AWS KMS key que Secrets Manager utiliza para cifrar el valor secreto. Para obtener más información, consulte Cifrado y descifrado de secretos.

        - Para la mayoría de los casos, elija aws/secretsmanager para utilizar la Clave administrada de AWS para Secrets Manager. No se aplica ningún cargo por el uso de esta clave.
        - Si necesita acceder al secreto desde otra Cuenta de AWS persona o si quiere usar su propia clave KMS para poder rotarla o aplicarle una política de claves, elija una clave gestionada por el cliente de la lista o seleccione Añadir nueva clave para crear una. Para obtener información sobre los costos por usar una clave administrada por el cliente, consulte Precios.

Debe tener the section called “Permisos para la clave KMS”. Para más información sobre el acceso entre cuentas, consulte the section called “Acceso entre cuentas”.

d. En Database (Base de datos), elija la base de datos.

e. Elija Siguiente.

4. En la página Configure secret (Configurar el secreto), haga lo siguiente:

   a. Ingrese un Nombre de secreto descriptivo y una Descripción. Los nombres de los secretos deben contener de 1 a 512 caracteres Unicode.

   b. (Opcional) En la sección Tags (Etiquetas), agregue etiquetas a su secreto. Para obtener información sobre estrategias de etiquetado, consultethe section called “Etiquetado de secretos de ”. No almacene información confidencial en etiquetas porque no están cifradas.

   c. (Opcional) En Resource permissions (Permisos de recursos), para agregar una política de recursos a su secreto, elija Edit permissions (Editar permisos). Para obtener más información, consulte the section called “Adjuntar una política de permisos a un secreto”.

   d. (Opcional) En Replicar secreto, para replicar tu secreto en otro Región de AWS, selecciona Replicar secreto. Puede replicar el secreto ahora o volver y replicarlo más tarde. Para obtener más información, consulte Replicar un secreto a otras regiones.

   e. Seleccione Siguiente.

5. (Opcional) En la página Configure rotation (Configurar rotación), puede activar la rotación automática. También puede mantener la rotación desactivada por ahora y activarla más tarde. Para obtener más información, consulte Rotar secretos de . Seleccione Siguiente.

6. En la página Review (Revisar), revise los detalles del secreto y, a continuación, elija Store (Almacenar).

   Secrets Manager vuelve a la lista de secretos. Si el nuevo secreto no aparece, elija el botón Refresh (Actualizar).

AWS CLI

Cuando utiliza ingresa comandos en un shell de comandos, existe el riesgo de que se acceda al historial de comandos o de que las utilidades tengan acceso a sus parámetros de comando. Consulte the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

Example Crear un secreto a partir de credenciales de un archivo JSON

En el siguiente ejemplo de `create-secret`, se crea un secreto a partir de las credenciales de un archivo. Para obtener más información, consulte Carga de AWS CLI parámetros desde un archivo en la Guía del AWS CLI usuario.

Para que Secrets Manager pueda rotar el secreto, debe asegurarse de que el JSON coincida con el Estructura JSON de un secreto.

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

Contenido de mycreds.json:

```
{
    "engine": "mysql",
    "username": "saanvis",
    "password": "EXAMPLE-PASSWORD",
    "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
    "dbname": "myDatabase",
    "port": "3306"
}
```

AWS SDK

Para crear un secreto mediante uno de los AWS SDK, usa la `CreateSecret`acción. Para obtener más información, consulte the section called “AWS SDKs”.

Estructura JSON de AWS Secrets Manager secretos

Puede almacenar cualquier texto o binario en los secretos de Secrets Manager. Si desea activar la rotación automática para un secreto de Secrets Manager, el secreto debe tener la estructura JSON correcta. Durante la rotación, Secrets Manager utiliza la información del secreto para conectarse al origen de credenciales y actualizar las credenciales allí. Los nombres de las claves JSON distinguen mayúsculas de minúsculas.

Tenga en cuenta que cuando se utiliza la consola para almacenar un secreto de base de datos Secrets Manager lo crea automáticamente con la estructura JSON correcta.

Puede agregar más pares clave/valor a un secreto, por ejemplo en un secreto de base de datos, para que contenga información de conexión de bases de datos de réplica de otras regiones.

Temas

Estructura del secreto de Amazon RDS Db2

En el caso de las instancias Db2 de Amazon RDS, dado que los usuarios no pueden cambiar sus propias contraseñas, debe proporcionar las credenciales de administrador en un secreto independiente.

```
{
  "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>"
}
```

Estructura del secreto de MariaDB en Amazon RDS

```
{
  "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>
}
```

Para utilizarlosthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura secreta de Amazon RDS y Amazon Aurora MySQL

```
{
  "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>
}
```

Para usar elthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura del secreto de Oracle en Amazon RDS

```
{
  "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>
}
```

Para usar elthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura del secreto de Amazon RDS y Amazon Aurora PostgreSQL

```
{
  "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>
}
```

Para usar elthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura del secreto de Microsoft SQL Server en Amazon RDS

```
{
  "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>
}
```

Para usar elthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura del secreto de Amazon DocumentDB

```
{
  "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>
}
```

Para usar elthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>
}
```

Estructura del secreto de Amazon Redshift

```
{
  "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>
}
```

Para usar elthe section called “Usuarios alternativos”, debes incluir el `masterarn` para el secreto que contiene las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura secreta de Amazon Redshift Serverless

```
{
  "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>
}
```

Para utilizarlathe section called “Usuarios alternativos”, debe incluir el secreto que contiene `masterarn` las credenciales de administrador o superusuario.

```
{
  "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>"
}
```

Estructura ElastiCache secreta de Amazon

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

Para obtener más información, consulta Rotación automática de contraseñas para los usuarios en la Guía del ElastiCache usuario de Amazon.

Creación de un secreto de AWS Secrets Manager

Para almacenar claves de API, tokens de acceso, credenciales que no son para bases de datos y otros secretos en Secrets Manager, siga estos pasos. En el caso de un secreto de Amazon ElastiCache, si se desea activar la rotación, se debe almacenar el secreto con la estructura JSON esperada.

Para crear un secreto, necesita los permisos otorgados por AWS políticas gestionadas SecretsManagerReadWrite.

Secrets Manager genera una entrada de registro de CloudTrail cuando crea un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para crear un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Elija Store a new secret (Almacenar un nuevo secreto).
3. En la página Choose secret type (Elegir tipo de secreto), haga lo siguiente:

   a. En Secret type (Tipo de secreto), elija Other type of secret (Otro tipo de secreto).

   b. En Pares clave-valor, ingrese su secreto en pares Clave/valor o elija la pestaña Texto no cifrado e ingrese el secreto en cualquier formato. Puede almacenar hasta 65536 bytes en el secreto.

   c. En Clave de cifrado, elija la AWS KMS key que Secrets Manager utiliza para cifrar el valor secreto. Para obtener más información, consulte Cifrado y descifrado de secretos.

      - Para la mayoría de los casos, elija aws/secretsmanager para utilizar la Clave administrada de AWS para Secrets Manager. No se aplica ningún cargo por el uso de esta clave.
      - Si necesita acceder al secreto desde otro Cuenta de AWS, o si desea usar su propia clave KMS para poder rotarla o aplicarle una política de clave, elija una clave administrada por el cliente de la lista o elija Agregar nueva clave para crear una. Para obtener información sobre los costos por usar una clave administrada por el cliente, consulte Precios.

        Debe tener the section called “Permisos para la clave KMS”. Para más información sobre el acceso entre cuentas, consulte the section called “Acceso entre cuentas”.

   d. Elija Next (Siguiente).

4. En la página Configure secret (Configurar el secreto), haga lo siguiente:

   a. Ingrese un Nombre de secreto descriptivo y una Descripción. Los nombres de los secretos deben contener de 1 a 512 caracteres Unicode.

   b. (Opcional) En la sección Tags (Etiquetas), agregue etiquetas a su secreto. Para obtener información sobre estrategias de etiquetado, consultethe section called “Etiquetado de secretos de ”. No almacene información confidencial en etiquetas porque no están cifradas.

   c. (Opcional) En Resource permissions (Permisos de recursos), para agregar una política de recursos a su secreto, elija Edit permissions (Editar permisos). Para obtener más información, consulte the section called “Adjuntar una política de permisos a un secreto”.

   d. (Opcional) En Replicate secret (Replicar secreto), para replicar el secreto a otra Región de AWS, elija Replicate secret (Replicar secreto). Puede replicar el secreto ahora o volver y replicarlo más tarde. Para obtener más información, consulte Replicar un secreto a otras regiones.

   e. Elija Next (Siguiente).

5. (Opcional) En la página Configure rotation (Configurar rotación), puede activar la rotación automática. También puede mantener la rotación desactivada por ahora y activarla más tarde. Para obtener más información, consulte Rotar secretos de . Elija Next (Siguiente).

6. En la página Review (Revisar), revise los detalles del secreto y, a continuación, elija Store (Almacenar).

   Secrets Manager vuelve a la lista de secretos. Si el nuevo secreto no aparece, elija el botón Refresh (Actualizar).

AWS CLI

Cuando utiliza ingresa comandos en un shell de comandos, existe el riesgo de que se acceda al historial de comandos o de que las utilidades tengan acceso a sus parámetros de comando. Consulte the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

Example Creación de un secreto

En el siguiente ejemplo de `create-secret` se crea un secreto con dos pares clave-valor.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
```

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

Example Crear un secreto a partir de credenciales de un archivo JSON

En el siguiente ejemplo de `create-secret`, se crea un secreto a partir de las credenciales de un archivo. Para obtener más información, consulte Carga de parámetros de la AWS CLI desde un archivo en la Guía del usuario de la AWS CLI.

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

Contenido de mycreds.json:

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

SDK de AWS

Para crear un secreto mediante uno de los SDK de AWS, utilice la acción `CreateSecret`. Para obtener más información, consulte the section called “AWS SDKs”.

Actualizar el valor de un secreto AWS Secrets Manager

Para actualizar el valor de un secreto, se puede utilizar la consola, la CLI o un SDK. Cuando actualiza el valor del secreto, Secrets Manager crea una nueva versión del secreto con la etiqueta transitoria `AWSCURRENT`. Puede seguir accediendo a la versión anterior, que tiene la etiqueta `AWSPREVIOUS`. También puede añadir sus propias etiquetas. Para obtener más información, consulte Secretos de Secrets Manager.

Para actualizar el valor del secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto.

3. En la página Detalles del secreto, en la pestaña Descripción general, en la sección Valor del secreto, elija Recuperar valor del secreto y luego elija Editar.

AWS CLI

Actualización del valor del secreto (AWS CLI)

- Cuando utiliza ingresa comandos en un shell de comandos, existe el riesgo de que se acceda al historial de comandos o de que las utilidades tengan acceso a sus parámetros de comando. Consulte the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

  En el siguiente ejemplo de `put-secret-value` se crea una nueva versión de un secreto con dos pares clave-valor.

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

  El siguiente `put-secret-value` crea una nueva versión con una etiqueta transitoria personalizada. La nueva versión tendrá las etiquetas `MyLabel` y `AWSCURRENT`.

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

SDK de AWS

Le recomendamos que evite llamar a `PutSecretValue` or `UpdateSecret` a un ritmo sostenido de más de una vez cada 10 minutos. Cuando llama a `PutSecretValue` o `UpdateSecret` para actualizar el valor del secreto, Secrets Manager crea una nueva versión del secreto. Secrets Manager elimina versiones sin etiquetar cuando hay más de 100, pero no elimina versiones creadas hace menos de 24 horas. Si actualiza el valor secreto más de una vez cada 10 minutos, crea más versiones de las que elimina Secrets Manager y alcanzará la cuota para las versiones de secretos.

Para actualizar un secreto, utilice las siguientes acciones: `UpdateSecret` o `PutSecretValue`. Para obtener más información, consulte the section called “AWS SDKs”.

Cambiar la clave de cifrado de un AWS Secrets Manager secreto

Secrets Manager utiliza el cifrado de sobres con AWS KMS claves y claves de datos para proteger cada valor secreto. Para cada secreto, puede elegir qué clave de KMS desea utilizar. Puede usar Clave administrada de AWS aws/secretsmanager o puede usar una clave administrada por el cliente. En la mayoría de los casos, se recomienda utilizar aws/secretsmanager, cuyo uso no tiene costo alguno. Si necesita acceder al secreto desde otra Cuenta de AWS persona o si quiere usar su propia clave de KMS para poder rotarla o aplicarle una política de claves, utilice una. clave administrada por el cliente Debe tener the section called “Permisos para la clave KMS”. Para obtener información sobre los costos por usar una clave administrada por el cliente, consulte Precios.

Puede cambiar la clave de cifrado de un secreto. Por ejemplo, si quieres acceder al secreto desde otra cuenta y el secreto está cifrado actualmente con la clave AWS gestionada`aws/secretsmanager`, puedes cambiar a una clave administrada por el cliente.

> **Tip**
>
> Si quieres rotar la tuya clave administrada por el cliente, te recomendamos que utilices la rotación AWS KMS automática de la clave. Para obtener más información, consulte AWS KMS Teclas giratorias.

Al cambiar la clave de cifrado, Secrets Manager vuelve a cifrar `AWSCURRENT` las `AWSPREVIOUS` versiones con la nueva clave. `AWSPENDING` Para evitar que descubras el secreto, Secrets Manager mantiene todas las versiones existentes cifradas con la clave anterior. Esto significa que puedes descifrar todas `AWSCURRENT` las `AWSPENDING` `AWSPREVIOUS` versiones con la clave anterior o con la nueva clave.

Para que solo se `AWSCURRENT` pueda descifrar con la nueva clave de cifrado, cree una nueva versión del secreto con la nueva clave. Luego, para poder descifrar la versión `AWSCURRENT` secreta, debe tener permiso para usar la nueva clave.

Si desactiva la clave de cifrado anterior, no podrá descifrar ninguna versión secreta excepto `AWSCURRENT`, `AWSPENDING` y `AWSPREVIOUS`. Si tiene otras versiones etiquetadas como secretas para las que desea conservar el acceso, tendrá que volver a crear esas versiones con la nueva clave de cifrado mediante the section called “AWS CLI”.

Cambiar la clave de cifrado de un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto.
3. En la sección Detalles de secreto, elija Acciones y, a continuación, elija Editar clave de cifrado.

AWS CLI

Si cambia la clave de cifrado anterior para un secreto y luego desactiva la clave de cifrado anterior, no podrá descifrar ninguna versión de secreto excepto `AWSCURRENT`, `AWSPENDING` y `AWSPREVIOUS`. Si tiene otras versiones etiquetadas como secretas para las que desea conservar el acceso, tendrá que volver a crear esas versiones con la nueva clave de cifrado mediante the section called “AWS CLI”.

Cambiar la clave de cifrado de un secreto (AWS CLI)

1. En el siguiente ejemplo de `update-secret` se actualiza la clave de KMS utilizada para cifrar el valor de secreto. La clave de KMS debe estar en la misma región que el secreto.

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

2. (Opcional) Si tiene versiones de secretos con etiquetas personalizadas, para volver a cifrarlas con la nueva clave, debe crear nuevamente esas versiones.

   Cuando utiliza ingresa comandos en un shell de comandos, existe el riesgo de que se acceda al historial de comandos o de que las utilidades tengan acceso a sus parámetros de comando. Consulte the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

   a. Obtenga el valor de la versión de secreto.

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

      Anote el valor del secreto.

b. Cree una nueva versión con ese valor.

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

Modificación de un secreto de AWS Secrets Manager

Puede modificar los metadatos de un secreto después de crearlo, según quién haya creado el secreto. En el caso de los secretos creados por otros servicios, es posible que necesite usar el otro servicio para actualizarlo o rotarlo.

Para determinar quién administra un secreto, puede revisar el nombre del secreto. Los secretos gestionados por otros servicios llevan el prefijo ID de ese servicio. O bien, en el AWS CLI, llame a describe-secret y, a continuación, revise el campo `OwningService`. Para obtener más información, consulte Secretos gestionados por otros servicios.

Para los secretos que administra, puede modificar la descripción, la política basada en recursos, la clave de cifrado y las etiquetas. También puede cambiar el valor cifrado del secreto, sin embargo le recomendamos que utilice la rotación para actualizar los valores del secreto que contengan credenciales. La rotación actualiza tanto el secreto en Secrets Manager como las credenciales de la base de datos o servicio. Esto mantiene al secreto sincronizado automáticamente para que cuando los clientes soliciten un valor del secreto, recuperen siempre un conjunto de credenciales en funcionamiento. Para obtener más información, consulte Rotar secretos de .

Secrets Manager genera una entrada de registro de CloudTrail cuando modifica un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para actualizar un secreto que administra (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto.
3. En la página de detalles del secreto, haga una de estas cosas:

   Tenga en cuenta que no debe cambiar el nombre ni el ARN de un secreto.

   - Para actualizar la descripción, en la sección Secrets details (Detalles de secreto), elija Actions (Acciones) y, a continuación, elija Edit description (Editar descripción).

- Para actualizar la clave de cifrado, consulte the section called “Cambiar la clave de cifrado de un secreto”.
- Para actualizar las etiquetas, en la pestaña Etiquetas, elija Editar. Consulte the section called “Etiquetado de secretos de ”.
- Si desea actualizar el valor del secreto, consulte the section called “Actualización del valor del secreto”.
- Para actualizar los permisos del secreto, seleccione Editar permisos en la pestaña Descripción general. Consulte the section called “Adjuntar una política de permisos a un secreto”.

- Para actualizar la rotación del secreto, seleccione Editar rotación en la pestaña Rotar. Consulte Rotar secretos de .
- Para replicar el secreto a otras regiones, consulte Replicar un secreto a otras regiones.
- Si el secreto tiene réplicas, puede cambiar la clave de cifrado de una réplica. En la sección Replicar secreto, seleccione el botón de radio correspondiente a la réplica y, a continuación, en el menú Acciones, elija Editar clave de cifrado. Consulte the section called “Cifrado y descifrado de secretos”.
- Para cambiar un secreto de modo que lo administre otro servicio, se debe volver a crear el secreto en ese servicio. Consulte Secretos gestionados por otros servicios.

AWS CLI

Example Actualizar la descripción de un secreto

En el siguiente ejemplo de `update-secret` se actualiza la descripción de un secreto.

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

SDK de AWS

Le recomendamos que evite llamar a `PutSecretValue` or `UpdateSecret` a un ritmo sostenido de más de una vez cada 10 minutos. Cuando llama a `PutSecretValue` o `UpdateSecret` para actualizar el valor del secreto, Secrets Manager crea una nueva versión del secreto. Secrets Manager elimina versiones sin etiquetar cuando hay más de 100, pero no elimina versiones creadas

hace menos de 24 horas. Si actualiza el valor secreto más de una vez cada 10 minutos, crea más versiones de las que elimina Secrets Manager y alcanzará la cuota para las versiones de secretos.

Para actualizar un secreto, utilice las siguientes acciones: `UpdateSecret` o `ReplicateSecretToRegions`. Para obtener más información, consulte the section called “AWS SDKs”.

Buscar secretos en AWS Secrets Manager

Cuando se buscan secretos sin un filtro, Secrets Manager busca coincidencias de palabras clave en el nombre, la descripción, la clave de etiqueta y el valor de etiqueta del secreto. La búsqueda sin filtros no distingue entre mayúsculas y minúsculas, e ignora los caracteres especiales, como el espacio, /, _, =, y #, y solo utiliza números y letras. Cuando realiza búsquedas sin filtro, Secrets Manager analiza la cadena de búsqueda para convertirla en palabras separadas. Las palabras están separadas por cualquier cambio de mayúsculas a minúsculas, de letra a número o de número/letra a puntuación. Por ejemplo, al ingresar el término de búsqueda `credsDatabase#892` se realiza una búsqueda de `creds`, `Database`, y `892` en nombre, descripción y clave y valor de etiqueta.

Secrets Manager genera una entrada de registro de CloudTrail cuando enumera los secretos. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Puede aplicar los siguientes filtros para la búsqueda:

Name (Nombre)

Busca coincidencias con el principio de los nombres de los secretos; distingue entre mayúsculas y minúsculas. Por ejemplo, Name: **`Data`** devuelve un secreto que se llame `DatabaseSecret`, pero no `databaseSecret`, ni `MyData`.

Description (Descripción)

Busca coincidencias con las palabras de las descripciones de los secretos; no distingue entre mayúsculas y minúsculas. Por ejemplo, Description: **`My Description`** devuelve secretos con las siguientes descripciones:

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

Servicio propietario

Busca coincidencias con el principio del ID del servicio de administración, sin distinguir entre mayúsculas y minúsculas. Por ejemplo, **`my-ser`** busca coincidencias de secretos administrados por servicios con el prefijo `my-serv` y `my-service`. Para obtener más información, consulte Secretos gestionados por otros servicios.

Secretos replicados

Puede filtrar por secretos principales, secretos de réplica o secretos que no se hayan replicado.

Tag key (Clave de etiqueta)

Busca coincidencias con el principio de las claves de etiqueta; distingue entre mayúsculas y minúsculas. Por ejemplo, Tag key: **`Prod`** devuelve secretos con la etiqueta `Production` y `Prod1`, pero no secretos con la etiqueta `prod` o `1 Prod`.

Tag value (Valor de etiqueta)

Busca coincidencias con el principio de los valores de etiqueta; distingue entre mayúsculas y minúsculas. Por ejemplo, Tag value: **`Prod`** devuelve secretos con la etiqueta `Production` y `Prod1`, pero no secretos con el valor de etiqueta `prod` o `1 Prod`.

Secrets Manager es un servicio regional y solo se devuelven secretos de la región seleccionada.

AWS CLI

Example Ver una lista de los secretos de la cuenta

En el siguiente ejemplo de `list-secrets` se obtiene una lista de los secretos de la cuenta.

```
aws secretsmanager list-secrets
```

Example Filtrar la lista de secretos de la cuenta

En el siguiente ejemplo de `list-secrets` se obtiene una lista de los secretos de la cuenta que incluyen Test en su nombre. El filtrado por nombres distingue entre mayúsculas y minúsculas.

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

Example Buscar secretos que son administrados por otros servicios de AWS

En el siguiente ejemplo de `list-secrets`, se obtiene una lista de los secretos gestionados por un servicio. Se debe especificar el servicio por el ID. Para obtener más información, consulte Secretos gestionados por otros servicios.

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

SDK de AWS

Para buscar secretos utilizando uno de los AWS SDK, utilice `ListSecrets`. Para obtener más información, consulte the section called “AWS SDKs”.

Eliminación de un secreto de AWS Secrets Manager

Dada la naturaleza crítica de los secretos, AWS Secrets Manager hace, de forma intencionada, que la eliminación de un secreto sea difícil. Secrets Manager no elimina los secretos inmediatamente. En su lugar, Secrets Manager hace que dejen de estar accesibles de inmediato y se programan para su eliminación tras un periodo de recuperación de un mínimo de siete días. Hasta que finaliza el periodo de recuperación, puede recuperar un secreto que ha eliminado anteriormente. No hay ningún cargo por los secretos que ha marcado para su eliminación.

No se puede eliminar un secreto principal si se ha replicado a otras regiones. Elimine primero las réplicas, y luego elimine el secreto principal. Cuando se elimina una réplica, la eliminación se realiza inmediatamente.

No puede eliminar directamente una versión de un secreto. En su lugar, elimine todas las etiquetas provisionales de la versión usando la AWS CLI o los SDK de AWS. Esto marca la versión como obsoleta y permite que Secrets Manager elimine automáticamente la versión en segundo plano.

Si no sabe si una aplicación sigue usando un secreto, puede crear una alarma de Amazon CloudWatch que le alerte de cualquier intento de acceder a un secreto durante el periodo de recuperación. Para obtener más información, consulte Monitoreo de los secretos de AWS Secrets Manager programados para su eliminación mediante Amazon CloudWatch.

Para eliminar un secreto, debe tener los permisos `secretsmanager:ListSecrets` y `secretsmanager:DeleteSecret`.

Secrets Manager genera una entrada de registro de CloudTrail cuando elimina un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para eliminar un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto que desea eliminar.
3. En la sección Secrets details (Detalles de secreto), elija Actions (Acciones) y, a continuación, elija Delete secret (Editar descripción).
4. En el cuadro de diálogo Disable secret and schedule deletion (Desactivar el secreto y programar la eliminación), en Waiting period (Periodo de espera), ingrese la cantidad de días que debe esperar antes de que la eliminación sea permanente. Secrets Manager adjunta un campo denominado `DeletionDate` y lo define en la fecha y hora actual además de la cantidad de días especificado en la ventana de recuperación.
5. Elija Schedule deletion.

Ver los secretos eliminados

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. En la página Secrets (Secretos) elija Preferences (Preferencias) ().
3. En el cuadro de diálogo de Preferencias, seleccione Ver secretos programados para su eliminación y luego elija Guardar.

Para eliminar un secreto de réplica

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija el secreto principal.
3. En la sección Replicate Secret (Replicar secreto), elija el secreto de réplica.
4. En el menú Actions (Acciones), elija Delete Replica (Eliminar la réplica).

AWS CLI

Example Eliminar un secreto

En el siguiente ejemplo de `delete-secret` se elimina un secreto. Se puede recuperar el secreto con `restore-secret` hasta la fecha y hora del campo de respuesta DeletionDate. Para eliminar un secreto que se replica en otras regiones, primero elimine sus réplicas con `remove-regions-from-replication` y, a continuación, llame a `delete-secret`.

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

Example Eliminar un secreto inmediatamente

En el siguiente ejemplo de `delete-secret` se elimina un secreto inmediatamente sin periodo de recuperación. Este secreto no se puede recuperar.

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

Example Eliminación de una réplica de secreto

En el siguiente ejemplo de `remove-regions-from-replication` se elimina un secreto de réplica de eu-west-3. Para eliminar un secreto principal que se replica en otras regiones, primero elimine las réplicas y, a continuación, llame a `delete-secret`.

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

SDK de AWS

Para eliminar un secreto, utilice el comando `DeleteSecret`. Para eliminar una versión de un secreto, use el comando `UpdateSecretVersionStage`. Para eliminar una réplica, utilice el comando `StopReplicationToReplica`. Para obtener más información, consulte the section called “AWS SDKs”.

Restauración de un secreto de AWS Secrets Manager

Secrets Manager considera que un secreto programado para su eliminación está obsoleto y ya no puede acceder directamente. Una vez transcurrida la ventana de recuperación, Secrets Manager elimina el secreto de manera permanente. Una vez que Secrets Manager elimina el secreto, no puede recuperarlo. Antes del final de la ventana de recuperación, puede recuperar el secreto y hacer que vuelva a estar accesible. Esto elimina el campo `DeletionDate` que cancela la eliminación permanente programada.

Para restaurar un secreto y los metadatos en la consola, debe tener permisos de `secretsmanager:ListSecrets` y `secretsmanager:RestoreSecret`.

Secrets Manager genera una entrada de registro de CloudTrail cuando restaura un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para restaurar un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto que desea modificar.

   Si los secretos eliminados no aparecen en la lista de secretos, elija Preferences (Preferencias) ( ).
   En el cuadro de diálogo de Preferencias, seleccione Ver secretos programados para su eliminación y luego elija Guardar.
3. En la página Secret details (Detalles del secreto), elija Cancel deletion (Cancelar eliminación).
4. En el cuadro de diálogo Cancel secret deletion (Cancelar eliminación del secreto), elija Cancel deletion (Cancelar eliminación).

AWS CLI

Example Restaurar un secreto eliminado previamente

En el siguiente ejemplo de `restore-secret` se restaura un secreto cuya eliminación se había programado previamente.

```
aws secretsmanager restore-secret \
    --secret-id MyTestSecret
```

SDK de AWS

Para restaurar un secreto marcado para eliminación, utilice el comando `RestoreSecret`. Para obtener más información, consulte the section called “AWS SDKs”.

Replicar un secreto de AWS Secrets Manager a otras Regiones de AWS

Puede replicar los secretos a varias Regiones de AWS para respaldar las aplicaciones repartidas por esas regiones con objeto de cumplir los requisitos de acceso y baja latencia regionales. Si lo necesita más adelante, puede promover un secreto de réplica a secreto independiente y configurarlo para que se replique de manera autónoma. Secrets Manager replica los datos y metadatos secretos cifrados, tales como etiquetas y políticas de recursos, a las regiones especificadas.

El ARN de un secreto replicado es el mismo que el secreto principal, excepto para la región, por ejemplo:

- Secreto principal: `arn:aws:secretsmanager:Region1:123456789012:secret:MySecret-a1b2c3`
- Secreto de réplica: `arn:aws:secretsmanager:Region2:123456789012:secret:MySecret-a1b2c3`

Para obtener información sobre precios para secretos de réplica, consulte Precios de AWS Secrets Manager.

Cuando se almacenan las credenciales de una base de datos de origen que se replica a otras regiones, el secreto contiene información de conexión para la base de datos de origen. Si luego replica el secreto, las réplicas son copias del secreto de origen y contienen la misma información de conexión. Puede agregar pares clave/valor adicionales al secreto para obtener información de conexión regional.

Si activa la rotación para el secreto principal, Secrets Manager rota ese secreto en la Región principal, y el nuevo valor del secreto se propaga a todos los secretos de réplica asociados. No es necesario administrar la rotación individualmente para todos los secretos de réplica.

Puede replicar secretos en todas sus regiones de AWS habilitadas. Sin embargo, si utiliza Secrets Manager en regiones de AWS especiales como AWS GovCloud (US) o regiones de China, sólo

puede configurar secretos y las réplicas dentro de estas regiones de AWS. No se puede replicar un secreto de las regiones de AWS habilitadas a una región especializada, ni replicar secretos de una región especializada a una región comercial.

Para poder replicar un secreto a otra región, debe habilitar esa región. Para obtener más información, consulte Administración de las regiones de AWS.

Es posible utilizar un secreto en varias regiones sin replicarlo llamando al punto de conexión Secrets Manager de la región donde se almacena el secreto. Para obtener una lista de puntos de enlace , consulte the section called “Puntos de conexión de Secrets Manager”. Para usar la replicación a fin de mejorar la resiliencia de su carga de trabajo, consulte Arquitectura de recuperación ante desastres (DR) en AWS, Parte I: Estrategias para la recuperación en la nube.

Secrets Manager genera una entrada de CloudTrail registro al replicar un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para replicar un secreto en otras regiones (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto.
3. En la página de detalles del secreto, en la pestaña Replicación, realice una de las siguientes operaciones:
   - Si el secreto no se ha replicado, elija Replicate secret (Replicar secreto).
   - Si el secreto se ha replicado, en la sección Replicate secret (Replicar secreto), elija Add region (Agregar región).
4. En el cuadro de diálogo Add replica regions (Agregar regiones de réplica), haga lo siguiente:
   a. En AWS Region (Región de), elija la región en la que desee replicar el secreto.
   b. (Opcional) En Encryption key (Clave de cifrado), elija una clave KMS con la que cifrar el secreto. La clave debe estar en la región de réplica.
   c. (Opcional) Para agregar otra región, elija Add more regions (Agregar más regiones).
   d. Elija Replicate (Replicar).

   Vuelve a la página de detalles del secreto. En la sección Replicate secret (Replicar secreto), aparece el Replication status (Estado de replicación) de cada región.

AWS CLI

Example Replicar un secreto a otra región

En el siguiente ejemplo de `replicate-secret-to-regions` se replica un secreto en eu-west-3. La réplica se cifra con la clave administrada de AWS aws/secretsmanager.

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

AWS SDK

Para replicar un secreto, utilice el comando `ReplicateSecretToRegions`. Para obtener más información, consulte the section called “AWS SDKs”.

Solución de problemas

A continuación aparecen algunos de los motivos por los que la replicación puede fallar.

Existe un secreto con el mismo nombre en la región seleccionada

Para solucionar este problema, puede sobrescribir el secreto del nombre duplicado en la región de la réplica. Vuelva a intentar la replicación, y luego, en el cuadro de diálogo Reintentar replicación, seleccione Sobrescribir.

No hay permisos disponibles en la clave KMS para completar la replicación

Secrets Manager primero descifra el secreto antes de volver a cifrarlo con la nueva clave de KMS de la región de réplica. Si no tiene permiso `kms:Decrypt` para la clave de cifrado en la región principal, se producirá este error. Para cifrar el secreto replicado con una clave de KMS que no sea `aws/secretsmanager`, necesita `kms:GenerateDataKey` y `kms:Encrypt` para la clave. Consulte the section called “Permisos para la clave KMS”.

No se encuentra la clave KMS o se ha deshabilitado

Si la clave de cifrado de la región principal está deshabilitada o eliminada, Secrets Manager no podrá replicar el secreto. Este error puede producirse incluso si ha cambiado la clave de cifrado, cuando el secreto tiene versiones con etiquetas personalizadas que se cifraron con la clave de cifrado deshabilitada o eliminada. Para obtener información sobre cómo realiza el cifrado Secrets Manager,

consulte the section called “Cifrado y descifrado de secretos”. Para evitar este problema, puede crear nuevamente las versiones de los secretos para que Secrets Manager las cifre con la clave de cifrado actual. Para obtener información, consulte Cómo cambiar la clave de cifrado de un secreto. Luego, vuelva a intentar la replicación.

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

No se ha habilitado la región donde se produce la replicación

Para obtener información sobre cómo habilitar una región, consulte Administración de regiones de AWS en la Guía de referencia de administración de cuentas de AWS.

Promover un secreto de réplica a secreto independiente en AWS Secrets Manager

Un secreto de réplica es un secreto que se replica desde uno principal a otra Región de AWS. Tiene el mismo valor secreto y los mismos metadatos que el principal, pero se puede cifrar con una clave KMS diferente. Un secreto de réplica no se puede actualizar de manera independiente de su secreto principal, con la excepción de su clave de cifrado. Al promover un secreto de réplica, se lo desvincula del secreto principal, y el secreto de réplica se convierte en un secreto independiente. Los cambios en el secreto principal ya no se replicarán al secreto independiente.

Se puede promover un secreto de réplica a secreto independiente como solución de recuperación de desastres si el secreto principal deja de estar disponible. O puede que quiera promover una réplica a secreto independiente, si desea activar la rotación para la réplica.

Si promueve una réplica, asegúrese de actualizar las aplicaciones correspondientes para que utilicen el secreto independiente.

Secrets Manager genera una entrada de registro de CloudTrail cuando promociona un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Para promover un secreto de réplica (consola)

1. Inicie sesión en el Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Navegue hasta la región de réplica.

3. En la página Secrets (Secretos), seleccione el secreto de réplica.
4. En la página de detalles del secreto de réplica, seleccione Promote to standalone secret (Promocionar a secreto independiente).
5. En el cuadro de diálogo Promote replica to standalone secret (Promocionar la réplica a secreto independiente), ingrese la región y, a continuación, seleccione Promocionar la réplica.

AWS CLI

Example Promocionar un secreto de réplica a principal

En el siguiente ejemplo de `stop-replication-to-replica`, se elimina el enlace entre un secreto de réplica y el principal. El secreto de réplica se promociona a secreto principal en la región de réplica. Debe llamar a `stop-replication-to-replica` desde la región de réplica.

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

SDK de AWS

Para promover una réplica a secreto independiente, utilice el comando `StopReplicationToReplica`. Debe llamar a este comando desde la región del secreto de réplica. Para obtener más información, consulte the section called “AWS SDKs”.

Etiquetado de secretos de AWS Secrets Manager

Secrets Manager define una etiqueta como un rótulo que consta de una clave definida por el usuario y un valor opcional. Se pueden utilizar etiquetas para facilitar la administración, búsqueda y filtrado de secretos y otros recursos de una cuenta de AWS. Para etiquetar secretos, utilice un esquema de nomenclatura estándar en todos los recursos. Para obtener más información, consulte el documento técnico Tagging Best Practices (Prácticas recomendadas de etiquetado).

Puede conceder o denegar el acceso a un secreto mediante la comprobación de las etiquetas adjuntas al secreto. Para obtener más información, consulte the section called “Ejemplo: controlar el acceso a los secretos mediante etiquetas”.

Puede buscar secretos por etiquetas en la consola, AWS CLI y los SDK. AWS también proporciona la herramienta Grupos de recursos para crear una consola personalizada que consolida y organiza

los recursos en función de las etiquetas. Para buscar secretos con una etiqueta específica, consulte the section called “Buscar secretos”. Secrets Manager no admite la asignación de costos basada en etiquetas.

Nunca almacene información confidencial de un secreto en una etiqueta.

Para conocer las cuotas de etiquetas y las restricciones de nombres, consulte Cuotas de servicio para el etiquetadoen laAWSGuía de referencia general. Las etiquetas distinguen entre mayúsculas y minúsculas.

Secrets Manager genera una entrada de registro de CloudTrail cuando etiqueta o le quita etiquetas a un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Cambiar las etiquetas del secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el secreto.
3. En la página de detalles del secreto, en la sección Etiquetas, elija Editar etiquetas. Los nombres y valores de clave de etiqueta distinguen entre mayúsculas y minúsculas y las claves de etiquetas deben ser únicas.

AWS CLI

Example Agregar una etiqueta a un secreto

En el siguiente ejemplo de `tag-resource` se muestra cómo asociar una etiqueta con sintaxis abreviada.

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

Example Agregar varias etiquetas a un secreto

En el siguiente ejemplo de `tag-resource` se asocian dos etiquetas de clave-valor a un secreto.

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

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

Example Eliminar etiquetas de un secreto

En el siguiente ejemplo de `untag-resource` se eliminan dos etiquetas de un secreto. Se eliminan tanto la clave como el valor de cada etiqueta.

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

SDK de AWS

Para cambiar las etiquetas de su secreto, utilice `TagResource` o `UntagResource`. Para obtener más información, consulte the section called “AWS SDKs”.

Recuperar secretos de AWS Secrets Manager

Puede recuperar sus secretos:

- En el código
- En otros servicios
- En la AWS CLI
- En la consola de AWS

Secrets Manager genera una entrada de registro de CloudTrail cuando recupera un secreto. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

En el código

En las aplicaciones, puede recuperar sus secretos si llama a `GetSecretValue` o `BatchGetSecretValue`en cualquiera de los AWS SDK. Para ver ejemplos, consulte Get a secret value en la AWS Biblioteca de ejemplos de código de SDK. No obstante, se recomienda que almacene en caché sus valores secretos mediante el almacenamiento en caché del lado del cliente. El almacenado en caché de los secretos mejora la velocidad y reduce los costos.

- Para aplicaciones Java:
  - Si almacena las credenciales de la base de datos en el secreto, utilice los controladores de conexión SQL de Secrets Manager para conectarse a una base de datos mediante esas credenciales.
  - Para otros tipos de secretos, utilice el componente de almacenamiento en caché basado en Java de Secrets Manager o llame al SDK directamente con `GetSecretValue`.
- Para aplicaciones Python, utilice el componente de almacenamiento en caché basado en Python de Secrets Manager o llame directamente al SDK con `get_secret_value` o `batch_get_secret_value`.
- Para aplicaciones .NET, utilice el componente de almacenamiento en caché basado en .NET de Secrets Managero o llame al SDK directamente con `GetSecretValue` o `BatchGetSecretValue`.
- Para aplicaciones Go, utilice el componente de almacenamiento en caché basado en Go de Secrets Manager o llame directamente al SDK con `GetSecretValue` o `BatchGetSecretValue`.

- Para aplicaciones de JavaScript, llame al SDK directamente con `getSecretValue` o `batchGetSecretValue`.
- Para aplicaciones de PHP, llame al SDK directamente con `GetSecretValue` o `BatchGetSecretValue`.
- Para aplicaciones de Ruby, llame al SDK directamente con `get_secret_value` o `batch_get_secret_value`.
- Para obtener información sobre GitHub Actions, consulte the section called “GitHub trabajos”.

En otros sistemas y servicios de AWS

También puede recuperar secretos en lo siguiente:

- En AWS Batch, puede hacer referencia a secretos en una definición de trabajo.
- En AWS CloudFormation, puede crear secretos y hacer referencia a secretos en una pila de CloudFormation.
- En Amazon ECS, puede hacer referencia a secretos en una definición de contenedor.
- En Amazon EKS puede utilizar Uso de secretos AWS Secrets Manager en Amazon Elastic Kubernetes Service (ASCP) para montar secretos como archivos en Amazon EKS.
- Para GitHub, puede usar la acción GitHub de Secrets Manager para agregar secretos como variables de entorno en los trabajos de GitHub.
- En AWS IoT Greengrass, puede hacer referencia a secretos en un grupo de Greengrass.
- En AWS Lambda, puede usar secretos de referencia en una función de Lambda.
- En Parameter Store, puede hacer referencia a secretos en un parámetro.

AWS CLI

Example Recuperar el valor de secreto cifrado de un secreto

El siguiente ejemplo de `get-secret-value` obtiene el valor de secreto actual.

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

Example Recuperar el valor de secreto anterior

El siguiente ejemplo de `get-secret-value` obtiene el valor de secreto anterior.

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

Consola de AWS

Recuperar un secreto (consola)

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la lista de secretos, elija el que desea recuperar.
3. En la sección Secret value (Valor secreto), elija Retrieve secret value (Recuperar valor secreto).

   Secrets Manager muestra la versión actual (`AWSCURRENT`) del secreto. Para ver otras versiones del secreto, como versiones `AWSPREVIOUS` o versiones con etiquetas personalizadas, utilice the section called “AWS CLI”.

Recupera un grupo de secretos en un lote de AWS Secrets Manager

Secrets Manager ofrece la API por lotes `BatchGetSecretValue` para recuperar un grupo de secretos en una llamada a la API. Para elegir qué secretos recuperar, puede especificar una lista de secretos por nombre o ARN, o usar filtros. Si Secrets Manager encuentra errores, por ejemplo, `AccessDeniedException` al intentar recuperar alguno de los secretos, podrá ver los errores `Errors` en la respuesta.

Permisos para recuperar los secretos de un lote

Debe tener `secretsmanager:GetSecretValue` permiso para cada uno de los secretos que desea recuperar. También debe tener el `secretsmanager:BatchGetSecretValue` permiso. Si utiliza filtros, también debe tenerlos `secretsmanager:ListSecrets`. Si desea ver un ejemplo de política de permisos, consulte the section called “Permiso para recuperar un grupo de valores secretos en un lote”.

> **Important**
>
> Si tiene una política de VPCE que deniega el permiso para recuperar un secreto individual del grupo en recuperación, `BatchGetSecretValue` no devolverá ningún valor secreto y mostrará un error.

AWS CLI

Example Recupere el valor secreto de un grupo de secretos enumerados por nombre

El siguiente ejemplo `batch-get-secret-value` obtiene el valor del secreto para tres secretos.

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

Example Recupere el valor secreto de un grupo de secretos seleccionado por el filtro

En el siguiente `batch-get-secret-value` ejemplo, se obtiene el valor secreto de los secretos que tienen una etiqueta denominada «Test».

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

Conexión a una base de datos SQL con credenciales en un secreto de AWS Secrets Manager

En las aplicaciones Java, puede utilizar los controladores de conexión SQL de Secrets Manager para conectarse a bases de datos MySQL, PostgreSQL, Oracle, MSSQLServer, Db2 y Redshift mediante las credenciales almacenadas en Secrets Manager. Cada controlador integra el controlador JDBC base, de modo que puede utilizar las llamadas JDBC para obtener acceso a su base de datos. Sin embargo, en lugar de indicar un nombre de usuario y una contraseña para conectarse, se proporciona el ID de un secreto. El controlador llama a Secrets Manager para recuperar el valor del secreto y, a continuación, utiliza las credenciales y la información de conexión que contiene el secreto para conectarse a la base de datos. El controlador también almacena en caché las credenciales mediante la biblioteca de almacenamiento en caché del lado del cliente de Java, por lo

que no es necesario llamar a Secrets Manager en futuras conexiones. La caché actualiza por defecto los secretos cada hora y también cuando se rota uno de ellos. Para configurar la caché, consulte the section called “SecretCacheConfiguration”.

Puede descargar el código fuente desde. GitHub

Para utilizar los controladores de conexión SQL de Secrets Manager:

- Su aplicación debe tener Java 8 o una versión posterior.
- El secreto debe ser uno de los siguientes:
  - Un secreto de base de datos con la estructura JSON esperada. Para comprobar el formato, en la consola de Secrets Manager, consulte su secreto y, a continuación, seleccione Retrieve secret value (Recuperar valor del secreto). Alternativamente, en AWS CLI la llamada get-secret-value.
  - Un secreto administrado de Amazon RDS. Para este tipo de secreto, debe especificar un punto de conexión y un puerto al establecer la conexión.
  - Un secreto gestionado por Amazon Redshift. Para este tipo de secreto, debe especificar un punto de conexión y un puerto al establecer la conexión.

Si la base de datos se replica en otras regiones, para conectarse a una base de datos de réplica de otra región, especifique el punto de conexión y el puerto regionales al crear la conexión. Puede almacenar información de conexión regional en secreto como pares clave/valor adicionales, en los parámetros del almacén de parámetros de SSM o en la configuración de código.

Para agregar el controlador al proyecto, en el archivo de compilación de Maven `pom.xml`, agregue la siguiente dependencia del controlador. Para obtener más información, consulte Secrets Manager SQL Connection Library en el sitio web del repositorio central de Maven.

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

El controlador utiliza la cadena de proveedores de credenciales predeterminada. Si ejecuta el controlador en Amazon EKS, es posible que recoja las credenciales del nodo en el que se ejecuta en lugar del rol de la cuenta de servicio. Para solucionar este problema, agregue la versión 1 de `com.amazonaws:aws-java-sdk-sts` a su archivo de proyecto de Gradle o Maven como una dependencia.

Para establecer una URL de punto final de AWS PrivateLink DNS y una región en el `secretsmanager.properties` archivo:

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

Para anular la región principal, defina la variable del entorno `AWS_SECRET_JDBC_REGION` o realice el siguiente cambio en el archivo `secretsmanager.properties`:

```
drivers.region = region
```

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

Ejemplos:

- Establecer una conexión a una base de datos
- Establecer una conexión especificando el punto de conexión y el puerto
- Uso de la agrupación de conexiones c3p0 para establecer una conexión
- Uso de la agrupación de conexiones c3p0 para establecer una conexión especificando el punto de conexión y el puerto

Establecer una conexión a una base de datos

En el siguiente ejemplo se muestra cómo establecer una conexión con una base de datos con las credenciales e información de conexión de un secreto. Una vez que tenga la conexión, puede utilizar las llamadas JDBC para obtener acceso a la base de datos. Para obtener más información, consulte JDBC Basics en el sitio web de documentación de Java.

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);
```

Establecer una conexión especificando el punto de conexión y el puerto

En el siguiente ejemplo se muestra cómo establecer una conexión con una base de datos mediante las credenciales de un secreto con el punto de conexión y puerto que se especifique.

Los secretos administrados de Amazon RDS no incluyen el punto de conexión ni el puerto de la base de datos. Para conectarse a una base de datos mediante las credenciales maestras de un secreto administrado de Amazon RDS, hay que especificarlas en el código.

Los secretos que se replican en otras regiones pueden mejorar la latencia de la conexión a la base de datos regional, pero no contienen información de conexión distinta del secreto de origen. Cada réplica es una copia del secreto de origen. Para almacenar información de conexión regional en secreto, agregue más pares clave/valor para la información de puerto y punto de conexión para las regiones.

Una vez que tenga la conexión, puede utilizar las llamadas JDBC para obtener acceso a la base de datos. Para obtener más información, consulte JDBC Basics en el sitio web de documentación de Java.

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);
```

Uso de la agrupación de conexiones c3p0 para establecer una conexión

En el siguiente ejemplo se muestra cómo establecer un grupo de conexiones con un archivo `c3p0.properties` que utiliza el controlador para recuperar las credenciales y la información de conexión del secreto. Para `user` y `jdbcUrl`, ingrese el ID del secreto y configure el grupo de conexiones. A continuación, puede recuperar las conexiones del grupo y utilizarlas como cualquier otra conexión de base de datos. Para obtener más información, consulte JDBC Basics en el sitio web de documentación de Java.

Para obtener más información sobre c3p0, consulte c3p0 en el sitio web Machinery For Change.

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
```

Uso de la agrupación de conexiones c3p0 para establecer una conexión especificando el punto de conexión y el puerto

En el siguiente ejemplo se muestra cómo establecer un grupo de conexiones con un archivo `c3p0.properties` que utiliza el controlador para recuperar las credenciales de un secreto con el punto de conexión y puerto que se especifique. A continuación, puede recuperar las conexiones del grupo y utilizarlas como cualquier otra conexión de base de datos. Para obtener más información, consulte JDBC Basics en el sitio web de documentación de Java.

Los secretos administrados de Amazon RDS no incluyen el punto de conexión ni el puerto de la base de datos. Para conectarse a una base de datos mediante las credenciales maestras de un secreto administrado de Amazon RDS, hay que especificarlas en el código.

Los secretos que se replican en otras regiones pueden mejorar la latencia de la conexión a la base de datos regional, pero no contienen información de conexión distinta del secreto de origen. Cada réplica es una copia del secreto de origen. Para almacenar información de conexión regional en secreto, agregue más pares clave/valor para la información de puerto y punto de conexión para las regiones.

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
```

Recuperar secretos de AWS Secrets Manager en aplicaciones Java

Cuando recupera un secreto, puede utilizar el componente de almacenamiento en caché basado en Java de Secrets Manager para utilizarlo más adelante. Recuperar un secreto almacenado en la memoria caché es más rápido que recuperarlo desde Secrets Manager. Dado que la llamada a las API de Secrets Manager conlleva un costo, el uso de una caché puede reducirlo. Para conocer todas las formas en las que puede recuperar secretos, consulte Recuperar secretos.

La política de la caché consiste en el algoritmo menos usado recientemente (LRU), por lo que, cada vez que la caché tiene que descartar un secreto, lo hace con el de uso menos reciente. De forma predeterminada, la caché actualiza secretos cada hora. Puede configurar la frecuencia con la que se actualiza el secreto en la memoria caché, y también conectarse a la recuperación del secreto para agregar más funcionalidad.

La memoria caché no fuerza la recopilación de elementos no utilizados una vez liberadas las referencias de la memoria caché. La implementación de la memoria caché no incluye la invalidación de la memoria caché. La implementación de la memoria caché se centra en la memoria caché en sí misma y no está reforzada ni centrada en la seguridad. Si necesita más seguridad, como cifrar elementos en la memoria caché, utilice las interfaces y los métodos abstractos que se proporcionan.

Para utilizar el componente, debe disponer de lo siguiente:

- Un entorno de desarrollo Java 8 o una versión posterior. Consulte las descargas de Java SE en el sitio web de Oracle.
- El SDK 1.x de AWS para Java. Puede utilizar ambas versiones del SDK de AWS para Java en sus proyectos. Para obtener más información, consulte Using the SDK for Java 1.x and 2.x side-by-side (Uso del SDK para Java 1.x y 2.x en paralelo).

Para descargar el código fuente, consulte Componente del cliente de almacenamiento en caché basado en Java de Secrets Manager en GitHub.

En el archivo pom.xml de Maven, incluya la siguiente dependencia para agregar el componente a su proyecto. Para obtener más información sobre Maven, consulte Getting Started Guide en el sitio web del proyecto de Apache Maven.

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

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

Referencia

- SecretCache
- SecretCacheConfiguration
- SecretCacheHook

Example Recuperación de un secreto

En el siguiente ejemplo de código se muestra una función de Lambda que recupera una cadena del secreto. Sigue la práctica recomendada que consiste en crear una instancia de la memoria caché fuera del controlador de la función, para que no siga llamando a la API en caso de que se vuelva a invocar la función de Lambda.

```
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

Una caché en memoria para los secretos solicitados a Secrets Manager. Utilice the section called “getSecretString” o the section called “getSecretBinary” para recuperar un secreto de la caché. Puede configurar los ajustes de la memoria caché si transfiere un objeto the section called “SecretCacheConfiguration” en el constructor.

Para obtener más información, incluidos algunos ejemplos, consulte the section called “Aplicaciones Java”.

Constructores

`public SecretCache()`

Constructor predeterminado de un objeto `SecretCache`.

`public SecretCache(AWSSecretsManagerClientBuilder builder)`

Construye una nueva memoria caché con un cliente de Secrets Manager creado a partir del `AWSSecretsManagerClientBuilder` proporcionado. Utilice este constructor para personalizar el cliente de Secrets Manager, como por ejemplo, para utilizar una región o un punto de conexión específicos.

`public SecretCache(AWSSecretsManager client)`

Construye una nueva memoria caché del secreto mediante el `AWSSecretsManagerClient` proporcionado. Utilice este constructor para personalizar el cliente de Secrets Manager, por ejemplo, para utilizar una región o un punto de conexión específicos.

`public SecretCache(SecretCacheConfiguration config)`

Construye una nueva memoria caché del secreto mediante el `the section called "SecretCacheConfiguration"` proporcionado.

Métodos

getSecretString

`public String getSecretString(final String secretId)`

Recupera un secreto de cadena de Secrets Manager. Devuelve `String`.

getSecretBinary

`public ByteBuffer getSecretBinary(final String secretId)`

Recupera un secreto en formato binario desde Secrets Manager. Devuelve `ByteBuffer`.

refreshNow

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

Obliga a la memoria caché a actualizarse. Devuelve `true` si la actualización se completa sin errores, en caso contrario, devuelve `false`.

close

`public void close()`

Cierra la caché.

SecretCacheConfiguration

Opciones de configuración de la caché para un the section called “SecretCache”, como el tamaño máximo de esta y el periodo de vida (TTL) de los secretos almacenados en ella.

Constructor

```
public SecretCacheConfiguration
```

Constructor predeterminado de un objeto `SecretCacheConfiguration`.

Métodos

getClient

```
public AWSSecretsManager getClient()
```

Devuelve el `AWSSecretsManagerClient` desde el cual la memoria caché recupera los secretos.

setClient

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

Establece el `AWSSecretsManagerClient` desde el cual la memoria caché recupera los secretos.

getCacheHook

```
public SecretCacheHook getCacheHook()
```

Devuelve la interfaz the section called “SecretCacheHook” utilizada para conectar las actualizaciones de la caché.

setCacheHook

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

Establece la interfaz the section called “SecretCacheHook” utilizada para conectar las actualizaciones de la caché.

getMaxCacheSize

```
public int getMaxCacheSize()
```

Devuelve el tamaño máximo de la caché. El valor predeterminado es de 1024 secretos.

setMaxCacheSize

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

Establece el tamaño máximo de la caché. El valor predeterminado es de 1024 secretos.

getCacheItemTTL

```
public long getCacheItemTTL()
```

Devuelve el TTL en milisegundos de los elementos almacenados en la caché. Si un secreto almacenado en la caché supera este TTL, la caché recupera una nueva copia del secreto del `AWSSecretsManagerClient`. El valor predeterminado es de 1 hora en milisegundos.

La caché actualiza el secreto de forma sincrónica en el momento en que se solicita el secreto después del TTL. Si se produce un error en la actualización sincrónica, la caché devuelve el secreto obsoleto.

setCacheItemTTL

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

Establece el TTL en milisegundos de los elementos almacenados en la caché. Si un secreto almacenado en la caché supera este TTL, la caché recupera una nueva copia del secreto del `AWSSecretsManagerClient`. El valor predeterminado es de 1 hora en milisegundos.

getVersionStage

```
public String getVersionStage()
```

Devuelve la versión de los secretos que desea almacenar en caché. Para obtener más información, consulte Versiones del secreto. El valor predeterminado es `"AWSCURRENT"`.

setVersionStage

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

Establece la versión de los secretos que desea almacenar en caché. Para obtener más información, consulte Versiones del secreto. El valor predeterminado es `"AWSCURRENT"`.

SecretCacheConfiguration withClient

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

Establece el `AWSSecretsManagerClient` desde el cual se recuperan los secretos. Devuelve el objeto `SecretCacheConfiguration` actualizado con la nueva configuración.

SecretCacheConfiguration withCacheHook

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

Establece la interfaz utilizada para conectarse a la caché en memoria. Devuelve el objeto `SecretCacheConfiguration` actualizado con la nueva configuración.

SecretCacheConfiguration withMaxCacheSize

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

Establece el tamaño máximo de la caché. Devuelve el objeto `SecretCacheConfiguration` actualizado con la nueva configuración.

SecretCacheConfiguration withCacheItemTTL

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

Establece el TTL en milisegundos de los elementos almacenados en la caché. Si un secreto almacenado en la caché supera este TTL, la caché recupera una nueva copia del secreto del `AWSSecretsManagerClient`. El valor predeterminado es de 1 hora en milisegundos. Devuelve el objeto `SecretCacheConfiguration` actualizado con la nueva configuración.

SecretCacheConfiguration withVersionStage

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

Establece la versión de los secretos que desea almacenar en caché. Para obtener más información, consulte Versiones del secreto. Devuelve el objeto `SecretCacheConfiguration` actualizado con la nueva configuración.

SecretCacheHook

Una interfaz para conectarse a una the section called “SecretCache” y realizar acciones sobre los secretos almacenados en ella.

put

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

Prepara el objeto para almacenarlo en la caché.

Devuelve el objeto que se almacenará en la caché.

get

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

Deriva el objeto a partir del objeto almacenado en caché.

Devuelve el objeto que se devolverá de la caché

Recuperar secretos de AWS Secrets Manager en aplicaciones Python

Cuando recupera un secreto, puede utilizar el componente de almacenamiento en caché basado en Python de Secrets Manager para utilizarlo más adelante. Recuperar un secreto almacenado en la memoria caché es más rápido que recuperarlo desde Secrets Manager. Dado que la llamada a las API de Secrets Manager conlleva un costo, el uso de una caché puede reducirlo. Para conocer todas las formas en las que puede recuperar secretos, consulte Recuperar secretos.

La política de la caché consiste en el algoritmo menos usado recientemente (LRU), por lo que, cada vez que la caché tiene que descartar un secreto, lo hace con el de uso menos reciente. De forma predeterminada, la caché actualiza secretos cada hora. Puede configurar la frecuencia con la que se actualiza el secreto en la memoria caché, y también conectarse a la recuperación del secreto para agregar más funcionalidad.

La memoria caché no fuerza la recopilación de elementos no utilizados una vez liberadas las referencias de la memoria caché. La implementación de la memoria caché no incluye la invalidación

de la memoria caché. La implementación de la memoria caché se centra en la memoria caché en sí misma y no está reforzada ni centrada en la seguridad. Si necesita más seguridad, como cifrar elementos en la memoria caché, utilice las interfaces y los métodos abstractos que se proporcionan.

Para utilizar el componente, debe disponer de lo siguiente:

- Python 3.6 o posterior
- botocore 1.12 o superior. Consulte AWS SDK para Python y Botocore.
- setuptools_scm 3.2 o superior. Consulte https://pypi.org/project/setuptools-scm/.

Para descargar el código fuente, consulte Componente del cliente de almacenamiento en caché basado en Python de Secrets Manager en GitHub.

Para instalar el componente, utilice el siguiente comando.

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

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

Referencia

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

Example Recuperación de un secreto

En el siguiente ejemplo se muestra cómo obtener el valor del secreto de un secreto denominado *`mysecret`*.

```
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

Una caché en memoria para los secretos recuperados de Secrets Manager. Utilice the section called “get_secret_string” o the section called “get_secret_binary” para recuperar un secreto de la caché. Puede configurar los ajustes de la memoria caché si transfiere un objeto the section called “SecretCacheConfig” en el constructor.

Para obtener más información, incluidos algunos ejemplos, consulte the section called “Aplicaciones Python”.

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

Estos son los métodos disponibles:

- get_secret_string
- get_secret_binary

get_secret_string

Recupera el valor de la cadena del secreto.

Sintaxis de la solicitud

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

Parámetros

- `secret_id` (cadena): [obligatorio] el nombre o ARN del secreto.
- `version_stage` (cadena): la versión de los secretos que desea recuperar. Para obtener más información, consulte Versiones del secreto. El valor predeterminado es “AWSCURRENT”.

Tipo de retorno

string

get_secret_binary

Recupera el valor binario del secreto.

Sintaxis de la solicitud

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

Parámetros

- `secret_id` (cadena): [obligatorio] el nombre o ARN del secreto.
- `version_stage` (cadena): la versión de los secretos que desea recuperar. Para obtener más información, consulte Versiones del secreto. El valor predeterminado es “AWSCURRENT”.

Tipo de retorno

Cadena codificada en base64

SecretCacheConfig

Opciones de configuración de la caché para un the section called “SecretCache”, como el tamaño máximo de esta y el periodo de vida (TTL) de los secretos almacenados en ella.

Parámetros

`max_cache_size` (int)

El tamaño máximo de la caché. El valor predeterminado es de `1024` secretos.

`exception_retry_delay_base` (int)

La cantidad de segundos que se debe esperar luego de que se haya producido una excepción antes de reintentar la solicitud. El valor predeterminado es `1`.

`exception_retry_growth_factor` (int)pur

El factor de crecimiento que se debe utilizar para calcular el tiempo de espera entre los reintentos de las solicitudes en las que se haya producido un error. El valor predeterminado es `2`.

`exception_retry_delay_max` (int)

La cantidad máxima de tiempo en segundos que se debe esperar entre las solicitudes en las que se haya producido un error. El valor predeterminado es `3600`.

`default_version_stage` (str)

La versión de los secretos que desea almacenar en caché. Para obtener más información, consulte Versiones del secreto. El valor predeterminado es `'AWSCURRENT'`.

`secret_refresh_interval` (int)

La cantidad de segundos que se debe esperar entre la actualización de la información del secreto en la caché. El valor predeterminado es `3600`.

`secret_cache_hook` (SecretCacheHook)

Implementación de la clase abstracta `SecretCacheHook`. El valor predeterminado es `None`.

SecretCacheHook

Una interfaz para conectarse a una the section called “SecretCache” y realizar acciones sobre los secretos almacenados en ella.

Estos son los métodos disponibles:

- put
- get

put

Prepara el objeto para almacenarlo en la caché.

Sintaxis de la solicitud

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

Parámetros

- `obj` (objeto): [obligatorio] el secreto o el objeto que contiene el secreto.

Tipo de retorno

objeto

get

Deriva el objeto a partir del objeto almacenado en caché.

Sintaxis de la solicitud

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

Parámetros

- `obj` (objeto): [obligatorio] el secreto o el objeto que contiene el secreto.

Tipo de retorno

objeto

@InjectSecretString

Este elemento Decorator espera una cadena de ID del secreto y una the section called "SecretCache" como primer y segundo argumento. El elemento Decorator devuelve el valor de la cadena del secreto. El nombre del secreto debe contener una cadena.

```
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

Este elemento Decorator espera una cadena de ID del secreto y una the section called "SecretCache" como primer y segundo argumento. Los argumentos restantes asignan parámetros de la función integrada a las claves JSON del secreto. El secreto debe contener una cadena en la estructura JSON.

Para un secreto que contenga este JSON:

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

En el siguiente ejemplo se muestra cómo extraer los valores JSON de `username` y `password` del secreto.

```
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')
```

Recuperar secretos de AWS Secrets Manager en aplicaciones .NET

Cuando recupera un secreto, puede utilizar el componente de almacenamiento en caché basado en .NET de Secrets Manager para utilizarlo más adelante. Recuperar un secreto almacenado en la

memoria caché es más rápido que recuperarlo desde Secrets Manager. Dado que la llamada a las API de Secrets Manager conlleva un costo, el uso de una caché puede reducirlo. Para conocer todas las formas en las que puede recuperar secretos, consulte Recuperar secretos.

La política de la caché consiste en el algoritmo menos usado recientemente (LRU), por lo que, cada vez que la caché tiene que descartar un secreto, lo hace con el de uso menos reciente. De forma predeterminada, la caché actualiza secretos cada hora. Puede configurar la frecuencia con la que se actualiza el secreto en la memoria caché, y también conectarse a la recuperación del secreto para agregar más funcionalidad.

La memoria caché no fuerza la recopilación de elementos no utilizados una vez liberadas las referencias de la memoria caché. La implementación de la memoria caché no incluye la invalidación de la memoria caché. La implementación de la memoria caché se centra en la memoria caché en sí misma y no está reforzada ni centrada en la seguridad. Si necesita más seguridad, como cifrar elementos en la memoria caché, utilice las interfaces y los métodos abstractos que se proporcionan.

Para utilizar el componente, debe disponer de lo siguiente:

- .NET Framework 4.6.2 o una versión posterior, o .NET Standard 2.0 o una versión posterior. Consulte Download .NET (Descargar .NET) en el sitio web de Microsoft .NET.
- SDK de AWS para .NET. Consulte the section called “AWS SDKs”.

Para descargar el código fuente, consulte Cliente de almacenamiento en caché para .NET en GitHub.

Para utilizar la caché, primero hay que crear una instancia y, a continuación, recuperar el secreto mediante `GetSecretString` o `GetSecretBinary`. En las recuperaciones posteriores, la caché devuelve la copia almacenada del secreto.

Para obtener el paquete de almacenamiento en caché

- Haga una de las siguientes acciones:
  - Ejecute el siguiente comando de la CLI de .NET en el directorio del proyecto.

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

  - Agregue la siguiente referencia de paquete al archivo `.csproj`.

    ```
    <ItemGroup>
    ```

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

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

Referencia

- SecretsManagerCache
- SecretCacheConfiguration
- ISecretCacheHook

Example Recuperación de un secreto

En el siguiente ejemplo de código se muestra un método capaz de recuperar un secreto denominado *MySecret*.

```
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 Configurar la duración de la actualización de la memoria caché del tiempo de vida (TTL)

En el siguiente ejemplo de código se muestra un método capaz de recuperar un secreto denominado *MySecret* y establecer la duración de la actualización de la memoria caché de TTL en 24 horas.

```
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

Una caché en memoria para los secretos solicitados a Secrets Manager. Utilice the section called “GetSecretString” o the section called “GetSecretBinary” para recuperar un secreto de la caché. Puede configurar los ajustes de la memoria caché si transfiere un objeto the section called “SecretCacheConfiguration” en el constructor.

Para obtener más información, incluidos algunos ejemplos, consulte the section called "Aplicaciones .NET".

Constructores

`public SecretsManagerCache()`

Constructor predeterminado de un objeto `SecretsManagerCache`.

`public SecretsManagerCache(IAmazonSecretsManager secretsManager)`

Construye una nueva memoria caché con un cliente de Secrets Manager creado a partir del AmazonSecretsManagerClient proporcionado. Utilice este constructor para personalizar el cliente de Secrets Manager, por ejemplo, para utilizar una región o un punto de conexión específicos.

Parámetros

secretsManager

El AmazonSecretsManagerClient desde el cual se recuperan los secretos.

`public SecretsManagerCache(SecretCacheConfiguration config)`

Construye una nueva caché del secreto mediante el the section called "SecretCacheConfiguration" proporcionado. Utilice este constructor para configurar la memoria caché, por ejemplo, la cantidad de secretos que se almacenarán en la caché y la frecuencia con la que se actualizará.

Parámetros

config

Una the section called "SecretCacheConfiguration" que contiene información de configuración de la caché.

`public SecretsManagerCache(IAmazonSecretsManager secretsManager, SecretCacheConfiguration config)`

Construye una nueva memoria caché con un cliente de Secrets Manager creado a partir del AmazonSecretsManagerClient y una the section called "SecretCacheConfiguration" proporcionados. Utilice este constructor para personalizar el cliente de Secrets Manager, por ejemplo, para utilizar una región o un punto de conexión específicos, así como para configurar la

caché, por ejemplo, la cantidad de secretos que se almacenarán en la caché y la frecuencia con la que se actualizará.

Parámetros

secretsManager

El AmazonSecretsManagerClient desde el cual se recuperan los secretos.

config

Una the section called “SecretCacheConfiguration” que contiene información de configuración de la caché.

Métodos

GetSecretString

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

Recupera un secreto de cadena de Secrets Manager.

Parámetros

secretId

El ARN o nombre del secreto que hay que recuperar.

GetSecretBinary

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

Recupera un secreto en formato binario desde Secrets Manager.

Parámetros

secretId

El ARN o nombre del secreto que hay que recuperar.

RefreshNowAsync

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

Solicita el valor del secreto a Secrets Manager y actualiza la caché con los cambios que se hayan producido. Si no hay ninguna entrada en la caché, creará una nueva. Devuelve `true` si la actualización se realiza correctamente.

Parámetros

secretId

El ARN o nombre del secreto que hay que recuperar.

GetCachedSecret

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

Devuelve la entrada de la caché para el secreto especificado si existe en la memoria. En caso contrario, recupera el secreto desde Secrets Manager y crea una nueva entrada en la caché.

Parámetros

secretId

El ARN o nombre del secreto que hay que recuperar.

SecretCacheConfiguration

Opciones de configuración de la caché para un the section called “SecretsManagerCache”, como el tamaño máximo de esta y el periodo de vida (TTL) de los secretos almacenados en ella.

Propiedades

CacheItemTTL

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

El TTL de un elemento de la caché en milisegundos. El valor predeterminado es de `3600000` ms o 1 hora. El máximo es `4294967295` ms, que son aproximadamente 49,7 días.

MaxCacheSize

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

El tamaño máximo de la caché. El valor predeterminado es de 1024 secretos. El máximo es 65 535.

VersionStage

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

La versión de los secretos que desea almacenar en caché. Para obtener más información, consulte Versiones del secreto. El valor predeterminado es `"AWSCURRENT"`.

Cliente

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

El AmazonSecretsManagerClient desde el cual se recuperan los secretos. Si es `null`, la caché crea instancias de un nuevo cliente. El valor predeterminado es `null`.

CacheHook

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

Una the section called “ISecretCacheHook”.

ISecretCacheHook

Una interfaz para conectarse a una the section called “SecretsManagerCache” y realizar acciones sobre los secretos almacenados en ella.

Métodos

Put

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

Prepara el objeto para almacenarlo en la caché.

Devuelve el objeto que se almacenará en la caché.

Get

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

Deriva el objeto a partir del objeto almacenado en caché.

Devuelve el objeto que se devolverá de la caché

Recuperar secretos de AWS Secrets Manager en aplicaciones Go

Cuando recupera un secreto, puede utilizar el componente de almacenamiento en caché basado en Go de Secrets Manager para utilizarlo más adelante. Recuperar un secreto almacenado en la memoria caché es más rápido que recuperarlo desde Secrets Manager. Dado que la llamada a las API de Secrets Manager conlleva un costo, el uso de una caché puede reducirlo. Para conocer todas las formas en las que puede recuperar secretos, consulte Recuperar secretos.

La política de la caché consiste en el algoritmo menos usado recientemente (LRU), por lo que, cada vez que la caché tiene que descartar un secreto, lo hace con el de uso menos reciente. De forma predeterminada, la caché actualiza secretos cada hora. Puede configurar la frecuencia con la que se actualiza el secreto en la memoria caché, y también conectarse a la recuperación del secreto para agregar más funcionalidad.

La memoria caché no fuerza la recopilación de elementos no utilizados una vez liberadas las referencias de la memoria caché. La implementación de la memoria caché no incluye la invalidación de la memoria caché. La implementación de la memoria caché se centra en la memoria caché en sí misma y no está reforzada ni centrada en la seguridad. Si necesita más seguridad, como cifrar elementos en la memoria caché, utilice las interfaces y los métodos abstractos que se proporcionan.

Para utilizar el componente, debe disponer de lo siguiente:

- AWS SDK para Go Consulte the section called “AWS SDKs”.

Para descargar el código fuente, consulte Secrets Manager Go caching client en GitHub.

Para configurar un entorno de desarrollo Go, consulte Golang Getting Started en el sitio web del lenguaje de programación Go.

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

Referencia

- type Cache

- type CacheConfig
- type CacheHook

Example Recuperación de un secreto

En el siguiente ejemplo de código se muestra una función de Lambda que recupera un secreto.

```
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)
}
```

type Cache

Una caché en memoria para los secretos solicitados a Secrets Manager. Se utiliza the section called "GetSecretString" o the section called "GetSecretBinary" para recuperar un secreto de la caché.

En el siguiente ejemplo se muestra cómo configurar los ajustes de la caché.

```
// 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 },
)
```

Para obtener más información, incluidos ejemplos, consulte the section called “Aplicaciones Go”.

Métodos

New

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

New crea una caché del secreto mediante una serie de opciones funcionales; en caso contrario, utiliza los valores predeterminados. Inicializa un cliente de SecretsManager desde una nueva sesión. Inicializa CacheConfig a los valores predeterminados. Inicializa la caché LRU con un tamaño máximo predeterminado.

GetSecretString

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

GetSecretString obtiene el valor de la cadena del secreto de la memoria caché para un determinado ID del secreto. Devuelve la cadena del secreto y un error si la operación no pudo llevarse a cabo.

GetSecretStringWithStage

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

GetSecretStringWithStage obtiene el valor de la cadena del secreto de la memoria caché para un ID del secreto y una fase de versión determinados. Devuelve la cadena del secreto y un error si la operación no pudo llevarse a cabo.

GetSecretBinary

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

GetSecretBinary obtiene el valor binario del secreto de la caché para un determinado ID del secreto. Devuelve el valor binario del secreto y un error si la operación no pudo llevarse a cabo.

GetSecretBinaryWithStage

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

GetSecretBinaryWithStage obtiene el valor binario del secreto de la memoria caché para un ID del secreto y una fase de versión determinados. Devuelve el valor binario del secreto y un error si la operación no pudo llevarse a cabo.

type CacheConfig

Opciones de configuración de la caché, como el tamaño máximo de esta, la fase de versión predeterminada y el período de vida (TTL) de los secretos almacenados en ella.

```
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
}
```

type CacheHook

Una interfaz para conectarse a una caché y realizar acciones sobre el secreto almacenado en ella.

Métodos

Put

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

Prepara el objeto para almacenarlo en la caché.

Get

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

Deriva el objeto a partir del objeto almacenado en caché.

Uso de secretos de AWS Secrets Manager en AWS Batch

AWS Batch le ayuda a ejecutar cargas de trabajo de computación por lotes en Nube de AWS. Con AWS Batch, puede introducir información confidencial en sus trabajos almacenándola en secretos de AWS Secrets Manager y, a continuación, haciendo referencia a ellos en la definición del trabajo. Para obtener más información, consulte Especificación de información confidencial mediante Secrets Manager.

Recuperar un secreto de AWS Secrets Manager en un recurso de AWS CloudFormation

Con AWS CloudFormation, puede recuperar un secreto para utilizarlo en otro recurso de AWS CloudFormation. Un escenario común consiste en crear primero un secreto con una contraseña generada por Secrets Manager y, a continuación, recuperar el nombre de usuario y la contraseña del secreto y utilizarlos como credenciales para una base de datos nueva. Para obtener más información sobre cómo crear secretos con AWS CloudFormation, consulte AWS CloudFormation.

Para recuperar un secreto en una plantilla de AWS CloudFormation, utilice una referencia dinámica. Al crear la pila, la referencia dinámica extrae el valor secreto del recurso AWS CloudFormation, por lo que no tiene que codificar la información secreta. En su lugar, se hace referencia al secreto por su nombre o ARN. Se puede utilizar una referencia dinámica para un secreto en cualquier propiedad de un recurso. No se puede utilizar una referencia dinámica para un secreto en metadatos de un recurso tales como `AWS::CloudFormation::Init`, ya que eso provocaría que el valor de secreto fuera visible en la consola.

Una referencia dinámica de un secreto tiene el siguiente patrón:

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

id-secreto

El nombre o el ARN del secreto. Para obtener acceso a un secreto en su cuenta de AWS, puede utilizar el nombre del secreto. Para acceder a un secreto en una cuenta de AWS diferente, utilice el ARN del secreto.

clave-json (Opcional)

El nombre de la clave del par clave-valor cuyo valor desea recuperar. Si no se especifica una `json-key`, AWS CloudFormation recupera todo el texto secreto. Este segmento no puede incluir el signo de dos puntos (`:`).

fase-versión (Opcional)

La version del secreto que se debe utilizar. Secrets Manager utiliza etiquetas provisionales para realizar un seguimiento de las diferentes versiones durante el proceso de rotación. Si usa `version-stage`, no especifique `version-id`. Si no especifica `version-stage` ni `version-id`, la versión predeterminada es la `AWSCURRENT`. Este segmento no puede incluir el signo de dos puntos (`:`).

id-versión (Opcional)

El identificador único de la versión del secreto a utilizar. Si especifica `version-id`, no especifique `version-stage`. Si no especifica `version-stage` ni `version-id`, la versión predeterminada es la `AWSCURRENT`. Este segmento no puede incluir el signo de dos puntos (`:`).

Para obtener más información, consulte Uso de referencias dinámicas para especificar secretos en Secrets Manager.

> **Note**
>
> No cree una referencia dinámica utilizando una barra invertida (`\`) como valor final. AWS CloudFormation no puede resolver esas referencias, lo que provoca un error en los recursos.

Uso de secretos de AWS Secrets Manager en Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) es un servicio de orquestación de contenedores completamente administrado que facilita la implementación, la administración y el escalado de

aplicaciones en contenedores. Puede inyectar datos confidenciales en contenedores haciendo referencia a secretos de Secrets Manager. Para obtener más información, consulte las siguientes páginas de la Guía para desarrolladores de Amazon Elastic Container Service:

- Tutorial: Especificación de datos confidenciales mediante secretos de Secrets Manager
- Recuperación de secretos mediante programación a través de la aplicación
- Recuperación de secretos a través de variables de entorno
- Recuperación de secretos para la configuración de registro

AWS Secrets Manager Secretos de uso en Amazon Elastic Kubernetes Service

Para mostrar los secretos de Secrets Manager como archivos montados en los pods de Amazon EKS, puede utilizar el proveedor de AWS secretos y configuración (ASCP) para el controlador CSI de Kubernetes Secrets Store. El ASCP funciona con Amazon Elastic Kubernetes Service (Amazon EKS) 1.17+ y ejecuta un grupo de nodos de Amazon EC2. AWS Fargate no se admiten grupos de nodos. Con el ASCP, puede almacenar y administrar sus secretos en Secrets Manager y recuperarlos a través de sus cargas de trabajo que se ejecutan en Amazon EKS. Si su secreto contiene varios pares clave-valor en formato JSON, puede elegir cuáles montar en Amazon EKS. El ASCP utiliza Sintaxis JMESPath para consultar los pares clave-valor en su secreto. El ASCP también funciona con parámetros del almacén de parámetros.

Debe utilizar roles y políticas de IAM para conceder acceso a los secretos a pods específicos de Amazon EKS de un clúster.

Para describir qué archivos se deben crear en el pod de Amazon EKS y qué secretos hay que incluir en ellos, se debe crear un archivo YAML the section called “`SecretProviderClass`”. La `SecretProviderClass` debe estar en el mismo espacio de nombres que el pod de Amazon EKS al que hace referencia.

Si utiliza un clúster privado de Amazon EKS, asegúrese de que la VPC en la que se encuentre el clúster tenga un punto de conexión de Secrets Manager. El controlador de CSI del almacén de secretos utiliza el punto de conexión para realizar llamadas a Secrets Manager. Para obtener información sobre cómo crear un punto de conexión en una VPC, consulte Punto de conexión VPC.

Si utiliza la rotación automática de Secrets Manager para sus secretos, también puede utilizar la característica de conciliador de rotación del controlador de CSI del almacén de secretos a fin de

asegurarse de que está recuperando el último secreto de Secrets Manager. Para obtener más información, consulte Rotación automática de los contenidos montados y los secretos de Kubernetes sincronizados.

Para obtener un tutorial acerca de cómo utilizar el ASCP, consulte the section called “Tutorial”.

Instale el ASCP

El ASCP está disponible GitHub en el repositorio secrets-store-csi-provider-aws. El repositorio también contiene archivos YAML de ejemplo para crear y montar un secreto.

Para instalar el ASCP

- Para instalar el controlador CSI de Secrets Store y ASCP mediante Helm, utilice los siguientes comandos. Para asegurarse de que el repositorio apunte al gráfico más reciente, utilice `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
  ```

  De manera alternativa, para realizar la instalación mediante el archivo YAML en el directorio de implementación, utilice los siguientes comandos.

  ```
  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
  ```

Paso 1: configurar el control de acceso

Para conceder al pod de Amazon EKS acceso a secretos en Secrets Manager, primero se debe crear una política de permisos que conceda permisos `secretsmanager:GetSecretValue` y `secretsmanager:DescribeSecret` para los secretos a los que el pod necesite acceder. Para ver ejemplos de políticas, consulte Ejemplos de políticas de permisos.

A continuación, debe crear un rol de IAM para la cuenta de servicio y adjuntar la política. Para obtener más información, consulte Roles de IAM para cuentas de servicio.

El ASCP recupera la identidad del pod y la cambia por el rol de IAM. ASCP asume el rol de IAM del pod, lo que le da acceso a los secretos autorizados por usted. Otros contenedores no pueden acceder a los secretos a menos que también los asocie con el rol de IAM.

Si utiliza un clúster privado de Amazon EKS, asegúrese de que la VPC en la que se encuentra el clúster tenga un AWS STS punto de conexión. Para obtener información sobre la creación de un punto final, consulte los puntos finales de interfaz de VPC en la Guía delAWS Identity and Access Management usuario.

Paso 2: identificar qué secretos hay que montar

Para determinar qué secretos debe montar el ASCP en Amazon EKS como archivos del sistema de archivos, se debe crear un archivo YAML `SecretProviderClass`. El archivo YAML `SecretProviderClass` contiene una lista de los secretos que hay que montar y el nombre de archivo con el que montarlos. La `SecretProviderClass` debe estar en el mismo espacio de nombres que el pod de Amazon EKS al que hace referencia.

En los siguientes ejemplos se muestra cómo utilizar `SecretProviderClass` para describir los secretos que se desee montar y qué nombre se debe dar a los archivos montados en el pod de Amazon EKS. Para obtener más información, consulte the section called “`SecretProviderClass`”.

Ejemplos:

- Ejemplo: Montar secretos por nombre o ARN
- Ejemplo: Montar pares clave/valor de un secreto
- Ejemplo: Definir una región de conmutación por error para un secreto de varias regiones
- Ejemplo: Seleccionar un secreto de conmutación por error para montarlo

Ejemplo: Montar secretos por nombre o ARN

En el siguiente ejemplo se muestra un `SecretProviderClass` que monta tres archivos en Amazon EKS:

1. Un secreto especificado por ARN completo.
2. Un secreto especificado por su nombre.
3. Una versión específica de un secreto.

```
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"
```

Ejemplo: Montar pares clave/valor de un secreto

En el siguiente ejemplo se muestra un `SecretProviderClass` que monta tres archivos en Amazon EKS:

1. Un secreto especificado por ARN completo.
2. El par clave-valor `username` del mismo secreto.
3. El par clave-valor `password` del mismo secreto.

```
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
```

Ejemplo: Definir una región de conmutación por error para un secreto de varias regiones

Para proporcionar disponibilidad durante interrupciones de conectividad o para configuraciones de recuperación de desastres, el ASCP dispone de una función de conmutación por error automática para recuperar secretos desde una región secundaria.

En el siguiente ejemplo se muestra un `SecretProviderClass` que recupera un secreto que se replica en varias regiones. En este ejemplo, el ASCP intenta recuperar el secreto tanto desde `us-east-1` como desde `us-east-2`. Si alguna de estas regiones devuelve un error 4xx, por ejemplo por un problema de autenticación, el ASCP no monta ninguno de los secretos. Si el secreto se recupera correctamente desde `us-east-1`, el ASCP monta ese valor de secreto. Si el secreto no se recupera correctamente desde `us-east-1`, pero sí desde `us-east-2`, el ASCP monta ese valor de secreto.

```
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"
```

Ejemplo: Seleccionar un secreto de conmutación por error para montarlo

En el siguiente ejemplo se muestra un `SecretProviderClass` que especifica qué secreto se debe montar en caso de conmutación por error. El secreto de conmutación por error no es una réplica. En este ejemplo, el ASCP intenta recuperar los dos secretos especificados por `objectName`. Si alguno devuelve un error 4xx, por ejemplo por un problema de autenticación, el ASCP no monta ninguno de los secretos. Si el secreto se recupera correctamente desde `us-east-1`, el ASCP monta ese valor de secreto. Si el secreto no se recupera correctamente desde `us-east-1`, pero sí desde `us-east-2`, el ASCP monta ese valor de secreto. El archivo montado en Amazon EKS recibe el nombre `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"
```

Solución de problemas

Puede ver la mayoría de los errores describiendo la implementación del pod.

Ver los mensajes de error del contenedor

1. Obtenga una lista de nombres de pods con el siguiente comando. Si no está utilizando el espacio de nombres predeterminado, use `-n <NAMESPACE>`.

   ```
   kubectl get pods
   ```

2. Para describir el pod, en el siguiente comando, en *<PODID>* use el ID de pod de los pods que encontró en el paso anterior. Si no está utilizando el espacio de nombres predeterminado, use -n <NAMESPACE>.

```
kubectl describe pod/<PODID>
```

Ver los errores del ASCP

- Para obtener más información en los registros del proveedor, utilice el siguiente comando para <PODID>utilizar el ID del pod csi-secrets-store-provider-aws.

```
kubectl -n kube-system get pods
kubectl -n kube-system logs pod/<PODID>
```

Tutorial: Crea y monta un AWS Secrets Manager secreto en un pod de Amazon EKS

En este tutorial, creará un secreto de ejemplo en Secrets Manager y, a continuación, lo montará en un pod de Amazon EKS y lo implementará.

Antes de comenzar, instale el ASCP: the section called “Instale el ASCP”.

Crear y montar un secreto

1. Configura el nombre Región de AWS y el nombre de tu clúster como variables de shell para poder utilizarlos en los comandos de bash. Para<REGION>, introduzca el Región de AWS lugar en el que se ejecuta su clúster de Amazon EKS. En *<CLUSTERNAME>*, ingrese el nombre del clúster.

```
REGION=<REGION>
CLUSTERNAME=<CLUSTERNAME>
```

2. Cree un secreto de prueba. Para obtener más información, consulte Cree y administre secretos.

```
aws --region "$REGION" secretsmanager  create-secret --name MySecret --secret-
string '{"username":"lijuan", "password":"hunter2"}'
```

3. Cree una política de recursos para el pod que limite su acceso al secreto que creó en el paso anterior. En *<SECRETARN>*, utilice el ARN del secreto. Guarde el ARN de la política en una variable de shell.

```
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. Cree un proveedor OIDC de IAM para el clúster si todavía no tiene uno. Para obtener más información, consulte Crear un proveedor OIDC de IAM para su clúster.

```
eksctl utils associate-iam-oidc-provider --region="$REGION" --
cluster="$CLUSTERNAME" --approve # Only run this once
```

5. Cree la cuenta de servicio que utiliza el pod y asocie la política de recursos que creó en el paso 3 con esa cuenta de servicio. Para este tutorial, utilice el nombre de la cuenta de servicio nginx-deployment-sa. Para obtener más información, consulte Crear un rol de IAM para una cuenta de servicio.

```
eksctl create iamserviceaccount --name nginx-deployment-sa --region="$REGION" --
cluster "$CLUSTERNAME" --attach-policy-arn "$POLICY_ARN" --approve --override-
existing-serviceaccounts
```

6. Cree la `SecretProviderClass` para especificar qué secreto montar en el pod. El siguiente comando se utiliza `ExampleSecretProviderClass.yaml` en el directorio de ejemplos de GitHub repositorios de ASCP para montar el secreto que creaste en el paso 2. Para obtener información acerca de la creación de su propia `SecretProviderClass`, consulte the section called “`SecretProviderClass`”.

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

7. Implemente el pod El siguiente comando se utiliza `ExampleDeployment.yaml` en el directorio de ejemplos de GitHub repositorios de ASCP para montar el secreto en `/mnt/secrets-store` el pod.

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

8. Para verificar que el secreto se ha montado correctamente, utilice el siguiente comando y confirme que el valor secreto aparece.

```
kubectl exec -it $(kubectl get pods | awk '/nginx-deployment/{print $1}' | head -1)
 cat /mnt/secrets-store/MySecret; echo
```

   El valor secreto aparece.

```
{"username":"lijuan", "password":"hunter2"}
```

SecretProviderClass

Se debe utilizar YAML para describir qué secretos hay que montar en Amazon EKS mediante el ASCP. Para ver ejemplos, consulte Identificar qué secretos hay que montar.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: <NAME>
spec:
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    objects:
```

El campo `parameters` contiene los detalles de la solicitud de montaje.

región

(Opcional) El Región de AWS del secreto. Si no utiliza este campo, el ASCP busca la región en la anotación en el nodo. Esta búsqueda agrega una sobrecarga a las solicitudes de montaje, por lo

que recomendamos que proporcione la Región para los clústeres que utilizan una gran cantidad de pods.

Si también se especifica `failoverRegion`, el ASCP intenta recuperar el secreto desde ambas regiones. Si alguna de estas regiones devuelve un error 4xx, por ejemplo por un problema de autenticación, el ASCP no monta ninguno de los secretos. Si el secreto se recupera correctamente desde `region`, el ASCP monta ese valor de secreto. Si el secreto no se recupera correctamente desde `region`, pero sí desde `failoverRegion`, el ASCP monta ese valor de secreto.

failoverRegion

(Opcional) Si se incluye este campo, la ASCP intenta recuperar el secreto desde las regiones definidas en `region` y este campo. Si alguna de estas regiones devuelve un error 4xx, por ejemplo por un problema de autenticación, el ASCP no monta ninguno de los secretos. Si el secreto se recupera correctamente desde `region`, el ASCP monta ese valor de secreto. Si el secreto no se recupera correctamente desde `region`, pero sí desde `failoverRegion`, el ASCP monta ese valor de secreto. Para ver un ejemplo sobre cómo utilizar este campo, consulte Definir una región de conmutación por error para un secreto de varias regiones.

pathTranslation

(Opcional) Un único carácter de sustitución para utilizarlo si el nombre del archivo de Amazon EKS contiene el carácter separador de ruta, por ejemplo la barra diagonal (/) en Linux. El ASCP no puede crear un archivo montado que contenga un carácter separador de ruta. En su lugar, el ASCP reemplaza el carácter separador de ruta por otro carácter. Si no se utiliza este campo, el carácter de reemplazo es el guion bajo (_), de modo que, por ejemplo, `My/Path/Secret` se monta como `My_Path_Secret`.

Para evitar la sustitución de caracteres, ingrese la cadena `False`.

objects

Una cadena que contiene una declaración YAML de los secretos que se van a montar. Se recomienda utilizar una cadena de varias líneas de YAML o una barra vertical (|).

objectName

El nombre o el ARN completo del secreto. Si utiliza el ARN, puede omitir el campo `objectType`. Este campo se convierte en el nombre de archivo del secreto en el pod de Amazon EKS, a menos que se especifique `objectAlias`. Si se utiliza un ARN, la región

del ARN debe coincidir con el campo `region`. Si se incluye `failoverRegion`, este campo representa el campo `objectName` principal.

objectType

Es requerido si no utiliza un ARN de Secrets Manager para `objectName`. Puede ser `secretsmanager` o `ssmparameter`.

objectAlias

(Opcional) El nombre de archivo del secreto en el pod de Amazon EKS. Si no especifica este campo, el `objectName` aparece como nombre de archivo.

objectVersion

(Opcional) El ID de versión del secreto. No se recomienda, porque se debe actualizar el ID de versión cada vez que se actualice el secreto. Se utiliza la versión más reciente de forma predeterminada. Si se incluye `failoverRegion`, este campo representa el campo `objectVersion` principal.

objectVersionLabel

(Opcional) El alias de la versión. El valor predeterminado es la versión más reciente AWSCURRENT. Para obtener más información, consulte the section called “Versión”. Si se incluye `failoverRegion`, este campo representa el campo `objectVersionLabel` principal.

jmesPath

(Opcional) Un mapa de las claves en el secreto a los archivos que se van a montar en Amazon EKS. Para utilizar este campo, el valor secreto debe estar en formato JSON. Si utiliza este campo, debe incluir los subcampos `path` y `objectAlias`.

ruta

Una clave de un par clave-valor en el JSON del valor secreto. Si el campo contiene un guion, aplique escape con comillas simples, por ejemplo: `path: '"hyphenated-path"'`

objectAlias

Nombre de archivo que se va a montar en el pod de Amazon EKS. Si el campo contiene un guion, aplique escape con comillas simples, por ejemplo: `objectAlias: '"hyphenated-alias"'`

failoverObject

(Opcional) Si se especifica este campo, el ASCP intenta recuperar tanto el secreto especificado en el campo `objectName` principal como el secreto especificado en el subcampo `failoverObject` `objectName`. Si alguno devuelve un error 4xx, por ejemplo por un problema de autenticación, el ASCP no monta ninguno de los secretos. Si el secreto se recupera correctamente desde el campo `objectName` principal, el ASCP monta ese valor de secreto. Si el secreto no se recupera correctamente desde el campo `objectName` principal, pero sí desde el `objectName` de conmutación por error, el ASCP monta ese valor de secreto. Si se incluye este campo, se debe incluir el campo `objectAlias`. Para ver un ejemplo sobre cómo utilizar este campo, consulte Seleccionar un secreto de conmutación por error para montarlo.

Este campo se suele utilizar cuando el secreto de conmutación por error no es una réplica. Para ver un ejemplo sobre cómo especificar una réplica, consulte Definir una región de conmutación por error para un secreto de varias regiones.

objectName

Nombre o ARN completo del secreto de conmutación por error. Si se utiliza un ARN, la región del ARN debe coincidir con el campo `failoverRegion`.

objectVersion

(Opcional) El ID de versión del secreto. Debe coincidir con el campo `objectVersion` principal. No se recomienda, porque se debe actualizar el ID de versión cada vez que se actualice el secreto. Se utiliza la versión más reciente de forma predeterminada.

objectVersionLabel

(Opcional) El alias de la versión. El valor predeterminado es la versión más reciente AWSCURRENT. Para obtener más información, consulte the section called “Versión”.

Usa AWS Secrets Manager secretos en los GitHub trabajos

Para usar un secreto en un GitHub trabajo, puedes usar una GitHub acción para recuperar datos secretos AWS Secrets Manager y añadirlos como variables de entorno enmascaradas en tu GitHub flujo de trabajo. Para obtener más información sobre GitHub las acciones, consulte Descripción de GitHub las acciones en los GitHub documentos.

Cuando añades un secreto a tu GitHub entorno, estará disponible para todos los demás pasos de tu GitHub trabajo. Siga las instrucciones de Security Hardening for GitHub Actions para evitar que se haga un uso indebido de los secretos de su entorno.

Puede establecer la cadena completa del valor del secreto como el valor de la variable de entorno o, si la cadena es JSON, puede analizar el elemento JSON para establecer variables de entorno individuales para cada par clave-valor de JSON. Si el valor del secreto es binario, la acción lo convierte en una cadena.

Para ver las variables de entorno creadas a partir de sus secretos, active el registro de depuración. Para obtener más información, consulta la sección sobre cómo habilitar el registro de depuración en los documentos. GitHub

Para usar las variables de entorno creadas a partir de tus secretos, consulta Variables de entorno en los GitHub documentos.

Requisitos previos

Para usar esta acción, primero debes configurar AWS las credenciales y configurarlas Región de AWS en tu GitHub entorno siguiendo este `configure-aws-credentials` paso. Siga las instrucciones de la acción Configurar AWS credenciales para que GitHub las acciones asuman el rol directamente mediante el proveedor GitHub OIDC. Esto permite utilizar credenciales de corta duración y evitar almacenar claves de acceso adicionales fuera de Secrets Manager.

El rol de IAM que asume la acción debe tener los siguientes permisos:

- `GetSecretValue` sobre los secretos que quiere recuperar.
- `ListSecrets` sobre todos los secretos.
- (Opcional) KMS key si `Decrypt` los secretos están cifrados con un. clave administrada por el cliente

Para obtener más información, consulte Autenticación y control de acceso.

Uso

Para utilizar la acción, agregue un paso al flujo de trabajo que emplea la siguiente sintaxis.

```
- 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
```

Parámetros

`secret-ids`

ARN, nombres y prefijos de nombres de los secretos.

De forma predeterminada, el paso crea el nombre de cada variable de entorno a partir del nombre del secreto, transformado para incluir solo letras mayúsculas, números y guiones bajos, de modo que no comience con un número.

Para establecer el nombre de la variable de entorno, escríbalo antes del identificador del secreto, seguido de una coma. Por ejemplo, `ENV_VAR_1, secretId` crea una variable de entorno denominada ENV_VAR_1 a partir del `secretId` del secreto. El nombre de las variables de entorno pueden componerse de letras mayúsculas, números y guiones bajos.

Para usar un prefijo, ingrese al menos tres caracteres seguidos de un asterisco. Por ejemplo, `dev*` hace coincidir todos los secretos con un nombre que comience por dev. El número máximo de secretos coincidentes que pueden recuperarse es de 100. Si establece el nombre de la variable y el prefijo coincide con varios secretos, la acción devuelve un error.

`parse-json-secrets`

(Opcional) De forma predeterminada, la acción establece el valor de la variable de entorno en toda la cadena JSON del valor del secreto. Establezca `parse-json-secrets` en `true` para crear variables de entorno para cada par clave-valor en el archivo JSON.

Tenga en cuenta que, si el archivo JSON utiliza claves que distinguen entre mayúsculas y minúsculas, como “nombre” y “Nombre”, la acción tendrá conflictos de nombres duplicados. En este caso, establezca `parse-json-secrets` en `false` y analice el valor del secreto de JSON por separado.

Denominación de variables de entorno

Las variables de entorno creadas por la acción reciben el mismo nombre que los secretos de los que provienen. Las variables de entorno tienen requisitos de nomenclatura más estrictos que los

secretos, por lo que la acción transforma los nombres secretos para cumplir con esos requisitos. Por ejemplo, la acción transforma las letras minúsculas en mayúsculas. Si analiza el JSON del secreto, el nombre de la variable de entorno incluye tanto el nombre del secreto como el nombre de la clave JSON, por ejemplo`MYSECRET_KEYNAME`.

Si dos variables de entorno terminan con el mismo nombre, la acción fallará. En este caso, debe especificar los nombres que quiere usar para las variables de entorno como alias.

Ejemplos de casos en los que los nombres pueden entrar en conflicto:

- Tanto un secreto llamado «MySecret» como un secreto llamado «mysecret» se convertirían en variables de entorno denominadas «MYSECRET».
- Tanto un secreto denominado «secret_keyname» como un secreto analizado por JSON denominado «Secret» con una clave denominada «keyname» se convertirían en variables de entorno denominadas «SECRET_KEYNAME».

Puede establecer el nombre de la variable de entorno especificando un alias, como se muestra en el siguiente ejemplo, que crea una variable denominada. `ENV_VAR_NAME`

```
secret-ids: |
  ENV_VAR_NAME, secretId2
```

Alias en blanco

- Si estableces `parse-json-secrets: true` e ingresas un alias en blanco, seguido de una coma y, por último, del ID secreto, la acción asignará a la variable de entorno el mismo nombre que a las claves JSON analizadas. Los nombres de las variables no incluyen el nombre secreto.

  Si el secreto no contiene un JSON válido, la acción crea una variable de entorno y le asigna el mismo nombre que el nombre del secreto.
- Si establece `parse-json-secrets: false` e introduce un alias en blanco, seguido de una coma y el ID secreto, la acción asigna un nombre a las variables de entorno como si no hubiera especificado un alias.

En el siguiente ejemplo, se muestra un alias en blanco.

```
,secret2
```

Ejemplos

Example 1. Obtención de secretos por nombre y por ARN

En el ejemplo siguiente, se crean variables de entorno para los secretos identificados por nombre y por ARN.

```
- 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
```

Variables de entorno creadas:

```
EXAMPLESECRETNAME: secretValue1
TEST1: secretValue2
_0_TEST_SECRET: secretValue3
_PROD_EXAMPLE_SECRET: secretValue4
SECRET_ALIAS_1: secretValue5
SECRET_ALIAS_2: secretValue6
SECRET2: secretValue7
```

Example 2. Obtención de todos los secretos que comienzan por un prefijo

El siguiente ejemplo crea variables de entorno para todos los secretos con nombres que comienzan por *beta*.

```
- name: Get Secret Names by Prefix
  uses: 2
  with:
    secret-ids: |
      beta*    # Retrieves all secrets that start with 'beta'
```

Variables de entorno creadas:

```
BETASECRETNAME: secretValue1
BETATEST: secretValue2
BETA_NEWSECRET: secretValue3
```

Example 3. Análisis del archivo JSON en secreto

En el siguiente ejemplo, se crean variables de entorno mediante el análisis del archivo JSON del secreto.

```
- 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
```

El secreto `test/secret` tiene el siguiente valor del secreto.

```
{
  "api_user": "user",
  "api_key": "key",
  "config": {
    "active": "true"
  }
}
```

El secreto `secret2` tiene el siguiente valor del secreto.

```
{
  "myusername": "alejandro_rosalez",
  "mypassword": "EXAMPLE_PASSWORD"
}
```

Variables de entorno creadas:

```
TEST_SECRET_API_USER: "user"
TEST_SECRET_API_KEY: "key"
TEST_SECRET_CONFIG_ACTIVE: "true"
MYUSERNAME: "alejandro_rosalez"
```

```
MYPASSWORD: "EXAMPLE_PASSWORD"
```

Uso de secretos de AWS Secrets Manager en AWS IoT Greengrass

AWS IoT Greengrass es un software que amplía las funcionalidades en la nube a los dispositivos locales. Esto permite que los dispositivos recopilen y analicen datos más cerca del origen de la información, reaccionen de forma autónoma a eventos locales y se comuniquen de forma segura entre sí en las redes locales.

AWS IoT Greengrass le permite autenticarse con servicios y aplicaciones desde dispositivos Greengrass sin contraseñas codificadas de forma rígida, tokens u otros secretos. Puede usar AWS Secrets Manager para almacenar y administrar los secretos de forma segura en la nube. AWS IoT Greengrass amplía Secrets Manager a dispositivos del núcleo Greengrass, de modo que los conectores y las funciones de Lambda puedan utilizar secretos locales para interactuar con los servicios y aplicaciones.

Para integrar un secreto en un grupo de Greengrass, cree un recurso de grupo que haga referencia al secreto de Secrets Manager. Este recurso de secretos hace referencia al secreto en la nube mediante el ARN asociado. Para obtener más información sobre cómo crear, administrar y utilizar recursos de secretos, consulte Trabajar con recursos de secretos en la Guía para desarrolladores de AWS IoT.

Para implementar secretos en el núcleo de AWS IoT Greengrass, consulte Implementación de secretos en núclo de AWS IoT Greengrass.

Usa AWS Secrets Manager secretos en AWS Lambda las funciones

Puede usar la extensión Lambda AWS Parameters and Secrets para recuperar y almacenar en caché AWS Secrets Manager los secretos de las funciones de Lambda sin usar un SDK. Recuperar un secreto almacenado en la memoria caché es más rápido que recuperarlo desde Secrets Manager. Dado que la llamada a las API de Secrets Manager conlleva un costo, el uso de una caché puede reducirlo. La extensión puede recuperar los secretos de Secrets Manager y los parámetros del almacén de parámetros. Para obtener información sobre el almacén de parámetros, consulte Parameter Store integration with Lambda extensions (Integración del almacén de parámetros con las extensiones de Lambda) en la Guía del usuario de AWS Systems Manager .

Una extensión de Lambda es un proceso complementario que se suma a las capacidades de una función de Lambda. Para obtener más información, consulte Lambda extensions (Extensiones de Lambda) en la Guía para desarrolladores de Lambda. Para obtener información sobre el uso de la extensión en una imagen de contenedor, consulte Trabajar con capas y extensiones de Lambda en imágenes de contenedor. Lambda registra la información de ejecución de la extensión junto con la función mediante Amazon CloudWatch Logs. De forma predeterminada, la extensión registra una cantidad mínima de información en. CloudWatch Para registrar más detalles, establezca la variable de entorno `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` en `debug`.

Para proporcionar la caché en memoria para los parámetros y secretos, la extensión expone un punto de conexión HTTP local, el puerto localhost 2773, al entorno Lambda. Para configurar el puerto, establezca la variable de entorno `PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`.

Lambda crea instancias independientes correspondientes al nivel de simultaneidad que requiere la función. Cada instancia está aislada y mantiene su propia memoria caché local de los datos de configuración. Para obtener más información sobre las instancias de Lambda y la simultaneidad, consulte Administración de la simultaneidad reservada de Lambda en la Guía para desarrolladores de Lambda.

Para agregar la extensión para ARM, debe usar la arquitectura `arm64` en la función de Lambda. Para obtener más información, consulte Arquitecturas del conjunto de instrucciones Lambda en la Guía para desarrolladores de Lambda. La extensión admite ARM en la regiones siguientes: Asia-Pacífico (Bombay), Este de EE. UU. (Ohio), Europa (Irlanda), Europa (Fráncfort), Europa (Zúrich), Este de EE. UU. (Norte de Virginia), Europa (Londres), Europa (España), Asia-Pacífico (Tokio), Oeste de EE. UU. (Oregón), Asia-Pacífico (Singapur), Asia-Pacífico (Hyderabad) y Asia-Pacífico (Sídney).

La extensión usa un AWS cliente. Para obtener información sobre la configuración del AWS cliente, consulte la referencia sobre la configuración en la Guía de referencia del AWS SDK y las herramientas. Si su función Lambda se ejecuta en una VPC, debe crear un punto final de VPC para que la extensión pueda realizar llamadas a Secrets Manager. Para obtener más información, consulte Punto de conexión VPC.

Permisos necesarios:

- La función de ejecución de Lambda debe tener `secretsmanager:GetSecretValue` permiso para usar el secreto.
- Si el secreto se cifra con una clave administrada por el cliente en lugar de con Clave administrada de AWS `aws/secretsmanager`, la función de ejecución también necesitará el `kms:Decrypt` permiso para la clave de KMS.

Para usar la extensión AWS Lambda Parameters and Secrets

1. Agregue la capa a la función de la siguiente manera:
   - Abra la AWS Lambda consola en https://console.aws.amazon.com/lambda/.
     a. Elija la función, elija Layers (Capas) y, a continuación, seleccione Add a layer (Agregar una capa).
     b. En la página Agregar capa, para Capas de AWS seleccione Extensión de Lambda para secretos y parámetros de AWS , y luego Agregar.
   - Utilice el siguiente AWS CLI comando con el ARN adecuado para su región. Para obtener una lista de los ARN, consulte AWS Parámetros y secretos de los ARN de la extensión de Lambda en la Guía del usuario AWS Systems Manager .

```
aws lambda update-function-configuration \
    --function-name my-function \
    --layers LayerARN
```

2. Conceda permisos al rol de ejecución de Lambda para poder acceder a los secretos:
   - Permiso `secretsmanager:GetSecretValue` para el secreto. Consulte the section called "Ejemplo: permiso para recuperar valores secretos".
   - (Opcional) Si el secreto está cifrado con una clave gestionada por el cliente en lugar de con Clave administrada de AWS `aws/secretsmanager`, el rol de ejecución también necesitará un `kms:Decrypt` permiso para la clave de KMS.
   - Puede usar el control de acceso basado en atributos (ABAC) con la función de Lambda para permitir un acceso más detallado a los secretos de la cuenta. Para más información, consulte the section called "Ejemplo: controlar el acceso a los secretos mediante etiquetas" y the section called "Ejemplo: limitar el acceso a identidades con etiquetas que coincidan con las etiquetas de los secretos".
3. Configure la memoria caché con variables de entorno de Lambda.
4. Para recuperar los secretos de la caché de extensiones, primero debe agregar `X-AWS-Parameters-Secrets-Token` al encabezado de la solicitud. Configure el token en `AWS_SESSION_TOKEN`, que Lambda proporciona para todas las funciones en ejecución. El uso de este encabezado indica que el intermediario se encuentra en el entorno de Lambda.

   El siguiente ejemplo de Python muestra cómo agregar el encabezado.

```
import os
```

```
headers = {"X-Aws-Parameters-Secrets-Token": os.environ.get('AWS_SESSION_TOKEN')}
```

5. Para recuperar un secreto en la función de Lambda, utilice una de las siguientes solicitudes HTTP GET:

   - Para recuperar un secreto, utilice el ARN o nombre del secreto para `secretId`.

     ```
     GET: /secretsmanager/get?secretId=secretId
     ```

   - Para recuperar el valor de secreto anterior o una versión específica por etiqueta provisional, utilice el ARN o nombre del secreto para `secretId` y la etiqueta provisional para `versionStage`.

     ```
     GET: /secretsmanager/get?secretId=secretId&versionStage=AWSPREVIOUS
     ```

   - Para recuperar una versión de secreto específica por ID, utilice el ARN o nombre del secreto para `secretId` y el ID de versión para `versionId`.

     ```
     GET: /secretsmanager/get?secretId=secretId&versionId=versionId
     ```

   Example Recuperar un secreto (Python)

   El siguiente ejemplo de Python muestra cómo recuperar un secreto y analizar el resultado mediante `json.loads`.

   ```
   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 Parámetros y secretos Variables de entorno de Lambda Extension

Puede configurar la extensión con las siguientes variables de entorno.

Para obtener información sobre cómo usar las variables de entorno, consulte Uso de variables de entorno de Lambda en la Guía para desarrolladores de Lambda.

`PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED`

Establezca el valor en true para almacenar en caché los parámetros y secretos. Se establece en false para no almacenar en caché. El valor predeterminado es verdadero.

`PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE`

La cantidad máxima de secretos y parámetros que almacenar en caché. Debe ser un valor entre 0 y 1000. Un valor de 0 indica que no hay almacenamiento en la memoria caché. Esta variable se ignora si los valores de `SSM_PARAMETER_STORE _TTL` y `SECRETS_MANAGER_TTL` son 0. El valor predeterminado es 1000.

`PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`

El puerto del servidor HTTP local. El valor predeterminado es 2773.

`PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL`

El nivel de registro que proporciona la extensión: `debug`, `info`, `warn`, `error` o `none`. Establezca esta opción en `debug` para ver la configuración de la memoria caché. El valor predeterminado es `info`.

`PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS`

Cantidad máxima de conexiones para los clientes HTTP que la extensión utiliza para hacer solicitudes al almacén de parámetros o Secrets Manager. Se trata de una configuración por cliente. El valor predeterminado es 3.

`SECRETS_MANAGER_TIMEOUT_MILLIS`

Tiempo de espera para las solicitudes a Secrets Manager en milisegundos. Un valor de 0 indica que no hay tiempo de espera. El valor predeterminado es 0.

`SECRETS_MANAGER_TTL`

TTL de un secreto en la memoria caché en segundos. Un valor de 0 indica que no hay almacenamiento en la memoria caché. El máximo es de 300 segundos. Esta variable se ignora si `PARAMETERS_SECRETS_CACHE_SIZE` es 0. El valor predeterminado es de 300 segundos.

`SSM_PARAMETER_STORE_TIMEOUT_MILLIS`

Tiempo de espera para las solicitudes al almacén de parámetros en milisegundos. Un valor de 0 indica que no hay tiempo de espera. El valor predeterminado es 0.

`SSM_PARAMETER_STORE_TTL`

TTL de un parámetro de la caché en segundos. Un valor de 0 indica que no hay almacenamiento en la memoria caché. El máximo es de 300 segundos. Esta variable se ignora si `PARAMETERS_SECRETS_CACHE_SIZE` es 0. El valor predeterminado es de 300 segundos.

Uso de secretos de AWS Secrets Manager en Parameter Store

El Parameter Store de Systems Manager de AWS proporciona un almacenamiento seguro y jerárquico para administrar los datos de configuración y administrar los secretos. Puede almacenar datos como contraseñas, cadenas de base de datos y códigos de licencia como valores de parámetros. No obstante, el Almacén de parámetros no proporciona servicios de rotación automática para los secretos almacenados. En su lugar, Parameter Store le permite almacenar el secreto en Secrets Manager y, a continuación, hacer referencia al secreto como parámetro de Parameter Store.

Cuando se configura Parameter Store con Secrets Manager, el `secret-id` de Parameter Store necesita que se incluya una barra diagonal (/) antes de la cadena de nombre.

Para obtener más información, consulte Referencia a los secretos de AWS Secrets Manager desde los parámetros de Paremeter Store en la Guía del usuario de AWS Systems Manager.

Rotar secretos de AWS Secrets Manager

La rotación es el proceso de actualización periódica de un secreto. Cuando Secrets Manager rota un secreto, se actualizan las credenciales tanto en el secreto como en la base de datos o el servicio. En Secrets Manager, puede configurar la rotación automática de sus secretos.

Temas

Cómo funciona la rotación

> **Tip**
>
> En algunos Secretos gestionados por otros servicios, se utiliza la rotación administrada. Para utilizar Rotación administrada, primero se debe crear el secreto a través del servicio de administración.

La rotación de Secrets Manager utiliza una AWS Lambda función para actualizar el secreto y la base de datos o el servicio. Para obtener información sobre los costos por usar una función de Lambda, consulte Precios.

Para rotar un secreto, Secrets Manager llama a una función de Lambda según la programación que haya configurado. Se puede establecer una programación para realizar la rotación al cabo de cierto tiempo, por ejemplo cada 30 días, o bien crear una expresión cron. Consulte Programación de

expresiones. Si también se actualiza manualmente el valor de secreto mientras está configurada la rotación automática, Secrets Manager la considerará una rotación válida cuando calcule la próxima fecha de rotación.

Por motivos de seguridad, Secrets Manager solo permite que una función de rotación de Lambda rote el secreto de forma directa. La función de rotación no puede llamar a una segunda función de Lambda para rotar el secreto.

Secrets Manager utiliza etiquetas provisionales para identificar las versiones de un secreto durante la rotación. Durante la rotación, Secrets Manager llama a la misma función varias veces, cada una con diferentes parámetros. Secrets Manager invoca la función con la siguiente estructura de parámetros de solicitud JSON:

```
{
    "Step" : "request.type",
    "SecretId" : "string",
    "ClientRequestToken" : "string"
}
```

La función de rotación hace el trabajo de rotar el secreto. Hay cuatro pasos para rotar un secreto, que se corresponden con los cuatro pasos de la función de rotación de Lambda que se incluyen a continuación:

1. Crear una nueva versión del secreto (**`createSecret`**)

   El primer paso de rotación es crear una versión nueva del secreto. En las plantillas de rotación de bases de datos proporcionadas por Secrets Manager, la función de Lambda de rotación genera una contraseña de 32 caracteres para la nueva versión. La nueva versión puede contener una nueva contraseña, una contraseña y un nombre de usuario nuevos o más información del secreto. La función de Lambda de rotación etiqueta la nueva versión `AWSPENDING`.

2. Cambiar las credenciales en la base de datos o el servicio (**`setSecret`**)

   A continuación, la función de Lambda de rotación cambia las credenciales en la base de datos o el servicio para que coincidan con las nuevas credenciales de la versión de `AWSPENDING` del secreto. En función de la estrategia de rotación, este paso puede crear un usuario nuevo con los mismos permisos que el usuario existente.

   Las funciones de rotación de Amazon RDS (a excepción de Oracle y Db2) y Amazon DocumentDB utilizan una capa de sockets seguros (SSL) o una seguridad de la capa de

transporte (TLS) de forma automática para conectarse a su base de datos, si está disponible. De lo contrario, utilizan una conexión no cifrada.

> **Note**
>
> Si configuró la rotación automática de secretos antes del 20 de diciembre de 2021, es posible que la función de rotación se base en una plantilla más antigua que no sea compatible con SSL/TLS. Consulte Determinar cuándo se creó la función de rotación. Si se creó antes del 20 de diciembre de 2021, es necesario Crear la función de rotación nuevamente para que sea compatible con las conexiones que utilizan SSL/TLS.

3. Probar la nueva versión del secreto (**`testSecret`**)

   A continuación, la función de Lambda de rotación comprueba la versión de `AWSPENDING` del secreto utilizándolo para acceder a la base de datos o el servicio. Funciones de rotación basadas en Plantillas de función de rotación prueban el nuevo secreto mediante el acceso de lectura. Según el tipo de acceso que necesiten las aplicaciones, puede actualizar la función para incluir otros accesos, como el acceso de escritura.

4. Finalizar la rotación (**`finishSecret`**)

   Por último, la función de Lambda de rotación mueve la etiqueta `AWSCURRENT` de la versión secreta anterior a esta versión, que también elimina la etiqueta `AWSPENDING` en la misma llamada a la API. No debe eliminar `AWSPENDING` antes de este punto y no debe eliminarlo mediante una llamada independiente a la API, ya que eso puede indicar a Secrets Manager que la rotación no se completó correctamente. Secrets Manager agrega la etiqueta provisional de `AWSPREVIOUS` a la versión anterior, para que usted conserve la última versión buena conocida del secreto.

Durante la rotación, Secrets Manager registra los eventos que indican el estado de rotación. Para obtener más información, consulte the section called “Inicio de sesión con AWS CloudTrail ”.

Si algún paso de la rotación falla, Secrets Manager vuelve a intentar todo el proceso de rotación varias veces.

Si la rotación se realiza correctamente, es posible que se asocie la etiqueta provisional `AWSPENDING` a la misma versión que la versión de `AWSCURRENT`, o que no se asocie a ninguna versión. Si la etiqueta provisional `AWSPENDING` está presente pero no está asociada a la misma versión que `AWSCURRENT`, cualquier invocación posterior de la rotación presupone que existe una solicitud de rotación anterior aún en curso y se devuelve un error. Si la rotación no se realiza correctamente,

es posible que se asocie la etiqueta provisional `AWSPENDING` a una versión de secreto vacía. Para obtener más información, consulte Solución de problemas de rotación de .

Una vez que la rotación se haya realizado correctamente, las aplicaciones que Recuperar secretos de AWS Secrets Manager desde Secrets Manager obtienen automáticamente las credenciales actualizadas. Para obtener más información acerca de cómo funciona cada paso de rotación, consulte las the section called “Plantillas de función de rotación”.

Rotación administrada para secretos de AWS Secrets Manager

Algunos servicios ofrecen rotación administrada, que permite que el servicio se encargue de configurar y administrar la rotación. En la rotación administrada, no se utiliza una función de AWS Lambda para actualizar el secreto y las credenciales de la base de datos. Los siguientes servicios ofrecen rotación administrada:

- Amazon ECS Service Connect ofrece la rotación gestionada de los certificados AWS Private Certificate Authority TLS. Para obtener más información, consulte TLS con Service Connect en la Guía para desarrolladores de Amazon Elastic Container Service.
- Amazon RDS ofrece rotación administrada para las credenciales de usuario maestras. Para obtener más información, consulte Administración de contraseñas con Amazon RDS y AWS Secrets Manager en la Guía del usuario de Amazon RDS.
- Amazon Aurora ofrece rotación administrada para las credenciales de usuario maestras. Para obtener más información, consulte Administración de contraseñas con Amazon Aurora y AWS Secrets Manager en la Guía del usuario de Amazon Aurora.
- Amazon Redshift ofrece rotación gestionada para contraseñas de administrador. Para obtener más información, consulte Administración de contraseñas de administrador de Amazon Redshift mediante AWS Secrets Manager en la Guía de administración de Amazon Redshift.

Para los demás tipos de secretos, consulte Rotar secretos de .

La rotación de los secretos gestionados generalmente se completa en un minuto. Durante la rotación, las nuevas conexiones que recuperan el secreto pueden obtener la versión anterior de las credenciales. En las aplicaciones, es muy recomendable respetar la práctica recomendada de utilizar un usuario de base de datos creado con los privilegios mínimos necesarios para su aplicación, en lugar de utilizar el usuario maestro. En el caso de los usuarios de la aplicación, para obtener la máxima disponibilidad, se puede utilizar la estrategia de rotación alterna de usuarios.

Para cambiar la programación de la rotación administrada (consola)

1. Abra el secreto administrado en la consola de Secrets Manager. Puede seguir un enlace del servicio de administración, o bien buscar el secreto en la consola de Secrets Manager.
2. En Rotation schedule (Programación de rotación), ingrese la programación en zona horaria UTC en Schedule expression builder (Generador de expresiones de programación) o como una expresión de programación. Secrets Manager almacena su programación como una expresión `rate()` o `cron()`. El periodo de rotación se inicia a medianoche de forma automática, excepto si se especifica una Start time (Hora de inicio). Se puede rotar un secreto con una frecuencia máxima de cuatro horas. Para obtener más información, consulte Programación de expresiones.
3. (Opcional) En Window duration (Duración del periodo), elija el tiempo durante el cual desea que Secrets Manager rote su secreto, por ejemplo, **3h**, para un periodo de tres horas. El periodo no debe prolongarse hasta el siguiente periodo de rotación. Si no se especifica Window duration (Duración del periodo) para una programación de rotación en horas, el periodo concluye automáticamente al cabo de una hora. Para una programación de rotación en días, el periodo concluye automáticamente al final del día.
4. Seleccione Guardar.

Para cambiar la programación de la rotación administrada (AWS CLI)

- Llamar a `rotate-secret`. En el siguiente ejemplo se rota el secreto entre las 16:00 h y las 18:00 h UTC del día 1 y 15 del mes. Para obtener más información, consulte Programación de expresiones.

```
aws secretsmanager rotate-secret \
    --secret-id MySecret \
    --rotation-rules "{\"ScheduleExpression\": \"cron(0 16 1,15 * ? *)\",
 \"Duration\": \"2h\"}"
```

Configurar la rotación automática de secretos de Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB con la consola

La rotación es el proceso de actualización periódica de un secreto. Cuando se rota un secreto, se actualizan las credenciales tanto en el secreto como en la base de datos. En Secrets Manager, puede configurar la rotación automática de sus secretos de bases de datos.

Secrets Manager utiliza funciones de Lambda para rotar los secretos. Para obtener una descripción general, consulte the section called “Cómo funciona la rotación”.

> **Tip**
>
> En algunos Secretos gestionados por otros servicios, se utiliza la rotación administrada. Para utilizar Rotación administrada, primero se debe crear el secreto a través del servicio de administración.

Para configurar la rotación con la consola, primero debe elegir una estrategia de rotación. A continuación, configure el secreto para la rotación, lo que crea una función de rotación de Lambda si aún no la tiene. La consola también establece los permisos para el rol de ejecución de la función de Lambda. El último paso consiste en asegurarse de que la función de rotación de Lambda pueda acceder tanto a Secrets Manager como a su base de datos a través de la red.

Para activar la rotación automática, debe tener permisos para crear el rol de ejecución de IAM y adjuntarle una política de permisos. Necesita ambos permisos, `iam:CreateRole` y `iam:AttachRolePolicy`.

> **Warning**
>
> Conceder a una identidad los permisos `iam:CreateRole` y `iam:AttachRolePolicy` permite que la identidad se conceda a sí misma cualquier permiso.

Pasos:

- Paso 1: elegir una estrategia de rotación y (opcionalmente) crear un secreto de superusuario
- Paso 2: configurar la rotación y crear una función de rotación

Paso 1: elegir una estrategia de rotación y (opcionalmente) crear un secreto de superusuario

En el caso de Amazon RDS, Amazon Redshift y Amazon DocumentDB, Secrets Manager ofrece dos estrategias de rotación:

Estrategia de rotación de un solo usuario

Esta estrategia actualiza las credenciales de un usuario en un secreto. En el caso de las instancias Db2 de Amazon RDS, dado que los usuarios no pueden cambiar sus propias contraseñas, debe proporcionar las credenciales de administrador en un secreto independiente. Esta es la estrategia de rotación más sencilla y es adecuada para la mayoría de los casos de uso. En particular, recomendamos que utilice esta estrategia para las credenciales de los usuarios interactivos o únicos (ad hoc).

Cuando el secreto rota, las conexiones de bases de datos abiertas no se eliminan. Mientras se produce la rotación, hay un breve periodo de tiempo entre el momento en que cambia la contraseña de la base de datos y el momento en que se actualiza el secreto. Durante este tiempo, existe un riesgo bajo de que la base de datos deniegue las llamadas que utilizan las credenciales rotadas. Puede mitigar este riesgo con una estrategia de reintentos apropiada. Tras la rotación, las nuevas conexiones utilizan las nuevas credenciales.

Estrategia de rotación de usuarios alternativos

Esta estrategia actualiza las credenciales de dos usuarios en un secreto. Se crea el primer usuario y, durante la primera rotación, la función de rotación lo clona para crear el segundo usuario. Cada vez que el secreto rota, la función de rotación alterna la contraseña de usuario que actualiza. Dado que la mayoría de los usuarios no tienen permiso para clonarse a sí mismos, debe proporcionar las credenciales de un usuario de tipo `superuser` en otro secreto. Recomendamos que utilice la estrategia de rotación de un solo usuario cuando los usuarios clonados en su base de datos no tienen los mismos permisos que el usuario original y para las credenciales de los usuarios interactivos o únicos (ad hoc).

Esa estrategia es adecuada para bases de datos con modelos de permisos en los que un rol es propietario de las tablas de base de datos y un segundo rol tiene permiso para acceder a las tablas de base de datos. También es adecuada para aplicaciones que requieren alta disponibilidad. Si una aplicación recupera el secreto durante la rotación, seguirá obteniendo un conjunto de credenciales válido. Tras la rotación, las credenciales de `user` y `user_clone` son válidas. Incluso hay menos posibilidades de que las aplicaciones sufran denegaciones durante este tipo de rotación que con la rotación de un solo usuario. Si la base de datos está alojada en una granja de servidores donde el cambio de contraseña tarda tiempo en propagarse a todos los servidores, existe el riesgo de que la base de datos deniegue las llamadas que utilicen las nuevas credenciales. Puede mitigar este riesgo con una estrategia de reintentos apropiada.

Secrets Manager crea el usuario clonado con los mismos permisos que el usuario original. Si cambia los permisos del usuario original después de crear el clon, también debe cambiar los permisos del usuario clonado.

> **Important**
>
> Si elige la estrategia de usuarios alternativos, debe Creación de un secreto de base de datos y almacenar en él las credenciales de superusuario de la base de datos. Necesita un secreto con credenciales de superusuario porque la rotación clona el primer usuario y la mayoría de los usuarios no tienen ese permiso.

Paso 2: configurar la rotación y crear una función de rotación

Las funciones de rotación de Amazon RDS (a excepción de Oracle y Db2) y Amazon DocumentDB utilizan una capa de sockets seguros (SSL) o una seguridad de la capa de transporte (TLS) de forma automática para conectarse a su base de datos, si está disponible. De lo contrario, utilizan una conexión no cifrada.

Activar la rotación de un secreto de Amazon RDS, Amazon DocumentDB o Amazon Redshift

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la página Secrets (Secretos), elija el secreto.
3. En la página Secret details (Detalles del secreto), en la sección Rotation configuration (Configuración de la rotación), elija Edit rotation (Editar rotación).
4. En el cuadro de diálogo Edit rotation configuration (Configuración para editar la rotación), haga lo siguiente:

a. Active Automatic rotation (Rotación automática).

b. En Rotation schedule (Programación de rotación), ingrese la programación en zona horaria UTC en Schedule expression builder (Generador de expresiones de programación) o como una expresión de programación. Secrets Manager almacena su programación como una expresión `rate()` o `cron()`. El periodo de rotación se inicia a medianoche de forma automática, excepto si se especifica una Start time (Hora de inicio). Se puede rotar un secreto con una frecuencia máxima de cuatro horas. Para obtener más información, consulte Programación de expresiones.

c. (Opcional) En Window duration (Duración del periodo), elija el tiempo durante el cual desea que Secrets Manager rote su secreto, por ejemplo, **3h**, para un periodo de tres horas. El periodo no debe prolongarse hasta el siguiente periodo de rotación. Si no se especifica Window duration (Duración del periodo) para una programación de rotación en horas, el periodo concluye automáticamente al cabo de una hora. Para una programación de rotación en días, el periodo concluye automáticamente al final del día.

d. (Opcional) Elija Rotate immediately when the secret is stored (Rotar inmediatamente cuando se almacene el secreto) a fin de rotar su secreto en cuanto guarde los cambios. Si desmarca la casilla de verificación, la primera rotación comenzará conforme a la programación establecida.

   Si se produce un error en la rotación (por ejemplo, porque los pasos 3 y 4 aún no se han completado), Secrets Manager reintenta el proceso de rotación varias veces.

e. En Rotation function (Función de rotación), realice una de las siguientes operaciones:

   - Elija Create a new Lambda function (Crear una nueva función de Lambda) y luego ingrese un nombre para la nueva función. Secrets Manager agrega `SecretsManager` al principio del nombre de la función. Secrets Manager crea la función a partir de la plantilla adecuada y establece los permisos necesarios para el rol de ejecución de Lambda.
   - Seleccione Use an existing Lambda function (Usar una función de Lambda existente) para reutilizar una función de rotación utilizada para otro secreto. Las funciones de rotación enumeradas en Recommended VPC configurations (Configuraciones recomendadas de VPC) tienen la misma VPC y el mismo grupo de seguridad que la base de datos, lo que facilita a la función el acceso a la base de datos.

f. Para la estrategia de rotación, elija la estrategia de usuario único o la de usuarios alternos. Para obtener más información, consulte the section called “Paso 1: elegir una estrategia de rotación y (opcionalmente) crear un secreto de superusuario”.

5. Seleccione Guardar.

Paso 3 (opcional): establecer condiciones de permisos adicionales en la función de rotación

En la política de recursos de la función de rotación, se recomienda incluir la clave de contexto `aws:SourceAccount` para poder evitar que Lambda se utilice como suplente confuso. En el caso de algunos servicios de AWS, para evitar un escenario de suplente confuso, AWS recomienda que se utilicen las claves de condición globales `aws:SourceArn` y `aws:SourceAccount`. No obstante, si se incluye la condición `aws:SourceArn` en la política de la función de rotación, la función de rotación solo se puede utilizar para rotar el secreto especificado por ese ARN. Se recomienda incluir solo la clave de contexto `aws:SourceAccount`, para poder utilizar la función de rotación para varios secretos.

Actualizar la política de recursos de la función de rotación

1. En la consola de Secrets Manager, elija el secreto y, a continuación, en la página de detalles, en Rotation configuration (Configuración de la rotación), elija la función de rotación de Lambda. Se abre la consola de Lambda.
2. Siga las instrucciones que se describen en Uso de políticas basadas en recursos para Lambda para agregar una condición `aws:sourceAccount`.

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

Si el secreto está cifrado con una clave de KMS distinta de Clave administrada de AWS `aws/secretsmanager`, Secrets Manager concede permiso al rol de ejecución de Lambda para utilizar la clave. Puede utilizar el contexto de cifrado SecretARN para limitar el uso de la función de descifrado, de modo que el rol de la función de rotación solo tenga acceso para descifrar el secreto que es responsable de rotar.

Para actualizar el rol de ejecución de la función de rotación

1. En la función de rotación de Lambda, elija Configuración y, a continuación, en Rol de ejecución, elija el Nombre del rol.
2. Siga las instrucciones que se indican en Modificación de una política de permisos de rol para agregar una condición `kms:EncryptionContext:SecretARN`.

```
"Condition": {
    "StringEquals": {
        "kms:EncryptionContext:SecretARN": "SecretARN"
    }
},
```

Paso 4: configurar el acceso a la red para la función de rotación

Para poder rotar un secreto, la función de rotación de Lambda debe poder acceder tanto al secreto como a la base de datos o el servicio.

Acceder a un secreto

La función de rotación de Lambda debe ser capaz de acceder a un punto de enlace de Secrets Manager. Si la función de Lambda puede acceder a Internet, puede utilizar un punto de enlace público. Para buscar un punto de conexión, consulte the section called “Puntos de conexión de Secrets Manager”.

Si la función de Lambda se ejecuta en una VPC que no tiene acceso a Internet, recomendamos configurar los puntos de enlace privados del servicio de Secrets Manager dentro de la VPC. La VPC puede interceptar entonces las solicitudes dirigidas al punto de enlace regional público y redirigirlas al punto de enlace privado. Para obtener más información, consulte Punto de conexión VPC.

También puede habilitar la función de Lambda para acceder a un punto de conexión público de Secrets Manager. Para ello, agregue una puerta de enlace NAT o una puerta de enlace de Internet a su VPC. Esto permite que el tráfico de la VPC alcance el punto de conexión público. Esto expone a la VPC a más riesgo, ya que desde la red pública de Internet se puede atacar la dirección IP de la gateway.

Acceder a la base de datos o al servicio

Si la base de datos o el servicio se ejecutan en una instancia de Amazon EC2 en una VPC, es recomendable configurar la función de Lambda para que se ejecute en la misma VPC. A continuación, la función de rotación puede comunicarse directamente con el servicio. Para obtener más información, consulte Configuración del acceso a la VPC.

Para permitir que la función de Lambda tenga acceso a la base de datos o el servicio, debe asegurarse de que los grupos de seguridad adjuntos a la función de rotación de Lambda permitan conexiones salientes a la base de datos o el servicio. Asimismo, debe asegurarse de que los grupos de seguridad adjuntos a la base de datos o el servicio permitan conexiones entrantes desde la función de rotación de Lambda.

Para la rotación alternativa de los usuarios donde otro AWS servicio es el que gestiona el superusuario, Lambda debe poder llamar al punto de conexión del servicio de Amazon RDS para obtener la información de conexión de la base de datos. Recomendamos configurar un punto de conexión de VPC para el servicio de base de datos. Para obtener más información, consulte:

- Puntos de conexión de VPC de la API y la interfaz de Amazon RDS en la Guía de usuario de Amazon RDS.
- Cómo trabajar con puntos de conexión de VPC en la Guía de administración de Amazon Redshift.

Paso 5: (Opcional) Personalizar la función de rotación

En raras ocasiones, es posible que desee personalizar la función de rotación. Por ejemplo, al alternar la rotación de los usuarios, Secrets Manager crea el usuario clonado copiando los parámetros de configuración del tiempo de ejecución del primer usuario. Si desea incluir más atributos o cambiar los que se otorgan al usuario clonado, debe actualizar el código de la función. `set_secret`

Por otro ejemplo, para Amazon RDS MySQL, al alternar la rotación de usuarios, Secrets Manager crea un usuario clonado con un nombre de no más de 16 caracteres. Puede modificar la función de rotación para permitir nombres de usuario más largos. La versión 5.7 y superior de MySQL admiten nombres de usuario de hasta 32 caracteres, sin embargo, Secrets Manager añade «_clone» (seis caracteres) al final del nombre de usuario, por lo que debe mantener el nombre de usuario con un máximo de 26 caracteres.

Abrir la función de rotación de Lambda para editarla

1. En la consola de Secrets Manager, elija el secreto.
2. En la sección Rotation configuration (Configuración de la rotación), en Lambda rotation function (Función de rotación de Lambda), elija su función de rotación.

   Se abre la consola de Lambda.

   - Para cambiar el código de la función, desplácese hacia abajo hasta la sección Código fuente.
   - Para MySQL 5.7 y versiones posteriores, para alternar la rotación de usuarios, para cambiar la longitud máxima del nombre de usuario, en Variables de entorno, cambie `USERNAME_CHARACTER_LIMIT`.

Siguientes pasos

Consulte the section called “Solución de problemas de rotación de ”.

Configurar la rotación automática de los secretos de AWS Secrets Manager con la consola

La rotación es el proceso de actualización periódica de un secreto. Cuando se rota un secreto, se actualizan las credenciales tanto en el secreto como en la base de datos o el servicio para el que está destinado el secreto.

Secrets Manager utiliza funciones de Lambda para rotar los secretos. Para obtener una descripción general, consulte the section called “Cómo funciona la rotación”.

También puede utilizar AWS CLI para configurar la rotación. Para obtener más información, consulte Rotación automática (AWS CLI).

A fin de configurar la rotación mediante la consola, primero configure el secreto para la rotación. Durante ese paso, cree también una función de rotación de Lambda vacía. A continuación, establezca los permisos para la función de rotación y para el rol de ejecución de Lambda. Luego, escriba el código de la función de rotación. El último paso consiste en asegurarse de que la función de rotación de Lambda pueda acceder tanto a Secrets Manager como a su base de datos o servicio a través de la red.

En el caso de los secretos de bases de datos, consulte the section called “Rotación automática de secretos de bases de datos (consola)”.

Para activar la rotación automática, debe tener permisos para crear el rol de ejecución de IAM y adjuntarle una política de permisos. Necesita ambos permisos, `iam:CreateRole` y `iam:AttachRolePolicy`.

> **Warning**
>
> Conceder a una identidad los permisos `iam:CreateRole` y `iam:AttachRolePolicy` permite que la identidad se conceda a sí misma cualquier permiso.

Pasos:

- Paso 1: configurar el secreto para la rotación
- Paso 2: establecer permisos para la función de rotación
- Paso 3: (Opcional) establecer una condición de permiso adicional en la función de rotación
- Paso 4: configurar el acceso a la red para la función de rotación
- Paso 5: escribir el código de la función de rotación
- Siguientes pasos

Paso 1: configurar el secreto para la rotación

En este paso, establecerá una programación de rotación para su secreto y creará una función de rotación vacía. Su secreto no se rotará hasta que termine de escribir la función de rotación. Si programa la rotación antes de que se escriba la función de rotación, o si esta falla por algún motivo, Secrets Manager reintentará la función de rotación varias veces.

Configuración de la rotación y creación de una función de rotación vacía

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. En la página Secrets (Secretos), elija el secreto.
3. En la página Secret details (Detalles del secreto), en la sección Rotation configuration (Configuración de la rotación), elija Edit rotation (Editar rotación). En el cuadro de diálogo Edit rotation configuration (Configuración para editar la rotación), haga lo siguiente:

    a. Active Automatic rotation (Rotación automática).

b. En Rotation schedule (Programación de rotación), ingrese la programación en zona horaria UTC en Schedule expression builder (Generador de expresiones de programación) o como una expresión de programación. Secrets Manager almacena su programación como una expresión `rate()` o `cron()`. El periodo de rotación se inicia a medianoche de forma automática, excepto si se especifica una Start time (Hora de inicio). Se puede rotar un secreto con una frecuencia máxima de cuatro horas. Para obtener más información, consulte Programación de expresiones.

c. (Opcional) En Window duration (Duración del periodo), elija el tiempo durante el cual desea que Secrets Manager rote su secreto, por ejemplo, **3h**, para un periodo de tres horas. El periodo no debe prolongarse hasta el siguiente periodo de rotación. Si no se especifica Window duration (Duración del periodo) para una programación de rotación en horas, el periodo concluye automáticamente al cabo de una hora. Para una programación de rotación en días, el periodo concluye automáticamente al final del día.

d. (Opcional) Elija Rotate immediately when the secret is stored (Rotar inmediatamente cuando se almacene el secreto) a fin de rotar su secreto en cuanto guarde los cambios. Si desmarca la casilla de verificación, la primera rotación comenzará conforme a la programación establecida.

e. En Rotation function (Función de rotación), elija Create function (Crear función). La consola de Lambda se abre en una nueva ventana.

   - En la consola de Lambda, en la página Create function (Crear una función), realice una de las siguientes acciones:
     - Si aparece la opción Browse serverless app repository (Examinar el repositorio de aplicaciones sin servidor), selecciónela.

       A. En Aplicaciones públicas, en el cuadro de búsqueda, ingresa SecretsManagerRotationTemplate.

       B. Elija Show apps that create custom IAM roles or resource policies (Mostrar aplicaciones que crean roles de IAM personalizados o políticas de recursos).

       C. Elige el SecretsManagerRotationTemplatemosaico.

       D. En la página Review, configure and deploy (Revisar, configurar e implementar), en el mosaico Application settings (Configuración de la aplicación), complete los campos obligatorios y, a continuación, elija Deploy (Implementar). Para obtener una lista de puntos de enlace , consulte the section called “Puntos de conexión de Secrets Manager”.

- Si no ve Browse serverless app repository (Examinar el repositorio de aplicaciones sin servidor), es posible que su Región de AWS no admita el AWS Serverless Application Repository. Elija Crear desde cero.

  A. En Function name (Nombre de la función), ingrese un nombre para la función de rotación.

  B. En Runtime (Tiempo de ejecución), elija Python 3.9.

  C. Cuando se abra la nueva función de Lambda, desplácese hacia abajo para elegir Configuration (Configuración) y, a continuación, a la izquierda, seleccione Permissions (Permisos).

  D. Desplácese hacia abajo hasta Resource-based policy (Política basada en recursos) y seleccione Add permissions (Agregar permisos) a fin de conceder permiso a Secrets Manager para invocar la función. Para adjuntar una política de recursos a una función de Lambda, consulte Uso de políticas basadas en recursos para Lambda.

  En la siguiente política se muestra cómo permitir a Secrets Manager invocar la función de Lambda.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "secretsmanager.amazonaws.com"
            },
        "Action": "lambda:InvokeFunction",
        "Resource": "LambdaRotationFunctionARN"
    }
    ]
}
```

f. Vuelva a la consola de Secrets Manager para adjuntar la nueva función de rotación a su secreto.

g. En Lambda rotation function (Función de rotación de Lambda), elija el botón de actualización. A continuación, en la lista de funciones, elija la nueva función.

h. Elija Save (Guardar).

Paso 2: establecer permisos para la función de rotación

La función de rotación de Lambda necesita permiso para acceder al secreto en Secrets Manager y también necesita permiso para acceder a su base de datos o servicio. En este paso, concederá estos permisos al rol de ejecución de Lambda. Si el secreto está cifrado con una clave KMS distinta de la Clave administrada de AWS `aws/secretsmanager`, tiene que conceder permiso al rol de ejecución de Lambda para utilizar la clave. Puede utilizar el contexto de cifrado SecretARN para limitar el uso de la función de descifrado, de modo que el rol de la función de rotación solo tenga acceso para descifrar el secreto que es responsable de rotar. Para ver ejemplos de políticas, consulte Permisos para rotación.

Consulte las instrucciones en Rol de ejecución de Lambda, en la Guía para desarrolladores de AWS Lambda.

Paso 3: (Opcional) establecer una condición de permiso adicional en la función de rotación

En la política de recursos de la función de rotación, se recomienda incluir la clave de contexto `aws:SourceAccount` para poder evitar que Lambda se utilice como suplente confuso. En el caso de algunos servicios de AWS, para evitar un escenario de suplente confuso, AWS recomienda que se utilicen las claves de condición globales `aws:SourceArn` y `aws:SourceAccount`. No obstante, si se incluye la condición `aws:SourceArn` en la política de la función de rotación, la función de rotación solo se puede utilizar para rotar el secreto especificado por ese ARN. Se recomienda incluir solo la clave de contexto `aws:SourceAccount`, para poder utilizar la función de rotación para varios secretos.

Actualizar la política de recursos de la función de rotación

1. En la consola de Secrets Manager, elija el secreto y, a continuación, en la página de detalles, en Rotation configuration (Configuración de la rotación), elija la función de rotación de Lambda. Se abre la consola de Lambda.
2. Siga las instrucciones que se describen en Uso de políticas basadas en recursos para Lambda para agregar una condición `aws:sourceAccount`.

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
```

```
},
```

Paso 4: configurar el acceso a la red para la función de rotación

Para poder rotar un secreto, la función de rotación de Lambda debe poder acceder al secreto. Si su secreto contiene credenciales, la función de Lambda también debe poder acceder al origen de esas credenciales, como una base de datos o un servicio.

Acceder a un secreto

La función de rotación de Lambda debe ser capaz de acceder a un punto de enlace de Secrets Manager. Si la función de Lambda puede acceder a Internet, puede utilizar un punto de enlace público. Para buscar un punto de conexión, consulte the section called “Puntos de conexión de Secrets Manager”.

Si la función de Lambda se ejecuta en una VPC que no tiene acceso a Internet, recomendamos configurar los puntos de enlace privados del servicio de Secrets Manager dentro de la VPC. La VPC puede interceptar entonces las solicitudes dirigidas al punto de enlace regional público y redirigirlas al punto de enlace privado. Para obtener más información, consulte Punto de conexión VPC.

También puede habilitar la función de Lambda para acceder a un punto de conexión público de Secrets Manager. Para ello, agregue una puerta de enlace NAT o una puerta de enlace de Internet a su VPC. Esto permite que el tráfico de la VPC alcance el punto de conexión público. Esto expone a la VPC a más riesgo, ya que desde la red pública de Internet se puede atacar la dirección IP de la gateway.

(Opcional) Acceder a la base de datos o al servicio

En el caso de los secretos, como las claves de API, no hay ninguna base de datos o servicio de origen que deba actualizar junto con el secreto.

Si la base de datos o el servicio se ejecutan en una instancia de Amazon EC2 en una VPC, es recomendable configurar la función de Lambda para que se ejecute en la misma VPC. A continuación, la función de rotación puede comunicarse directamente con el servicio. Para obtener más información, consulte Configuración del acceso a la VPC.

Para permitir que la función de Lambda tenga acceso a la base de datos o el servicio, debe asegurarse de que los grupos de seguridad adjuntos a la función de rotación de Lambda permitan

conexiones salientes a la base de datos o el servicio. Asimismo, debe asegurarse de que los grupos de seguridad adjuntos a la base de datos o el servicio permitan conexiones entrantes desde la función de rotación de Lambda.

Paso 5: escribir el código de la función de rotación

La función de rotación que creó en el paso 1 es el punto de partida de la función. Tiene que escribir el código para su caso de uso específico. Para una función que pueda rotar un ElastiCache secreto de Amazon, puedes copiar el código de la plantilla correspondiente proporcionada por Secrets Manager.

Al escribir la función, tenga cuidado con la inclusión de instrucciones de depuración o registro. Estas declaraciones pueden provocar que la información de tu función se escriba en Amazon CloudWatch, por lo que debes asegurarte de que el registro no incluya ninguna información confidencial recopilada durante el desarrollo.

Por motivos de seguridad, Secrets Manager solo permite que una función de rotación de Lambda rote el secreto de forma directa. La función de rotación no puede llamar a una segunda función de Lambda para rotar el secreto.

Para ver ejemplos de instrucciones de registro, consulte el código fuente de the section called “Plantillas de función de rotación”.

Si utilizas bibliotecas y archivos binarios externos, por ejemplo, para conectarte a un recurso, tendrás que gestionar su aplicación de parches y su conservación. up-to-date

Puede consultar sugerencias de depuración en Testing and debugging serverless applications (Prueba y depuración de aplicaciones sin servidor).

Abrir la función de rotación de Lambda para editarla

1. En la consola de Secrets Manager, elija el secreto.
2. En la sección Rotation configuration (Configuración de la rotación), en Lambda rotation function (Función de rotación de Lambda), elija su función de rotación.

   Se abre la consola de Lambda.

   - Para cambiar el código de la función, desplácese hacia abajo hasta la sección Código fuente.

- Para MySQL 5.7 y versiones posteriores, para alternar la rotación de usuarios, para cambiar la longitud máxima del nombre de usuario, en Variables de entorno, cambie `USERNAME_CHARACTER_LIMIT`.

Si tu función aún no lo tiene, copia el código del. SecretsManagerRotationTemplate

Hay cuatro pasos para rotar un secreto, que se corresponden con los cuatro métodos de una función de rotación de Lambda que se incluyen a continuación.

Métodos

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

En `create_secret`, primero compruebe si existe un secreto con una llamada a `get_secret_value` con el valor transmitido de `ClientRequestToken`. Si no hay ningún secreto, cree uno nuevo con `create_secret` y el token como `VersionId`. A continuación, puede generar un nuevo valor de secreto con `get_random_password`. Debe asegurarse de que el nuevo valor de secreto solo incluya caracteres válidos para la base de datos o el servicio. Excluya caracteres con el parámetro `ExcludeCharacters`. Llame a `put_secret_value` para almacenarlo con la etiqueta provisional `AWSPENDING`. Almacenar el nuevo valor de secreto en `AWSPENDING` ayuda a garantizar la idempotencia. Si se produce un error en la rotación por cualquier motivo, puede hacer referencia a ese valor de secreto en llamadas posteriores. Consulte How do I make my Lambda function idempotent (¿Cómo puedo hacer que mi función de Lambda sea idempotente?).

Al probar la función, utilice AWS CLI para ver las fases de la versión: llame a `describe-secret` y consulte `VersionIdsToStages`.

`set_secret`

En `set_secret`, cambie la credencial en la base de datos o el servicio para que coincidan con el nuevo valor de secreto en la versión de `AWSPENDING` del secreto.

Si se transmiten instrucciones a un servicio que las interpreta, como una base de datos, utilice la parametrización de consultas. Para obtener más información, consulte la Query Parameterization Cheat Sheet (Hoja de referencia de parametrización de consultas) en el sitio web de OWASP.

La función de rotación es un suplente privilegiado que tiene autorización para acceder a las credenciales del cliente y modificarlas tanto en el secreto de Secrets Manager como en el recurso de destino. Para evitar un posible ataque de falsificación por solicitud, debe asegurarse de que ningún atacante pueda usar la función para acceder a otros recursos. Antes de actualizar la credencial, haga lo siguiente:

- Compruebe que la credencial de la versión de `AWSCURRENT` del secreto sea válida. Si la credencial de `AWSCURRENT` no es válida, deje de intentar la rotación.
- Compruebe que los valores de secreto de `AWSCURRENT` y `AWSPENDING` sean para el mismo recurso. En el caso de un nombre de usuario y una contraseña, compruebe que los nombres de usuario de `AWSCURRENT` y `AWSPENDING` sean los mismos.
- Compruebe que el recurso del servicio de destino sea el mismo. En el caso de una base de datos, compruebe que los nombres de host de `AWSCURRENT` y `AWSPENDING` sean los mismos.

`test_secret`

En `test_secret`, pruebe la versión de `AWSPENDING` del secreto; para ello, utilícelo para acceder a la base de datos o el servicio.

`finish_secret`

En `finish_secret`, utilice `update_secret_version_stage` para mover la etiqueta provisional `AWSCURRENT` de la versión anterior del secreto a la nueva. Secrets Manager agrega automáticamente la etiqueta provisional `AWSPREVIOUS` a la versión anterior, para que retenga la última versión buena conocida del secreto.

Siguientes pasos

Consulte the section called “Solución de problemas de rotación de ”.

Configuración de la rotación automática para los secretos de AWS Secrets Manager mediante AWS CLI

La rotación es el proceso de actualización periódica de un secreto. Cuando se rota un secreto, se actualizan las credenciales tanto en el secreto como en la base de datos o el servicio para el que está destinado el secreto.

Secrets Manager utiliza funciones de Lambda para rotar los secretos. Para obtener una descripción general, consulte the section called “Cómo funciona la rotación”.

También puede usar la consola para configurar la rotación. Para obtener más información, consulte Rotación automática (consola).

Para configurar la rotación mediante AWS CLI, si va a rotar un secreto de Amazon RDS, Amazon Redshift o Amazon DocumentDB, primero debe elegir una the section called “Estrategia de rotación”. Si elige la estrategia de usuarios alternativos, debe almacenar un secreto independiente con las credenciales de un superusuario de base de datos. A continuación, escriba el código de la función de rotación. Secrets Manager proporciona plantillas en las que puede basar su función. A continuación, cree una función de Lambda con el código y establezca los permisos tanto para la función de Lambda como para el rol de ejecución de Lambda. El siguiente paso consiste en asegurarse de que la función de rotación de Lambda pueda acceder a Secrets Manager y a la base de datos o el servicio a través de la red. Por último, configure el secreto para la rotación.

Para activar la rotación automática, debe tener permisos para crear el rol de ejecución de IAM y adjuntarle una política de permisos. Necesita ambos permisos, `iam:CreateRole` y `iam:AttachRolePolicy`.

> **Warning**
>
> Conceder a una identidad los permisos `iam:CreateRole` y `iam:AttachRolePolicy` permite que la identidad se conceda a sí misma cualquier permiso.

Pasos:

- (Opcional) Paso 1: crear un secreto de superusuario
- Paso 2: escribir el código de la función de rotación
- Paso 3: crear el rol de ejecución y la función de Lambda
- Paso 4: configurar el acceso a la red

- Paso 5: configurar el secreto para la rotación
- Siguientes pasos

(Opcional) Paso 1: crear un secreto de superusuario

En el caso de Amazon RDS, Amazon Redshift y Amazon DocumentDB, Secrets Manager ofrece dos estrategias de rotación:

Estrategia de rotación de un solo usuario

Esta estrategia actualiza las credenciales de un usuario en un secreto. En el caso de las instancias Db2 de Amazon RDS, dado que los usuarios no pueden cambiar sus propias contraseñas, debe proporcionar las credenciales de administrador en un secreto independiente. Esta es la estrategia de rotación más sencilla y es adecuada para la mayoría de los casos de uso. En particular, recomendamos que utilice esta estrategia para las credenciales de los usuarios interactivos o únicos (ad hoc).

Cuando el secreto rota, las conexiones de bases de datos abiertas no se eliminan. Mientras se produce la rotación, hay un breve periodo de tiempo entre el momento en que cambia la contraseña de la base de datos y el momento en que se actualiza el secreto. Durante este tiempo, existe un riesgo bajo de que la base de datos deniegue las llamadas que utilizan las credenciales rotadas. Puede mitigar este riesgo con una estrategia de reintentos apropiada. Tras la rotación, las nuevas conexiones utilizan las nuevas credenciales.

Estrategia de rotación de usuarios alternativos

Esta estrategia actualiza las credenciales de dos usuarios en un secreto. Se crea el primer usuario y, durante la primera rotación, la función de rotación lo clona para crear el segundo usuario. Cada vez que el secreto rota, la función de rotación alterna la contraseña de usuario que actualiza. Dado que la mayoría de los usuarios no tienen permiso para clonarse a sí mismos, debe proporcionar las credenciales de un usuario de tipo `superuser` en otro secreto. Recomendamos que utilice la estrategia de rotación de un solo usuario cuando los usuarios clonados en su base de datos no tienen los mismos permisos que el usuario original y para las credenciales de los usuarios interactivos o únicos (ad hoc).

Esa estrategia es adecuada para bases de datos con modelos de permisos en los que un rol es propietario de las tablas de base de datos y un segundo rol tiene permiso para acceder a las tablas de base de datos. También es adecuada para aplicaciones que requieren alta disponibilidad. Si una aplicación recupera el secreto durante la rotación, seguirá obteniendo un

conjunto de credenciales válido. Tras la rotación, las credenciales de `user` y `user_clone` son válidas. Incluso hay menos posibilidades de que las aplicaciones sufran denegaciones durante este tipo de rotación que con la rotación de un solo usuario. Si la base de datos está alojada en una granja de servidores donde el cambio de contraseña tarda tiempo en propagarse a todos los servidores, existe el riesgo de que la base de datos deniegue las llamadas que utilicen las nuevas credenciales. Puede mitigar este riesgo con una estrategia de reintentos apropiada.

Secrets Manager crea el usuario clonado con los mismos permisos que el usuario original. Si cambia los permisos del usuario original después de crear el clon, también debe cambiar los permisos del usuario clonado.

> **Important**
>
> Si elige la estrategia de usuarios alternativos, debe Creación de un secreto de base de datos y almacenar en él las credenciales de superusuario de la base de datos. Necesita un secreto con credenciales de superusuario porque la rotación clona el primer usuario y la mayoría de los usuarios no tienen ese permiso.

Paso 2: escribir el código de la función de rotación

Para rotar un secreto, se necesita una función de rotación. Una función de rotación es una función de Lambda a la que Secrets Manager llama para rotar un secreto.

Para una función que pueda rotar un secreto de Amazon RDS, Amazon Aurora, Amazon Redshift, Amazon DocumentDB o ElastiCache Amazon, puede copiar el código de la plantilla correspondiente proporcionada por Secrets Manager.

En el caso de todos los demás tipos de secretos, utilice la plantilla de rotación genérica como punto de partida para escribir su propia función de rotación.

Guarde la función de rotación en un archivo ZIP llamado *`my-function.zip`* junto con las dependencias necesarias.

Al escribir la función, tenga cuidado con la inclusión de instrucciones de depuración o registro. Estas declaraciones pueden provocar que la información de tu función se escriba en Amazon CloudWatch, por lo que debes asegurarte de que el registro no incluya ninguna información confidencial recopilada durante el desarrollo.

Por motivos de seguridad, Secrets Manager solo permite que una función de rotación de Lambda rote el secreto de forma directa. La función de rotación no puede llamar a una segunda función de Lambda para rotar el secreto.

Para ver ejemplos de instrucciones de registro, consulte el código fuente de the section called “Plantillas de función de rotación”.

Si utilizas bibliotecas y archivos binarios externos, por ejemplo, para conectarte a un recurso, tendrás que gestionar su aplicación de parches y su conservación. up-to-date

Puede consultar sugerencias de depuración en Testing and debugging serverless applications (Prueba y depuración de aplicaciones sin servidor).

Abrir la función de rotación de Lambda para editarla

1. En la consola de Secrets Manager, elija el secreto.
2. En la sección Rotation configuration (Configuración de la rotación), en Lambda rotation function (Función de rotación de Lambda), elija su función de rotación.

   Se abre la consola de Lambda.

   - Para cambiar el código de la función, desplácese hacia abajo hasta la sección Código fuente.
   - Para MySQL 5.7 y versiones posteriores, para alternar la rotación de usuarios, para cambiar la longitud máxima del nombre de usuario, en Variables de entorno, cambie `USERNAME_CHARACTER_LIMIT`.

Si tu función aún no lo tiene, copia el código del. SecretsManagerRotationTemplate

Hay cuatro pasos para rotar un secreto, que se corresponden con los cuatro métodos de una función de rotación de Lambda que se incluyen a continuación.

Métodos

- create_secret
- set_secret
- test_secret
- finish_secret

create_secret

En `create_secret`, primero compruebe si existe un secreto con una llamada a `get_secret_value` con el valor transmitido de `ClientRequestToken`. Si no hay ningún secreto, cree uno nuevo con `create_secret` y el token como `VersionId`. A continuación, puede generar un nuevo valor de secreto con `get_random_password`. Debe asegurarse de que el nuevo valor de secreto solo incluya caracteres válidos para la base de datos o el servicio. Excluya caracteres con el parámetro `ExcludeCharacters`. Llame a `put_secret_value` para almacenarlo con la etiqueta provisional `AWSPENDING`. Almacenar el nuevo valor de secreto en `AWSPENDING` ayuda a garantizar la idempotencia. Si se produce un error en la rotación por cualquier motivo, puede hacer referencia a ese valor de secreto en llamadas posteriores. Consulte How do I make my Lambda function idempotent (¿Cómo puedo hacer que mi función de Lambda sea idempotente?).

Al probar la función, utilice AWS CLI para ver las fases de la versión: llame a `describe-secret` y consulte `VersionIdsToStages`.

set_secret

En `set_secret`, cambie la credencial en la base de datos o el servicio para que coincidan con el nuevo valor de secreto en la versión de `AWSPENDING` del secreto.

Si se transmiten instrucciones a un servicio que las interpreta, como una base de datos, utilice la parametrización de consultas. Para obtener más información, consulte la Query Parameterization Cheat Sheet (Hoja de referencia de parametrización de consultas) en el sitio web de OWASP.

La función de rotación es un suplente privilegiado que tiene autorización para acceder a las credenciales del cliente y modificarlas tanto en el secreto de Secrets Manager como en el recurso de destino. Para evitar un posible ataque de falsificación por solicitud, debe asegurarse de que ningún atacante pueda usar la función para acceder a otros recursos. Antes de actualizar la credencial, haga lo siguiente:

- Compruebe que la credencial de la versión de `AWSCURRENT` del secreto sea válida. Si la credencial de `AWSCURRENT` no es válida, deje de intentar la rotación.
- Compruebe que los valores de secreto de `AWSCURRENT` y `AWSPENDING` sean para el mismo recurso. En el caso de un nombre de usuario y una contraseña, compruebe que los nombres de usuario de `AWSCURRENT` y `AWSPENDING` sean los mismos.
- Compruebe que el recurso del servicio de destino sea el mismo. En el caso de una base de datos, compruebe que los nombres de host de `AWSCURRENT` y `AWSPENDING` sean los mismos.

`test_secret`

En `test_secret`, pruebe la versión de `AWSPENDING` del secreto; para ello, utilícelo para acceder a la base de datos o el servicio.

`finish_secret`

En `finish_secret`, utilice `update_secret_version_stage` para mover la etiqueta provisional `AWSCURRENT` de la versión anterior del secreto a la nueva. Secrets Manager agrega automáticamente la etiqueta provisional `AWSPREVIOUS` a la versión anterior, para que retenga la última versión buena conocida del secreto.

Paso 3: crear el rol de ejecución y la función de Lambda

Un rol de ejecución de Lambda es un rol que Lambda asume cuando se invoca la función.

Crear un rol de ejecución y una función de rotación de Lambda

1. Cree una política de confianza para el rol de ejecución de Lambda y guárdela como un archivo JSON. Para ver ejemplos, consulte Permisos para rotación. La política debe:
   - Permitir que el rol llame a las operaciones de Secrets Manager relacionadas con el secreto.
   - Permitir que el rol utilice la clave de KMS si el secreto está cifrado con una clave distinta de `aws/secretsmanager`.
   - Permitir que el rol llame al servicio para el que está destinado el secreto.
2. Cree el rol de ejecución de Lambda y aplique la política de confianza mediante una llamada a `iam create-role`.

   ```
   aws iam create-role \
       --role-name rotation-lambda-role \
       --assume-role-policy-document file://trust-policy.json
   ```

3. (Opcional) Para un secreto que contenga credenciales de Amazon RDS, si utiliza la estrategia de usuarios alternativos y Amazon RDS administra el secreto del superusuario, entonces debe permitir que la función de rotación llame a las API de solo lectura de Amazon RDS para obtener la información de conexión de la base de datos. Para ello, asocie la política AWS gestionada AmazonRDS ReadOnlyAccess a la función de ejecución de la función Lambda mediante una llamada. `iam attach-role-policy`

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/AmazonRDSReadOnlyAccess \
    --role-name rotation-lambda-role
```

4. Cree la función de Lambda a partir del archivo ZIP mediante una llamada a `lambda create-function`.

```
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. Establezca una política de recursos en la función de Lambda para permitir que Secrets Manager la invoque mediante una llamada a `lambda add-permission`. El comando de ejemplo incluye `source-account` para ayudar a evitar que Lambda se utilice para un ataque de falsificación por solicitud.

```
aws lambda add-permission \
    --function-name my-rotation-function \
    --action lambda:InvokeFunction \
    --statement-id SecretsManager \
    --principal secretsmanager.amazonaws.com \
    --source-account 123456789012
```

Paso 4: configurar el acceso a la red

Para poder rotar un secreto, la función de rotación de Lambda debe poder acceder tanto al secreto como a la base de datos o el servicio.

Acceder a un secreto

La función de rotación de Lambda debe ser capaz de acceder a un punto de enlace de Secrets Manager. Si la función de Lambda puede acceder a Internet, puede utilizar un punto de enlace público. Para buscar un punto de conexión, consulte the section called “Puntos de conexión de Secrets Manager”.

Si la función de Lambda se ejecuta en una VPC que no tiene acceso a Internet, recomendamos configurar los puntos de enlace privados del servicio de Secrets Manager dentro de la VPC. La VPC puede interceptar entonces las solicitudes dirigidas al punto de enlace regional público y redirigirlas al punto de enlace privado. Para obtener más información, consulte Punto de conexión VPC.

También puede habilitar la función de Lambda para acceder a un punto de conexión público de Secrets Manager. Para ello, agregue una puerta de enlace NAT o una puerta de enlace de Internet a su VPC. Esto permite que el tráfico de la VPC alcance el punto de conexión público. Esto expone a la VPC a más riesgo, ya que desde la red pública de Internet se puede atacar la dirección IP de la gateway.

Acceder a la base de datos o al servicio

Si la base de datos o el servicio se ejecutan en una instancia de Amazon EC2 en una VPC, es recomendable configurar la función de Lambda para que se ejecute en la misma VPC. A continuación, la función de rotación puede comunicarse directamente con el servicio. Para obtener más información, consulte Configuración del acceso a la VPC.

Para permitir que la función de Lambda tenga acceso a la base de datos o el servicio, debe asegurarse de que los grupos de seguridad adjuntos a la función de rotación de Lambda permitan conexiones salientes a la base de datos o el servicio. Asimismo, debe asegurarse de que los grupos de seguridad adjuntos a la base de datos o el servicio permitan conexiones entrantes desde la función de rotación de Lambda.

Para la rotación alternativa de los usuarios donde otro AWS servicio es el que gestiona el superusuario, Lambda debe poder llamar al punto de conexión del servicio de Amazon RDS para obtener la información de conexión de la base de datos. Recomendamos configurar un punto de conexión de VPC para el servicio de base de datos. Para obtener más información, consulte:

- Puntos de conexión de VPC de la API y la interfaz de Amazon RDS en la Guía de usuario de Amazon RDS.
- Cómo trabajar con puntos de conexión de VPC en la Guía de administración de Amazon Redshift.

Paso 5: configurar el secreto para la rotación

Para activar la rotación automática de su secreto, llame a `rotate-secret`. Puede establecer una programación de rotación con una expresión de programación `cron()` o `rate()` y definir una

duración del periodo de rotación. Se puede rotar un secreto con una frecuencia máxima de cuatro horas. Para obtener más información, consulte Programación de expresiones.

```
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\"}"
```

Siguientes pasos

Consulte the section called “Solución de problemas de rotación de ”.

Rotar un secreto de AWS Secrets Manager inmediatamente

Solo se puede rotar un secreto cuya rotación se haya configurado previamente. Para determinar si se ha configurado un secreto para la rotación, en la consola, consulte el secreto y desplácese hacia abajo hasta la sección Rotation configuration (Configuración de rotación). Si el valor de Rotation status (Estado de rotación) es Enabled (Habilitada), el secreto está configurado para la rotación. También puede llamar a `describe-secret` en AWS CLI. Si la respuesta tiene un valor de `RotationLambdaARN` y `RotationRules`, el secreto está configurado para la rotación. De lo contrario, puede configurar la rotación automática:

- Rotación automática de secretos de bases de datos (consola)
- Rotación automática (consola)
- Rotación automática (AWS CLI)

Para rotar un secreto inmediatamente (consola)

1. Abra la consola de Secrets Manager enhttps://console.aws.amazon.com/secretsmanager/.
2. Elija el secreto.
3. En la página de detalles del secreto, en Rotation configuration (Configuración de rotación), elija Rotate secret immediately (Rotar secreto inmediatamente).
4. En el cuadro de diálogo Rotate secret (Rotar secreto), seleccione Rotate (Rotar).

AWS CLI

Example Rotar un secreto inmediatamente

En el siguiente ejemplo de `rotate-secret` se inicia una rotación inmediata. El resultado muestra el valor de VersionId de la nueva versión de secreto creada por la rotación. El secreto ya debe tener configurada la rotación.

```
aws secretsmanager rotate-secret \
    --secret-id MyTestSecret
```

AWS Secrets Manager plantillas de funciones de rotación

Secrets Manager proporciona plantillas de función de rotación para:

- Amazon RDS y Amazon Aurora
- Amazon DocumentDB (con compatibilidad con MongoDB)
- Amazon Redshift
- Amazon ElastiCache
- Otros tipos de secretos

Para utilizar las plantillas, consulte lo siguiente:

- Rotación de credenciales para Amazon RDS,Amazon Aurora Amazon Redshift y Amazon DocumentDB
- Otros tipos de credenciales (instrucciones de la consola)
- Otros tipos de credenciales (AWS CLI instrucciones)

Las plantillas son compatibles con Python 3.9.

Para escribir su propia función de rotación, consulte Escribir una función de rotación.

Amazon RDS y Amazon Aurora

Temas

- Amazon RDS Db2 para un solo usuario
- Usuarios alternos de Amazon RDS Db2

- Un solo usuario de MariaDB en Amazon RDS
- Usuarios alternativos de MariaDB en Amazon RDS
- Amazon RDS y Amazon Aurora MySQL para un solo usuario
- Usuarios alternos de Amazon RDS y Amazon Aurora MySQL
- Un solo usuario de Oracle en Amazon RDS
- Usuarios alternativos de Oracle en Amazon RDS
- Amazon RDS y Amazon Aurora PostgreSQL para un solo usuario
- Usuarios alternos de Amazon RDS y Amazon Aurora PostgreSQL
- Un solo usuario de Microsoft SQL Server en Amazon RDS
- Usuarios alternativos de Microsoft SQL Server en Amazon RDS

Amazon RDS Db2 para un solo usuario

- Nombre de plantilla: SecretsManager RDSdb2 RotationSingleUser
- Estrategia de rotación: Estrategia de rotación: un solo usuario.
- Estructura de **`SecretString`**: the section called “Estructura del secreto de Amazon RDS Db2”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ rdsDB2 /lambda_function.py SecretsManager RotationSingleUser
- Dependencia: python-ibmdb

Usuarios alternos de Amazon RDS Db2

- Nombre de SecretsManager plantilla: RDSdb2 RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`**: the section called “Estructura del secreto de Amazon RDS Db2”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ rdsDB2 /lambda_function.py SecretsManager RotationMultiUser
- Dependencia: python-ibmdb

Un solo usuario de MariaDB en Amazon RDS

- Nombre de SecretsManager plantilla: RDSMariaDB RotationSingleUser

- Estrategia de rotación: Estrategia de rotación: un solo usuario.
- Estructura de **`SecretString`**: the section called “Estructura del secreto de MariaDB en Amazon RDS”.
- Código fuente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSMariaDB / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationSingleUser
- DependenciaPyMy: SQL 1.0.2

Usuarios alternativos de MariaDB en Amazon RDS

- Nombre de plantilla: RDSMariaDB SecretsManager RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`**: the section called “Estructura del secreto de MariaDB en Amazon RDS”.
- Código fuente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSMariaDB / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationMultiUser
- DependenciaPyMy: SQL 1.0.2

Amazon RDS y Amazon Aurora MySQL para un solo usuario

- Nombre de plantilla: RDSmySQL SecretsManager RotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Estructura secreta de Amazon RDS y Amazon Aurora MySQL”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ RDSmySQL /lambda_function.py SecretsManager RotationSingleUser
- DependenciaPyMy: SQL 1.0.2

Usuarios alternos de Amazon RDS y Amazon Aurora MySQL

- Nombre de plantilla: RDSmySQL SecretsManager RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Estructura secreta de Amazon RDS y Amazon Aurora MySQL”.

- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ RDSmySQL /lambda_function.py SecretsManager RotationMultiUser
- DependenciaPyMy: SQL 1.0.2

Un solo usuario de Oracle en Amazon RDS

- Nombre de plantilla: RDS SecretsManager OracleRotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Oracle en Amazon RDS”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDS /lambda_function.py OracleRotationSingleUser
- Dependencia: python-oracledb 2.0.1

Usuarios alternativos de Oracle en Amazon RDS

- Nombre de la plantilla: RDS SecretsManager OracleRotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Oracle en Amazon RDS”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDS /lambda_function.py OracleRotationMultiUser
- Dependencia: python-oracledb 2.0.1

Amazon RDS y Amazon Aurora PostgreSQL para un solo usuario

- Nombre de plantilla: RDSPostgreSQL SecretsManager RotationSingleUser
- Estrategia de rotación: Estrategia de rotación: un solo usuario.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Amazon RDS y Amazon Aurora PostgreSQL”.
- Código fuente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSPostgreSQL / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationSingleUser
- DependenciaPyGre: SQL 5.0.7

Usuarios alternos de Amazon RDS y Amazon Aurora PostgreSQL

- Nombre de plantilla: RDSPostgreSQL SecretsManager RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Amazon RDS y Amazon Aurora PostgreSQL”.
- Código fuente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSPostgreSQL / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationMultiUser
- DependenciaPyGre: SQL 5.0.7

Un solo usuario de Microsoft SQL Server en Amazon RDS

- Nombre de plantilla: RDSSQL SecretsManager ServerRotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Microsoft SQL Server en Amazon RDS”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDSSQL /lambda_function.py ServerRotationSingleUser
- Dependencia: Pymssql 2.2.2

Usuarios alternativos de Microsoft SQL Server en Amazon RDS

- Nombre de SecretsManager plantilla: RDSSQL ServerRotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Microsoft SQL Server en Amazon RDS”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDSSQL /lambda_function.py ServerRotationMultiUser
- Dependencia: Pymssql 2.2.2

Amazon DocumentDB (con compatibilidad con MongoDB)

Usuario único de Amazon DocumentDB

- Nombre SecretsManagerMongo de la plantilla: DB RotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Amazon DocumentDB”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerMongo -lambdas/tree/master/ DB /lambda_function.py RotationSingleUser
- Dependencia: Pymongo 3.2

Usuarios alternativos de Amazon DocumentDB

- Nombre SecretsManagerMongo de la plantilla: DB RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Estructura del secreto de Amazon DocumentDB”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerMongo -lambdas/tree/master/ DB /lambda_function.py RotationMultiUser
- Dependencia: Pymongo 3.2

Amazon Redshift

Usuario único de Amazon Redshift

- Nombre de la plantilla: SecretsManagerRedshiftRotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- **`SecretString`**Estructura esperada: the section called “Estructura del secreto de Amazon Redshift” othe section called “Estructura secreta de Amazon Redshift Serverless”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerRedshiftRotationSingleUser -lambdas/tree/master/ /lambda_function.py
- DependenciaPyGre: SQL 5.0.7

Usuarios alternativos de Amazon Redshift

- Nombre de la plantilla: SecretsManagerRedshiftRotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- **`SecretString`**Estructura esperada: the section called “Estructura del secreto de Amazon Redshift” othe section called “Estructura secreta de Amazon Redshift Serverless”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerRedshiftRotationMultiUser -lambdas/tree/master/ /lambda_function.py
- DependenciaPyGre: SQL 5.0.7

Amazon ElastiCache

Para usar esta plantilla, consulta Rotación automática de contraseñas para los usuarios en la Guía del ElastiCache usuario de Amazon.

- Nombre de la plantilla: SecretsManagerElasticacheUserRotation
- Estructura de **`SecretString`** esperada: the section called “Estructura ElastiCache secreta de Amazon”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerElasticacheUserRotation -lambdas/tree/master/ /lambda_function.py

Otros tipos de secretos

Secrets Manager proporciona esta plantilla como punto de partida para que pueda crear una función de rotación para cualquier tipo de secreto.

- Nombre de la plantilla: SecretsManagerRotationTemplate
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerRotationTemplate -lambdas/tree/master/ /lambda_function.py

Al escribir la función, tenga cuidado con la inclusión de instrucciones de depuración o registro. Estas declaraciones pueden provocar que la información de tu función se escriba en Amazon CloudWatch, por lo que debes asegurarte de que el registro no incluya ninguna información confidencial recopilada durante el desarrollo.

Por motivos de seguridad, Secrets Manager solo permite que una función de rotación de Lambda rote el secreto de forma directa. La función de rotación no puede llamar a una segunda función de Lambda para rotar el secreto.

Para ver ejemplos de instrucciones de registro, consulte el código fuente de the section called "Plantillas de función de rotación".

Si utilizas bibliotecas y archivos binarios externos, por ejemplo, para conectarte a un recurso, tendrás que gestionar su aplicación de parches y su conservación. up-to-date

Puede consultar sugerencias de depuración en Testing and debugging serverless applications (Prueba y depuración de aplicaciones sin servidor).

Hay cuatro pasos para rotar un secreto, que se corresponden con los cuatro métodos de una función de rotación de Lambda que se incluyen a continuación.

Métodos

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

En `create_secret`, primero compruebe si existe un secreto con una llamada a `get_secret_value` con el valor transmitido de `ClientRequestToken`. Si no hay ningún secreto, cree uno nuevo con `create_secret` y el token como `VersionId`. A continuación, puede generar un nuevo valor de secreto con `get_random_password`. Debe asegurarse de que el nuevo valor de secreto solo incluya caracteres válidos para la base de datos o el servicio. Excluya caracteres con el parámetro `ExcludeCharacters`. Llame a `put_secret_value` para almacenarlo con la etiqueta provisional `AWSPENDING`. Almacenar el nuevo valor de secreto en `AWSPENDING` ayuda a garantizar la idempotencia. Si se produce un error en la rotación por cualquier motivo, puede hacer referencia a ese valor de secreto en llamadas posteriores. Consulte How do I make my Lambda function idempotent (¿Cómo puedo hacer que mi función de Lambda sea idempotente?).

A medida que pruebes la función, usa AWS CLI para ver las etapas de la versión: llama `describe-secret` y mira. `VersionIdsToStages`

set_secret

En `set_secret`, cambie la credencial en la base de datos o el servicio para que coincidan con el nuevo valor de secreto en la versión de `AWSPENDING` del secreto.

Si se transmiten instrucciones a un servicio que las interpreta, como una base de datos, utilice la parametrización de consultas. Para obtener más información, consulte la Query Parameterization Cheat Sheet (Hoja de referencia de parametrización de consultas) en el sitio web de OWASP.

La función de rotación es un suplente privilegiado que tiene autorización para acceder a las credenciales del cliente y modificarlas tanto en el secreto de Secrets Manager como en el recurso de destino. Para evitar un posible ataque de falsificación por solicitud, debe asegurarse de que ningún atacante pueda usar la función para acceder a otros recursos. Antes de actualizar la credencial, haga lo siguiente:

- Compruebe que la credencial de la versión de `AWSCURRENT` del secreto sea válida. Si la credencial de `AWSCURRENT` no es válida, deje de intentar la rotación.
- Compruebe que los valores de secreto de `AWSCURRENT` y `AWSPENDING` sean para el mismo recurso. En el caso de un nombre de usuario y una contraseña, compruebe que los nombres de usuario de `AWSCURRENT` y `AWSPENDING` sean los mismos.
- Compruebe que el recurso del servicio de destino sea el mismo. En el caso de una base de datos, compruebe que los nombres de host de `AWSCURRENT` y `AWSPENDING` sean los mismos.

test_secret

En `test_secret`, pruebe la versión de `AWSPENDING` del secreto; para ello, utilícelo para acceder a la base de datos o el servicio.

finish_secret

En `finish_secret`, utilice `update_secret_version_stage` para mover la etiqueta provisional `AWSCURRENT` de la versión anterior del secreto a la nueva. Secrets Manager agrega automáticamente la etiqueta provisional `AWSPREVIOUS` a la versión anterior, para que retenga la última versión buena conocida del secreto.

Programación de expresiones en la rotación de Secrets Manager

Cuando activa la rotación automática, puede utilizar una expresión cron() o rate() para establecer la programación de la rotación del secreto. Con una expresión rate, se puede crear una programación de rotación que se repita con un intervalo de horas o días. Con una expresión cron, puede crear programaciones de rotación más detalladas que un intervalo de rotación. Las programaciones de rotación de Secrets Manager utilizan la zona horaria UTC. Se puede rotar un secreto con una frecuencia máxima de cuatro horas. Secrets Manager rota el secreto en cualquier momento durante el periodo de rotación.

Para activar la rotación, consulte:

- the section called “Rotación automática de secretos de bases de datos (consola)”
- the section called “Rotación automática (consola)”
- the section called “Rotación automática (AWS CLI)”

Expresiones rate

Las expresiones rate de Secrets Manager tienen el siguiente formato, donde *`Value`* (Valor) es un número entero positivo y *`Unit`* (Unidad) puede ser `hour`, `hours`, `day` o `days`:

```
rate(Value Unit)
```

Se puede rotar un secreto con una frecuencia máxima de cuatro horas. Ejemplos:

- `rate(4 hours)` significa que el secreto se rota cada cuatro horas.
- `rate(1 day)` significa que el secreto se rota todos los días.
- `rate(10 days)` significa que el secreto se rota cada 10 días.

Para una frecuencia en horas, el periodo de rotación predeterminado comienza a medianoche y concluye al cabo de una hora. Se puede establecer el valor de Window duration (Duración del periodo) para cambiar el periodo de rotación. El periodo de rotación no debe prolongarse hasta el siguiente periodo de rotación. Una forma de comprobar esto es verificar que el periodo de rotación sea inferior o igual al número de horas entre rotaciones.

Para una frecuencia en días, el periodo de rotación predeterminado comienza a medianoche y concluye al final del día. Se puede establecer el valor de Window duration (Duración del periodo)

para cambiar el periodo de rotación. El periodo de rotación no debe prolongarse hasta el siguiente día UTC. Una forma de comprobar esto es verificar que la suma de la hora de inicio más la duración del periodo sea inferior o igual a 24 horas.

Expresiones cron

Las expresiones del tipo cron tienen el siguiente formato:

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

Una expresión cron que incluye incrementos de horas se restablece todos los días. Por ejemplo, `cron(0 4/12 * * ? *)` significa 4:00 h, 16:00 h, y al día siguiente 4:00 h, 16:00 h. Las programaciones de rotación de Secrets Manager utilizan la zona horaria UTC.

Para una programación en horas, la ventana de rotación predeterminada se cierra al cabo de una hora. Se puede establecer el valor de Window duration (Duración del periodo) para cambiar el periodo de rotación. El periodo de rotación no debe prolongarse hasta el siguiente periodo de rotación. Se puede rotar un secreto con una frecuencia máxima de cuatro horas.

Ejemplo de programación	Expresión
Cada ocho horas a partir de la medianoche.	`cron(0 /8 * * ? *)`
Cada ocho horas a partir de las 8:00 h.	`cron(0 8/8 * * ? *)`
Cada diez horas a partir de las 2:00 h. Los períodos de rotación comenzarán a las 2:00 h, 12:00 h y 22:00 h, y luego al día siguiente a las 2:00 h, 12:00 h y 22:00 h.	`cron(0 2/10 * * ? *)`
Todos los días a las 10:00 h.	`cron(0 10 * * ? *)`
Todos los sábados a las 18:00 h.	`cron(0 18 ? * SAT *)`
El primer día de cada mes a las 08:00 h.	`cron(0 8 1 * ? *)`
Los domingos a la 01:00 h, cada tres meses.	`cron(0 1 ? 1/3 SUN#1 *)`
El último día de cada mes a las 17:00 h.	`cron(0 17 L * ? *)`

Ejemplo de programación	Expresión
De lunes a viernes a las 08:00 h.	`cron(0 8 ? * MON-FRI *)`
Los días 1 y 15 de cada mes a las 16:00 h.	`cron(0 16 1,15 * ? *)`
El primer domingo de cada mes a medianoche.	`cron(0 0 ? * SUN#1 *)`

Requisitos para expresiones cron en Secrets Manager

En Secrets Manager existen algunas restricciones en cuanto a qué se puede utilizar en las expresiones cron. Una expresión cron para Secrets Manager debe tener el valor 0 en el campo correspondiente a los minutos, ya que los periodos de rotación de Secrets Manager comienzan a la hora en punto. Debe tener * en el campo correspondiente al año, ya que Secrets Manager no admite programaciones de rotación que tengan más de un año de diferencia. En la siguiente tabla se muestran las opciones que se pueden utilizar.

Campos	Valores	Caracteres comodín
Minutos	Debe ser 0	Ninguno
Hours	0–23	Utilice / (barra diagonal) para especificar los incrementos. Por ejemplo, `2/10` significa cada 10 horas a partir de las 2:00 h. Se puede rotar un secreto con una frecuencia máxima de cuatro horas.
Day-of-month	1–31	Utilice , (coma) para incluir valores adicionales. Por ejemplo, `1,15` significa el primer día y el día 15 del mes. Utilice - (guion) para especific ar un rango. Por ejemplo, `1–15` significa del día 1 al 15 del mes.

Campos	Valores	Caracteres comodín
		Utilice * (asterisco) para incluir todos los valores en el campo. Por ejemplo, * significa todos los días del mes. El comodín ? (signo de interrogación) especifica uno u otro. No se pueden especificar los campos `Day-of-month` y `Day-of-week` en la misma expresión Cron. Si especifica un valor en uno de los campos, debe utilizar un ? (signo de interrogación) en el otro. Utilice / (barra diagonal) para especificar los incrementos. Por ejemplo, `1/2` significa cada dos días a partir del día 1; es decir, los días 1, 3, 5, y así sucesivamente. Utilice L para especificar el último día del mes. Utilice ***DÍA***L para especificar el último día indicado del mes. Por ejemplo, SUNL significa el último domingo del mes.

Campos	Valores	Caracteres comodín
Mes	1–12 o ENE-DIC	Utilice , (coma) para incluir valores adicionales. Por ejemplo, `JAN,APR,JUL,OCT` significa enero, abril, julio y octubre. Utilice - (guion) para especificar un rango. Por ejemplo, `1-3` significa los meses del 1 al 3 del año. Utilice * (asterisco) para incluir todos los valores en el campo. Por ejemplo, `*` significa todos los meses. Utilice / (barra diagonal) para especificar los incrementos. Por ejemplo, `1/3` significa cada tres meses a partir del mes 1; es decir, los meses 1, 4, 7 y 10.

Campos	Valores	Caracteres comodín
Day-of-week	1–7 o DOM-SÁB	Utilice # para especificar el día de la semana de un mes. Por ejemplo, `TUE#3` significa el tercer martes del mes. Utilice , (coma) para incluir valores adicionales. Por ejemplo, `1,4` significa el primer y el cuarto día de la semana. Utilice - (guion) para especific ar un rango. Por ejemplo, `1-4` significa los días del 1 al 4 de la semana. Utilice * (asterisco) para incluir todos los valores en el campo. Por ejemplo, `*` significa todos los días de la semana. El comodín ? (signo de interrogación) especifica uno u otro. No se pueden especific ar los campos `Day-of-mo nth` y `Day-of-week` en la misma expresión Cron. Si especifica un valor en uno de los campos, debe utilizar un ? (signo de interrogación) en el otro. Utilice / (barra diagonal) para especificar los incrementos. Por ejemplo, `1/2` significa cada dos días de la semana a

Campos	Valores	Caracteres comodín
		partir del primer día; es decir, los días 1, 3, 5 y 7. Utilice L para especificar el último día de la semana.
Año	Debe ser *	Ninguno

Solucionar problemas de rotación AWS Secrets Manager

Para muchos servicios, Secrets Manager utiliza una función de Lambda para rotar secretos. Para obtener más información, consulte the section called “Cómo funciona la rotación”. La función de rotación de Lambda interactúa con la base de datos o el servicio para el que está destinado el secreto, así como con Secrets Manager. Si la rotación no funciona de la manera esperada, primero debe comprobar los CloudWatch registros.

> **Note**
>
> Algunos servicios pueden administrar los secretos por usted, incluida la administración de la rotación automática. Para obtener más información, consulte the section called “Rotación administrada”.

Para ver los CloudWatch registros de la función Lambda

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Elija el secreto y, a continuación, en la página de detalles, enRotation configuration (Configuración de rotación), elija la función de rotación de Lambda. Se abre la consola de Lambda.
3. En la pestaña Supervisar, elija Registros y, a continuación, elija Ver los registros iniciados CloudWatch.

   La CloudWatch consola se abre y muestra los registros de su función.

Interpretar los registros

No hay actividad después de “Found credentials in environment variables” (Se encontraron credenciales en variables de entorno)

Si no hay actividad después de “Found credentials in environment variables” (Se encontraron credenciales en variables de entorno) y la duración de la tarea es larga (por ejemplo, el tiempo de espera predeterminado de Lambda de 30 000 ms), es posible que la función de Lambda agote el tiempo de espera al intentar llegar al punto de conexión de Secrets Manager.

La función de rotación de Lambda debe ser capaz de acceder a un punto de enlace de Secrets Manager. Si la función de Lambda puede acceder a Internet, puede utilizar un punto de enlace público. Para buscar un punto de conexión, consulte the section called “Puntos de conexión de Secrets Manager”.

Si la función de Lambda se ejecuta en una VPC que no tiene acceso a Internet, recomendamos configurar los puntos de enlace privados del servicio de Secrets Manager dentro de la VPC. La VPC puede interceptar entonces las solicitudes dirigidas al punto de enlace regional público y redirigirlas al punto de enlace privado. Para obtener más información, consulte Punto de conexión VPC.

También puede habilitar la función de Lambda para acceder a un punto de conexión público de Secrets Manager. Para ello, agregue una puerta de enlace NAT o una puerta de enlace de Internet a su VPC. Esto permite que el tráfico de la VPC alcance el punto de conexión público. Esto expone a la VPC a más riesgo, ya que desde la red pública de Internet se puede atacar la dirección IP de la gateway.

No hay actividad después de createSecret

A continuación, se indican los problemas que pueden provocar que la rotación se detenga después de createSecret:

Las ACL de red de VPC no permiten la entrada ni la salida de tráfico HTTPS.

Para obtener más información, consulte Controlar el tráfico hacia las subredes utilizando las ACL de red en la Guía del usuario de Amazon VPC.

La configuración del tiempo de espera de la función de Lambda es demasiado corta para realizar la tarea.

Para obtener más información, consulte Configuración de las opciones de las funciones de Lambda en la Guía para desarrolladores de AWS Lambda .

El punto de conexión de VPC de Secrets Manager no permite los CIDR de VPC en la entrada en los grupos de seguridad asignados.

Para obtener más información, consulte Controlar el tráfico hacia los recursos mediante grupos de seguridad en la Guía del usuario de Amazon VPC.

La política de puntos de conexión de VPC de Secrets Manager no permite que Lambda utilice el punto de conexión de VPC.

Para obtener más información, consulte Punto de conexión VPC.

El secreto utiliza la rotación de usuarios alternativos, Amazon RDS administra el secreto del superusuario y la función de Lambda no puede acceder a la API de RDS.

Para alternar la rotación de usuarios en la que otro AWS servicio administra el secreto del superusuario, la función de rotación de Lambda debe poder llamar al punto final del servicio para obtener la información de conexión a la base de datos. Recomendamos configurar un punto de conexión de VPC para el servicio de base de datos. Para obtener más información, consulte:

- Puntos de conexión de VPC de la API y la interfaz de Amazon RDS en la Guía de usuario de Amazon RDS.
- Cómo trabajar con puntos de conexión de VPC en la Guía de administración de Amazon Redshift.

Error: “No se permite el acceso a KMS”

Si ve `ClientError: An error occurred (AccessDeniedException) when calling the GetSecretValue operation: Access to KMS is not allowed`, la función de rotación no tiene permiso para descifrar el secreto mediante la clave de KMS que se utilizó para cifrarlo. Es posible que haya una condición en la política de permisos que limite el contexto de cifrado a un secreto específico. Para obtener más información acerca del permiso necesario, consulte the section called “Instrucción de política para una clave administrada por el cliente”.

Error: “Key is missing from secret JSON” (Falta la clave en el JSON del secreto)

Una función de rotación de Lambda requiere que el valor del secreto esté en una estructura JSON específica. Si aparece este error, es posible que falte una clave en el JSON a la que la función de rotación intentó acceder. Para obtener información sobre la estructura JSON de cada tipo de secreto, consulte the section called “Estructura JSON de un secreto”.

Error: “setSecret: Unable to log into database” (setSecret: no se puede iniciar sesión en la base de datos)

A continuación, se indican los problemas que pueden provocar este error:

La función de rotación no puede acceder a la base de datos.

- Si la duración de la tarea es larga (por ejemplo, más de 5000 ms), es posible que la función de rotación de Lambda no pueda acceder a la base de datos a través de la red.

- Si la base de datos o el servicio se ejecutan en una instancia de Amazon EC2 en una VPC, es recomendable configurar la función de Lambda para que se ejecute en la misma VPC. A continuación, la función de rotación puede comunicarse directamente con el servicio. Para obtener más información, consulte Configuración del acceso a la VPC.

- Para permitir que la función de Lambda tenga acceso a la base de datos o el servicio, debe asegurarse de que los grupos de seguridad adjuntos a la función de rotación de Lambda permitan conexiones salientes a la base de datos o el servicio. Asimismo, debe asegurarse de que los grupos de seguridad adjuntos a la base de datos o el servicio permitan conexiones entrantes desde la función de rotación de Lambda.

Las credenciales del secreto son incorrectas.

Si la duración de la tarea es corta, es posible que la función de rotación de Lambda no pueda autenticarse con las credenciales del secreto. Compruebe las credenciales iniciando sesión manualmente con la información de `AWSPREVIOUS` las versiones `AWSCURRENT` y del secreto mediante el AWS CLI comando. `get-secret-value`

La base de datos utiliza **`scram-sha-256`** para cifrar las contraseñas.

Si la base de datos es Aurora PostgreSQL versión 13 o posterior y utiliza `scram-sha-256` para cifrar contraseñas, pero la función de rotación utiliza `libpq` versión 9 o posterior, que no admite `scram-sha-256`, la función de rotación no se puede conectar a la base de datos.

Para determinar qué usuarios de bases de datos utilizan cifrado con **`scram-sha-256`**

- Consulte Checking for users with non-SCRAM passwords (Búsqueda de usuarios con contraseñas que no sean de Scram) en la entrada de blog SCRAM Authentication in RDS for PostgreSQL 13 (Autenticación SCRAM en RDS para PostgreSQL 13).

Para determinar qué versión de **`libpq`** utiliza la función de rotación

1. En un equipo basado en Linux, en la consola de Lambda, vaya a la función de rotación y descargue el paquete de implementación. Descomprima el archivo zip en un directorio de trabajo.
2. En una línea de comandos, en el directorio de trabajo, ejecute:

   ```
   readelf -a libpq.so.5 | grep RUNPATH
   ```
3. Si ve la cadena *`PostgreSQL-9.4.x`*, o bien una versión principal inferior a 10, entonces la función de rotación no admite `scram-sha-256`.
   - Salida de una función de rotación que no admite `scram-sha-256`:

     ```
     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]
     ```
   - Salida de una función de rotación que admite `scram-sha-256`:

```
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**
>
> Si la rotación de secretos automática se configuró antes del 30 de diciembre de 2021, la función de rotación incluía en el paquete una versión antigua de `libpq` que no admite `scram-sha-256`. Para que se admita `scram-sha-256`, se debe volver a crear la función de rotación.

La base de datos requiere acceso SSL/TLS.

Si su base de datos requiere una conexión SSL/TLS, pero la función de rotación utiliza una conexión sin cifrar, dicha función no podrá conectarse a la base de datos. Las funciones de rotación de Amazon RDS (a excepción de Oracle y Db2) y Amazon DocumentDB utilizan una capa de sockets seguros (SSL) o una seguridad de la capa de transporte (TLS) de forma automática para conectarse a su base de datos, si está disponible. De lo contrario, utilizan una conexión no cifrada.

> **Note**
>
> Si configuró la rotación automática de secretos antes del 20 de diciembre de 2021, es posible que la función de rotación se base en una plantilla más antigua que no sea compatible con SSL/TLS. Es necesario Crear la función de rotación nuevamente para que sea compatible con las conexiones que utilizan SSL/TLS.

Para determinar cuándo se creó la función de rotación

1. Ingrese a la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/ y abra su secreto. En la sección Rotation configuration (Configuración de rotación), en Lambda rotation function (Función de rotación de Lambda), podrá ver Lambda

function ARN (ARN de la función de Lambda), por ejemplo, `arn:aws:lambda:aws-region:123456789012:function:SecretsManagerMyRotationFunction`. Copie el nombre de la función desde el final del ARN, que en este ejemplo sería `SecretsManagerMyRotationFunction`.

2. En la AWS Lambda consola https://console.aws.amazon.com/lambda/, en Funciones, pegue el nombre de la función Lambda en el cuadro de búsqueda, elija Entrar y, a continuación, elija la función Lambda.
3. En la página de detalles de la función, en la pestaña Configuration (Configuración), en Tags (Etiquetas), copie el valor junto a la clave aws:cloudformation:stack-name.
4. En la AWS CloudFormation consola https://console.aws.amazon.com/cloudformation, en Stacks, pegue el valor de la clave en el cuadro de búsqueda y, a continuación, seleccione Entrar.
5. La lista de pilas se filtra para que, de esta manera, aparezca únicamente la pila que creó la función de rotación de Lambda. En la columna Created date (Fecha de creación), consulte la fecha en que se creó la pila. Esta es la fecha en que se creó la función de rotación de Lambda.

Error: “No se puede importar el módulo 'lambda_function'”

Es posible que reciba este error si ejecuta una función de Lambda anterior que se actualizó automáticamente de Python 3.7 a una versión más reciente de Python. Para resolver el error, puede volver a cambiar la versión de la función de Lambda a Python 3.7 y, a continuación, the section called “Se ha actualizado una función de rotación existente de Python 3.7 a 3.9”. Para obtener más información, consulte ¿Por qué no se pudo rotar la función de Lambda de Secrets Manager y recibí el error “No se encontró el módulo pg”? en AWS re:Post.

Se ha actualizado una función de rotación existente de Python 3.7 a 3.9

Algunas funciones de rotación creadas antes de noviembre de 2022 utilizaban Python 3.7. El AWS SDK para Python dejó de ser compatible con Python 3.7 en diciembre de 2023. Para obtener más información, consulte Actualizaciones de la política de soporte de Python para AWS los SDK y las herramientas. Para cambiar a una nueva función de rotación que utilice Python 3.9, puede añadir una propiedad de tiempo de ejecución a una función de rotación existente o volver a crear la función de rotación.

Para encontrar las funciones de rotación de Lambda, utilice Python 3.7

1. Inicie sesión en la AWS Lambda consola AWS Management Console y ábrala en https://console.aws.amazon.com/lambda/.
2. En la lista Funciones, filtre por **`SecretsManager`**.
3. En la lista filtrada de funciones, en Tiempo de ejecución, busque Python 3.7.

Para actualizar a Python 3.9:

- Opción 1: Vuelva a crear la función de rotación mediante AWS CloudFormation
- Opción 2: actualice el tiempo de ejecución de la función de rotación existente mediante AWS CloudFormation
- Opción 3: Para AWS CDK los usuarios, actualice la biblioteca de CDK

Opción 1: Vuelva a crear la función de rotación mediante AWS CloudFormation

Cuando se utiliza la consola de Secrets Manager para activar la rotación, Secrets Manager se utiliza AWS CloudFormation para crear los recursos necesarios, incluida la función de rotación de Lambda. Si ha utilizado la consola para activar la rotación o ha creado la función de rotación mediante una AWS CloudFormation pila, puede utilizar la misma AWS CloudFormation pila para volver a crear la función de rotación con un nombre nuevo. La nueva función usa la versión más reciente de Python.

Para buscar la AWS CloudFormation pila que creó la función de rotación

- En la página de detalles de la función de Lambda, seleccione la pestaña Configuración, y elija Etiquetas. Vea el ARN junto a aws:cloudformation:stack-id.

  El nombre de la pila está incrustado en el ARN, como se muestra en el siguiente ejemplo.

  - ARN: `arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`
  - Nombre de pila: **`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**

Para recrear una función de rotación (AWS CloudFormation)

1. En AWS CloudFormation, busque la pila por su nombre y, a continuación, seleccione Actualizar.

Si aparece un cuadro de diálogo en el que se recomienda actualizar la pila raíz, seleccione Ir a la pila raíz y, a continuación, elija Actualizar.

2. En la página Actualizar pila, elija Editar plantilla en el diseñador y, a continuación, elija Ver en el diseñador.
3. En el diseñador, en el código de la plantilla `SecretRotationScheduleHostedRotationLambda`, sustituya el valor de para `"functionName": "SecretsManagerTestRotationRDS"` por un nuevo nombre de función, por ejemplo, en JSON, **`"functionName": "SecretsManagerTestRotationRDSupdated"`**
4. Continúe con el flujo de trabajo de la AWS CloudFormation pila y, a continuación, seleccione Enviar.

Opción 2: actualice el tiempo de ejecución de la función de rotación existente mediante AWS CloudFormation

Cuando se utiliza la consola de Secrets Manager para activar la rotación, Secrets Manager se utiliza AWS CloudFormation para crear los recursos necesarios, incluida la función de rotación de Lambda. Si ha utilizado la consola para activar la rotación o ha creado la función de rotación mediante una AWS CloudFormation pila, puede utilizar la misma AWS CloudFormation pila para actualizar el tiempo de ejecución de la función de rotación.

Para buscar la AWS CloudFormation pila que creó la función de rotación

- En la página de detalles de la función de Lambda, seleccione la pestaña Configuración, y elija Etiquetas. Vea el ARN junto a aws:cloudformation:stack-id.

  El nombre de la pila está incrustado en el ARN, como se muestra en el siguiente ejemplo.

  - ARN: `arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`
  - Nombre de pila: **`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**

Para actualizar el tiempo de ejecución de una función de rotación (AWS CloudFormation)

1. En AWS CloudFormation, busque la pila por su nombre y, a continuación, seleccione Actualizar.

   Si aparece un cuadro de diálogo en el que se recomienda actualizar la pila raíz, seleccione Ir a la pila raíz y, a continuación, elija Actualizar.
2. En la página Actualizar pila, elija Editar plantilla en el diseñador y, a continuación, elija Ver en el diseñador.
3. En el diseñador, en la plantilla JSON, para`SecretRotationScheduleHostedRotationLambda`, under`Properties`, under`Parameters`, agrega **`"runtime": "python3.9"`**
4. Continúe con el flujo de trabajo de la AWS CloudFormation pila y, a continuación, elija Enviar.

Opción 3: Para AWS CDK los usuarios, actualice la biblioteca de CDK

Si usó la versión AWS CDK anterior a la v2.94.0 para configurar la rotación de su secreto, puede actualizar la función Lambda actualizándola a la v2.94.0 o posterior. Para obtener más información, consulte Guía para desarrolladores de AWS Cloud Development Kit (AWS CDK) v2.

Secretos de AWS Secrets Manager administrados por otros servicios de AWS

Muchos servicios de AWS almacenan y utilizan secretos en AWS Secrets Manager. En algunos casos, estos secretos son secretos administrados, lo que significa que el servicio que los creó ayuda a administrarlos. Por ejemplo, algunos secretos administrados incluyen rotación administrada, de modo que no es necesario preocuparse de configurar la rotación. Además, es posible que el servicio de administración impida actualizar o eliminar secretos sin un periodo de recuperación, lo que ayuda a evitar interrupciones, ya que el servicio administrador depende del secreto.

Los secretos administrados utilizan una convención de nomenclatura que incluye el ID del servicio de administración para ayudar a identificarlos.

```
Secret name: ServiceID!MySecret
Secret ARN : arn:aws:us-east-1:ServiceID!MySecret-a1b2c3
```

ID de servicios que administran secretos

- `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 y Aurora”
- `redshift` – the section called “Amazon Redshift”
- `sqlworkbench` – the section called “Editor de consultas de Amazon Redshift v2”

Para buscar secretos administrados por otros servicios de AWS, consulte Búsqueda de secretos administrados.

Para obtener una lista completa de los servicios que utilizan secretos, consultethe section called "AWS servicios que utilizan AWS Secrets Manager secretos".

Amazon AppFlow

En Amazon AppFlow, al configurar una aplicación SaaS como origen o destino, se crea una conexión. Esto incluye la información necesaria para conectarse a las aplicaciones SaaS, como tokens de autenticación, nombres de usuario y contraseñas. Amazon AppFlow almacena los datos de tu conexión en un secreto gestionado por Secrets Manager con el prefijo`appflow`. El costo de almacenar el secreto está incluido en el cargo de Amazon AppFlow. Para obtener más información, consulta Protección de datos en Amazon AppFlow en la Guía del AppFlow usuario de Amazon.

AWS Glue DataBrew

AWS Glue DataBrew proporciona los siguientes pasos de receta `DETERMINISTIC_DECRYPT`, `DETERMINISTIC_ENCRYPT` y `CRYPTOGRAPHIC_HASH` para realizar transformaciones en la información de identificación personal (PII) de un conjunto de datos, que utiliza una clave de cifrado almacenada en un secreto de Secrets Manager. Si utiliza el secreto DataBrew predeterminado para almacenar la clave de cifrado, DataBrew crea un secreto gestionado con el prefijo`databrew`. El coste de almacenar el secreto está incluido en el coste de su uso DataBrew.

AWS DataSync

Para recopilar información sobre un sistema de almacenamiento local, AWS DataSync Discovery utiliza las credenciales de la interfaz de administración del sistema de almacenamiento. DataSync almacena esas credenciales en un secreto gestionado por Secrets Manager con el prefijo`datasync`. Se le cobrará ese secreto. Para obtener más información, consulte Añadir un sistema de almacenamiento local a DataSync Discovery en la Guía del AWS DataSyncusuario.

AWS Direct Connect

AWS Direct Connect almacena un nombre de clave de asociación de conectividad y un par de claves de asociación de conectividad (par CKN/CAK) en un secreto administrado con el prefijo `directconnect`. El costo del secreto está incluido en el cargo por AWS Direct Connect. Para actualizar el secreto, debe usar AWS Direct Connect en lugar de Secrets Manager. Para obtener más información, consulteAsociar un CKN/CAK de MACsec con un LAG en la Guía del usuario de AWS Direct Connect.

Amazon Elastic Container Service

Cuando utiliza Amazon ECS Service Connect, Amazon ECS utiliza los secretos de Secrets Manager para almacenar los certificados AWS Private Certificate Authority TLS. El costo de almacenar el secreto está incluido en los cargos de Amazon ECS. Para actualizar el secreto, debe utilizar Amazon ECS en lugar de Secrets Manager. Para obtener más información, consulte TLS con Service Connect en la Guía para desarrolladores de Amazon Elastic Container Service.

Amazon EventBridge

Al crear un destino de la EventBridge API de Amazon, EventBridge guarda su conexión en un secreto gestionado por Secrets Manager con el prefijo`events`. El costo de almacenar el secreto está incluido en el cargo por utilizar un destino de API. Para actualizar el secreto, debe usar EventBridge en lugar de Secrets Manager. Para obtener más información, consulta los destinos de las API en la Guía del EventBridge usuario de Amazon.

AWS Marketplace

Cuando utiliza AWS Marketplace Quick Launch, AWS Marketplace distribuye el software junto con la clave de licencia. AWS Marketplace almacena la clave de licencia en su cuenta como un secreto gestionado por Secrets Manager. El costo del secreto está incluido en el cargo por AWS Marketplace. Para actualizar el secreto, debe usar AWS Marketplace en lugar de Secrets Manager. Para obtener más información, consulte la configuración de Inicio Rápido en la AWS Marketplace Guía del Vendedor.

AWS OpsWorks for Chef Automate

Al crear un nuevo servidor enAWS OpsWorks CM, OpsWorks CM almacena la información del servidor en un secreto gestionado por Secrets Manager con el prefijo`opsworks-cm`. El costo del secreto está incluido en el cargo por AWS OpsWorks. Para obtener más información, consulte Integración con AWS Secrets Manager en la Guía del usuario de AWS OpsWorks.

Amazon RDS y Aurora

Para administrar las credenciales de usuario maestras de Amazon Relational Database Service (Amazon RDS), incluyendo Aurora, Amazon RDS puede encargarse de crear un secreto

administrado. Se le cobrará ese secreto. Amazon RDS también administra la rotación de estas credenciales. Para obtener más información, consulte Administración de contraseñas con Amazon RDS y AWS Secrets Manager en la Guía del usuario de Amazon RDS y Administración de contraseñas con Amazon Aurora y AWS Secrets Manager en la Guía del usuario de Amazon Aurora.

Para otras credenciales de Amazon RDS, consulte the section called “Creación de un secreto de base de datos”.

Amazon Redshift

Para administrar credenciales de administración de Amazon Redshift, Amazon Redshift puede crear un secreto administrado para usted. Se le cobrará ese secreto. Amazon Redshift también administra la rotación de estas credenciales. Para obtener más información, consulte Administración de contraseñas de administrador de Amazon Redshift mediante AWS Secrets Manager en la Guía de administración de Amazon Redshift.

Para obtener más credenciales de Amazon Redshift, consulte the section called “Creación de un secreto de base de datos”. Para usar un secreto de credenciales cuando se llama a la API de datos, consulte Uso de la API de datos de Amazon Redshift. Para utilizar un secreto cuando se usa el editor de consultas de Amazon Redshift con la finalidad de conectarse con una base de datos, visite Consultar una base de datos mediante el editor de consultas en la Guía de administración de Amazon Redshift y the section called “Editor de consultas de Amazon Redshift v2”.

Editor de consultas de Amazon Redshift v2

Cuando se utiliza el editor de consultas de Amazon Redshift v2 para conectarse a una base de datos, Amazon Redshift puede almacenar sus credenciales en un secreto administrado de Secrets Manager con el prefijo `sqlworkbench`. El costo de almacenar el secreto está incluido en el cargo por utilizar Amazon Redshift. Para actualizar el secreto, debe usar Amazon Redshift en lugar de Secrets Manager. Para obtener más información, consulte Trabajo con el editor de consultas v2 en la Guía de administración de Amazon Redshift.

Uso de un punto de conexión de VPC de AWS Secrets Manager

Recomendamos que ejecute tanto como pueda de su infraestructura en redes privadas que no sean accesibles desde la internet pública. Puede establecer una conexión privada entre su VPC y Secrets Manager mediante la creación de un punto de conexión de VPC de la interfaz. Los puntos de conexión de la interfaz cuentan con AWS PrivateLink, una tecnología que permite acceder de forma privada a las API de Secrets Manager sin necesidad de contar con una puerta de enlace de Internet, un dispositivo NAT, una conexión VPN o una conexión de AWS Direct Connect. Las instancias de la VPC no necesitan direcciones IP públicas para comunicarse con las API de Secrets Manager. El tráfico entre su VPC y Secrets Manager no sale de la red de AWS. Para obtener más información, consulte Puntos de conexión de VPC de interfaz (AWS PrivateLink) en la Guía del usuario de Amazon VPC.

Cuando Secrets Manager rota un secreto mediante una función de rotación de Lambda, por ejemplo, un secreto que contiene credenciales de base de datos, la función Lambda realiza solicitudes a la base de datos y a Secrets Manager. Cuando activa la rotación automática al utilizar la consola, Secrets Manager crea la función de Lambda en la misma VPC que la base de datos. Se recomienda que cree un punto de conexión de Secrets Manager en la misma VPC para que las solicitudes de la función de rotación de Lambda a Secrets Manager no salgan de la red de Amazon.

Si habilita un DNS privado para el punto de conexión, puede realizar solicitudes de API a Secrets Manager mediante su nombre de DNS predeterminado para la región, por ejemplo, `secretsmanager.us-east-1.amazonaws.com`. Para obtener más información, consulte Acceso a un servicio a través de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC.

Puede asegurarse de que las solicitudes a Secrets Manager provengan del acceso de la VPC mediante la inclusión de una condición en las políticas de permisos. Para obtener más información, consulte the section called “Ejemplo: permisos y VPC”.

También puede utilizar registros de AWS CloudTrail para auditar el uso de secretos a través del punto de conexión de VPC.

Para crear un punto de conexión de VPC de Secrets Manager

1. Consulte Creación de un punto final de interfaz en la Guía del usuario de Amazon VPC. Utilice el nombre del servicio: com.amazonaws.*region*.secretsmanager

2. Para controlar el acceso al punto final, consulte Controlar el acceso a los puntos finales de la VPC mediante políticas de punto final.

Subredes compartidas

No puede crear, describir, modificar ni eliminar puntos de conexión de VPC en subredes que se compartan con usted. No obstante, puede usar los puntos de conexión de VPC en las subredes que se compartan con usted. Para obtener información sobre el uso compartido de VPC, consulte Compartir su VPC con otras cuentas en la Guía del usuario de Amazon Virtual Private Cloud.

Creación de secretos de AWS Secrets Manager en AWS CloudFormation

Puede crear secretos en una pila de CloudFormation mediante el recurso `AWS::SecretsManager::Secret` en una plantilla de CloudFormation, tal como se muestra en Creación de un secreto.

Para crear un secreto de administrador para Amazon RDS o Aurora, le recomendamos que utilice `ManageMasterUserPassword` en `AWS::RDS::DBCluster`. A continuación, Amazon RDS crea el secreto y administra la rotación por usted. Para obtener más información, consulte Rotación administrada.

Para las credenciales de Amazon Redshift y Amazon DocumentDB, cree primero un secreto con una contraseña generada por Secrets Manager y, luego, utilice una referencia dinámica para recuperar el nombre de usuario y la contraseña del secreto y utilizarlos como credenciales para una base de datos nueva. A continuación, utilice el recurso `AWS::SecretsManager::SecretTargetAttachment` para agregar detalles sobre la base de datos al secreto que Secrets Manager necesita para rotar el secreto. Por último, para activar la rotación automática, utilice el recurso `AWS::SecretsManager::RotationSchedule` y proporcione una función de rotación y una programación. Consulte los siguientes ejemplos:

- Crear un secreto con credenciales de Amazon Redshift
- Crear un secreto con credenciales de Amazon DocumentDB

Para adjuntar una política de recursos a su secreto, utilice el recurso `AWS::SecretsManager::ResourcePolicy`.

Para obtener información sobre cómo crear recursos con AWS CloudFormation, consulte Información sobre los aspectos básicos de las plantillas en la Guía del usuario de AWS CloudFormation. También puede utilizar la AWS Cloud Development Kit (AWS CDK). Para obtener más información, consulte Biblioteca de construcción AWS Secrets Manager.

Creación de un secreto de AWS Secrets Manager con AWS CloudFormation

En este ejemplo, se crea un secreto denominado **`CloudFormationCreatedSecret-a1b2c3d4e5f6`**. El valor del secreto es el siguiente JSON, con una contraseña que consta de 32 caracteres y que se genera cuando se crea el secreto.

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

En este ejemplo, se utiliza el siguiente recurso de CloudFormation:

- AWS::SecretsManager::Secret

Para obtener información sobre cómo crear recursos con AWS CloudFormation, consulte Información sobre los aspectos básicos de las plantillas en la Guía del usuario de AWS CloudFormation.

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
```

Creación de un secreto de AWS Secrets Manager con rotación automática y una instancia de base de datos MySQL en Amazon RDS con AWS CloudFormation

Para crear un secreto de administrador para Amazon RDS o Aurora, le recomendamos que utilice `ManageMasterUserPassword`, como se muestra en el ejemplo Create a Secrets Manager secret for a master password (Crear un secreto de Secrets Manager para una contraseña maestra) en `AWS::RDS::DBCluster`. A continuación, Amazon RDS crea el secreto y administra la rotación por usted. Para obtener más información, consulte Rotación administrada.

Cree un AWS Secrets Manager secreto y un clúster de Amazon Redshift con AWS CloudFormation

Para crear un secreto de administrador para Amazon Redshift, le recomendamos que utilice los ejemplos en `AWS::Redshift::Cluster`y. `AWS::RedshiftServerless::Namespace`

Creación de un secreto de AWS Secrets Manager y una instancia de Amazon DocumentDB con AWS CloudFormation

En este ejemplo, se crea un secreto y una instancia de Amazon DocumentDB con las credenciales del secreto como el usuario y la contraseña. El secreto tiene asociada una política basada en recursos que define quién puede obtener acceso al secreto. La plantilla también crea una función de rotación de Lambda a partir de las Plantillas de función de rotación y configura el secreto para

que rote de forma automática entre las 8:00 h y las 10:00 h UTC del primer día de cada mes. Como práctica recomendada de seguridad, la instancia se encuentra en una Amazon VPC.

En este ejemplo, se utilizan los siguientes recursos de CloudFormation para Secrets Manager:

- AWS::SecretsManager::Secret
- AWS::SecretsManager::SecretTargetAttachment
- AWS::SecretsManager::RotationSchedule

Para obtener información sobre cómo crear recursos con AWS CloudFormation, consulte Información sobre los aspectos básicos de las plantillas en la Guía del usuario de AWS CloudFormation.

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 * ? *)'
```

Cómo Secrets Manager utiliza AWS CloudFormation

Cuando usa la consola para activar la rotación, Secrets Manager usa AWS CloudFormation para crear recursos para la rotación. Si crea una nueva función de rotación durante ese proceso, AWS CloudFormation crea un recurso `AWS::Serverless::Function` en función de las Plantillas de función de rotación adecuadas. Luego AWS CloudFormation establece la propiedad de `RotationSchedule`, que establece la función de rotación y las reglas de rotación del secreto. Puede ver la pila de AWS CloudFormation seleccionando View stack (Ver pila) en el banner después de activar la rotación automática.

Para obtener información sobre la activación de la rotación automática, consulte Rotar secretos de .

Creación de secretos de AWS Secrets Manager en AWS Cloud Development Kit (AWS CDK)

Para crear, administrar y recuperar secretos en una aplicación de CDK, puede usar la Biblioteca de constructos de AWS Secrets Manager, que contiene constructos de `ResourcePolicy`, `RotationSchedule`, `Secret`, `SecretRotation` y `SecretTargetAttachment`.

Para ver ejemplos, consulte estos temas:

- Creación de un secreto
- Importación de un secreto
- Recuperación de un secreto
- Concesión de permiso para usar el secreto
- Rotación de un secreto
- Rotación de un secreto de base de datos
- Replicación de un secreto a otras regiones

Para obtener más información acerca del CDK, consulte la Guía para desarrolladores del AWS Cloud Development Kit (AWS CDK) v2.

Monitorear secretos de AWS Secrets Manager

AWS proporciona herramientas de monitoreo para ver los secretos de Secrets Manager, informa cuando algo está mal y toma acciones automáticas cuando corresponde. Puede utilizar los registros si necesita investigar cualquier uso o cambio inesperado para luego poder revertir los cambios no deseados. También puede establecer verificaciones automatizadas para el uso inadecuado de los secretos y cualquier intento de eliminarlos.

Temas

Registro de eventos de AWS Secrets Manager con AWS CloudTrail

AWS CloudTrail registra todas las llamadas a la API para Secrets Manager como eventos, incluidas las llamadas de la consola de Secrets Manager, así como otros eventos para la rotación y la eliminación de la versión del secreto. Para obtener una lista de las entradas de registro de los registros de Secrets Manager, consulte CloudTrail entradas.

Puede utilizar la CloudTrail consola para ver los últimos 90 días de los eventos registrados. Para tener un registro continuo de los eventos de su AWS cuenta, incluidos los eventos de Secrets Manager, cree un registro que CloudTrail entregue los archivos de registro a un bucket de Amazon S3. Consulte Creación de una traza para la cuenta de AWS. También puede configurarlo CloudTrail para recibir archivos de CloudTrail registro de varios Cuentas de AWS y Regiones de AWS.

Puede configurar otros AWS servicios para analizar más a fondo los datos recopilados en los CloudTrail registros y actuar en función de ellos. Consulte las integraciones de AWS servicios con CloudTrail registros. También puede recibir notificaciones cuando CloudTrail publique nuevos archivos de registro en su bucket de Amazon S3. Consulte Configuración de las notificaciones de Amazon SNS para. CloudTrail

Para recuperar los eventos de Secrets Manager de CloudTrail los registros (consola)

1. Abra la CloudTrail consola en https://console.aws.amazon.com/cloudtrail/.

2. Asegúrese de que la consola apunta a la región en la que se han producido los eventos. La consola muestra únicamente aquellos eventos que se han producido en la región seleccionada. Elija la región en la lista desplegable en la esquina superior derecha de la consola.
3. En el panel de navegación de la izquierda, elija Event history (Historial de eventos).
4. Elija los criterios de Filter (Filtro) o un Time range (Intervalo de tiempo) para contribuir a encontrar el evento que está buscando. Por ejemplo, para ver todos los eventos Secrets Manager, para Select attribute (Seleccionar atributo), elija Event source (Origen del evento). A continuación, para Enter event source (Escribir origen del evento), elija **`secretsmanager.amazonaws.com`**.
5. Para ver otros detalles, elija la flecha de expansión situada junto al evento. Para ver toda la información disponible, elija View event (Ver evento).

AWS CLI

Example Recupera eventos de Secrets Manager de CloudTrail los registros

En el siguiente ejemplo de `lookup-events` se buscan eventos de Secrets Manager.

```
aws cloudtrail lookup-events \
    --region us-east-1 \
    --lookup-attributes
 AttributeKey=EventSource,AttributeValue=secretsmanager.amazonaws.com
```

Entradas de AWS CloudTrail para Secrets Manager

AWS Secrets Manager escribe entradas en su registro de AWS CloudTrail para todas las operaciones de Secrets Manager y para otros eventos relacionados con la rotación y la eliminación. Para obtener información acerca de cómo tomar medidas sobre estos eventos, consulte Combina los eventos de Secrets Manager con EventBridge.

Tipos de entrada de registro

- Entradas de registro para las operaciones de Secrets Manager
- Entradas de registro para la eliminación
- Entradas de registro para replicación
- Entradas de registro para la rotación

Entradas de registro para las operaciones de Secrets Manager

Los eventos que se generan mediante llamadas a las operaciones de Secrets Manager tienen `"detail-type": ["AWS API Call via CloudTrail"]`.

> **Note**
>
> Antes de febrero de 2024, algunas operaciones de Secrets Manager informaron de eventos que contenían «ArN» en lugar de «arn» para el ARN secreto. Para obtener más información, consulte AWS re:Post.

Las siguientes son CloudTrail entradas generadas cuando usted o un servicio llaman a las operaciones de Secrets Manager a través de la API, el SDK o la CLI.

BatchGetSecretValue

Generadas por la BatchGetSecretValueoperación. Para obtener información sobre cómo recuperar secretos, consulte Recuperar secretos.

CancelRotateSecret

Generado por la CancelRotateSecretoperación. Para obtener información acerca de la rotación, consulte Rotar secretos de .

CreateSecret

Generado por la CreateSecretoperación. Para obtener información sobre cómo crear secretos, consulte Cree y administre secretos.

DeleteResourcePolicy

Generado por la DeleteResourcePolicyoperación. Para obtener información acerca de los permisos, consulte Autenticación y control de acceso.

DeleteSecret

Generado por la DeleteSecretoperación. Para obtener información sobre la eliminación de secretos, consulte the section called “Eliminar un secreto”.

DescribeSecret

Generado por la DescribeSecretoperación.

GetRandomPassword

Generado por la GetRandomPasswordoperación.

GetResourcePolicy

Generado por la GetResourcePolicyoperación. Para obtener información acerca de los permisos, consulte Autenticación y control de acceso.

GetSecretValue

Generado por las BatchGetSecretValueoperaciones GetSecretValuey. Para obtener información sobre cómo recuperar secretos, consulte Recuperar secretos.

ListSecrets

Generado por la ListSecretsoperación. Para obtener información sobre cómo enumerar secretos, consulte the section called “Buscar secretos”.

ListSecretVersionIds

Generado por la ListSecretVersionIdsoperación.

PutResourcePolicy

Generado por la PutResourcePolicyoperación. Para obtener información acerca de los permisos, consulte Autenticación y control de acceso.

PutSecretValue

Generado por la PutSecretValueoperación. Para obtener información sobre la actualización de un secreto, consulte the section called “Modificar un secreto”.

RemoveRegionsFromReplication

Generado por la RemoveRegionsFromReplicationoperación. Para obtener información acerca de un secreto, consulte the section called “Replicar un secreto a otras regiones”.

ReplicateSecretToRegions

Generado por la ReplicateSecretToRegionsoperación. Para obtener información acerca de un secreto, consulte the section called “Replicar un secreto a otras regiones”.

RestoreSecret

Generado por la RestoreSecretoperación. Para obtener información sobre cómo restaurar un secreto eliminado, consulte the section called “Restaurar un secreto”.

RotateSecret

Generado por la RotateSecretoperación. Para obtener información acerca de la rotación, consulte Rotar secretos de .

StopReplicationToReplica

Generado por la StopReplicationToReplicaoperación. Para obtener información acerca de un secreto, consulte the section called “Replicar un secreto a otras regiones”.

TagResource

Generado por la TagResourceoperación. Para obtener más información acerca del etiquetado de un secreto, consulte the section called “Etiquetado de secretos de ”.

UntagResource

Generado por la UntagResourceoperación. Para obtener más información acerca de quitar las etiquetas de un secreto, consulte the section called “Etiquetado de secretos de ”.

UpdateSecret

Generado por la UpdateSecretoperación. Para obtener información sobre la actualización de un secreto, consulte the section called “Modificar un secreto”.

UpdateSecretVersionStage

Generado por la UpdateSecretVersionStageoperación. Para obtener información sobre las fases de versiones, consulte the section called “Versión”.

ValidateResourcePolicy

Generado por la ValidateResourcePolicyoperación. Para obtener información acerca de los permisos, consulte Autenticación y control de acceso.

Entradas de registro para la eliminación

Además de los eventos para las operaciones de Secrets Manager, Secrets Manager genera los siguientes eventos relacionados con la eliminación. Estos eventos tienen `"detail-type": ["AWS Service Event via CloudTrail"]`.

CancelSecretVersionDelete

Generado por el servicio de Secrets Manager. Si llama `DeleteSecret` en un secreto que tenga versiones, y luego llame a `RestoreSecret`, Secrets Manager registra este evento para cada

versión secreta que se ha restaurado. Para obtener información sobre cómo restaurar un secreto eliminado, consulte the section called “Restaurar un secreto”.

EndSecretVersionDelete

Generado por el servicio de Secrets Manager cuando se elimina una versión secreta. Para obtener más información, consulte the section called “Eliminar un secreto”.

StartSecretVersionDelete

Generado por el servicio de Secrets Manager cuando inicia la eliminación de una versión secreta. Para obtener información sobre la eliminación de secretos, consulte the section called “Eliminar un secreto”.

SecretVersionDeletion

Generado por el servicio de Secrets Manager cuando este elimina una versión obsoleta del secreto. Para obtener más información, consulte Versiones del secreto.

Entradas de registro para replicación

Además de los eventos para las operaciones de Secrets Manager, Secrets Manager genera los siguientes eventos relacionados con la replicación. Estos eventos tienen `"detail-type": ["AWS Service Event via CloudTrail"]`.

ReplicationFailed

Generado por el servicio de Secrets Manager cuando se produce un error en la replicación. Para obtener información acerca de un secreto, consulte the section called “Replicar un secreto a otras regiones”.

ReplicationStarted

Generado por el servicio de Secrets Manager cuando Secrets Manager inicia la replicación de un secreto. Para obtener información acerca de un secreto, consulte the section called “Replicar un secreto a otras regiones”.

ReplicationSucceeded

Generado por el servicio de Secrets Manager cuando un secreto se replica correctamente. Para obtener información acerca de un secreto, consulte the section called “Replicar un secreto a otras regiones”.

Entradas de registro para la rotación

Además de los eventos para las operaciones de Secrets Manager, Secrets Manager genera los siguientes eventos relacionados con la rotación. Estos eventos tienen `"detail-type": ["AWS Service Event via CloudTrail"]`.

RotationStarted

Generado por el servicio de Secrets Manager cuando inicia la rotación de un secreto. Para obtener información acerca de la rotación, consulte Rotar secretos de .

RotationAbandoned

Generado por el servicio de Secrets Manager cuando abandona un intento de rotación y elimina la etiqueta `AWSPENDING` de una versión existente de un secreto. Secrets Manager abandona la rotación cuando se crea una nueva versión de un secreto durante la rotación. Para obtener información acerca de la rotación, consulte Rotar secretos de .

RotationFailed

Generado por el servicio de Secrets Manager cuando se produce un error en la rotación. Para obtener información acerca de la rotación, consulte the section called “Solución de problemas de rotación de ”.

RotationSucceeded

Generado por el servicio de Secrets Manager cuando un secreto se rota correctamente. Para obtener información acerca de la rotación, consulte Rotar secretos de .

TestRotationStarted

Generado por el servicio de Secrets Manager cuando comienza a probar la rotación de un secreto que no está programado para la rotación inmediata. Para obtener información acerca de la rotación, consulte Rotar secretos de .

TestRotationSucceeded

Generado por el servicio de Secrets Manager cuando prueba, de forma exitosa, la rotación de un secreto que no está programado para la rotación inmediata. Para obtener información acerca de la rotación, consulte Rotar secretos de .

TestRotationFailed

Generado por el servicio de Secrets Manager cuando prueba la rotación de un secreto que no está programado para la rotación inmediata y se produce un error en la rotación. Para obtener

información acerca de la rotación, consulte the section called “Solución de problemas de rotación de ”.

Combina AWS Secrets Manager eventos con Amazon EventBridge

En Amazon EventBridge, puedes hacer coincidir los eventos de Secrets Manager con las entradas de CloudTrail registro. Puede configurar EventBridge reglas que busquen estos eventos y, a continuación, envíen los nuevos eventos generados a un objetivo para que tome medidas. Para obtener una lista de CloudTrail las entradas que Secrets Manager registra, consulteCloudTrail entradas. Para obtener instrucciones de configuración EventBridge, consulte Primeros pasos EventBridge en la Guía del EventBridge usuario.

Combinación de todos los cambios con un secreto especificado

El siguiente ejemplo muestra un patrón de EventBridge eventos que coincide con las entradas de registro para los cambios en un secreto.

```
{
    "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"]
        }
    }
}
```

Combinación de los eventos cuando rota un valor secreto

En el siguiente ejemplo, se muestra un patrón de EventBridge eventos que coincide con las entradas del CloudTrail registro para los cambios en los valores secretos que se producen como consecuencia de actualizaciones manuales o rotaciones automáticas. Como algunos de estos eventos provienen de las operaciones de Secrets Manager y otros están generados por el servicio de Secrets Manager, debe incluir `detail-type` para ambos.

```
{
    "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"]
    }
}
```

Monitoriza AWS Secrets Manager con Amazon CloudWatch

Puedes monitorizar AWS Secrets Manager con Amazon CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

En el caso de Secrets Manager, puedes utilizarlo CloudWatch para avisarte cuando tu ratio de solicitudes de API o el número de datos secretos de tu cuenta alcance un umbral específico. También puedes utilizarla CloudWatch para controlar los cargos estimados de Secrets Manager. Para obtener más información, consulte Creación de una alarma de facturación para monitorear los cargos estimados de AWS.

Temas

- Métricas y dimensiones de Secrets Manager
- Crear alarmas para supervisar las métricas de Secrets Manager
- Amazon CloudWatch Synthetics canarios

Métricas y dimensiones de Secrets Manager

El espacio de nombres de `AWS/SecretsManager` incluye las siguientes métricas.

Métrica	Descripción
`ResourceCount`	Número de secretos de su cuenta, incluidos los secretos marcados para su eliminación. La métrica se publica cada hora. Unidades: recuento

Dimensiones para las métricas de Secrets Manager.

Dimensión	Descripción
`Service`	El nombre del servicio de AWS que contiene el recurso. Para Secrets Manager, el valor de esta dimensión es `Secrets Manager`.
`Type`	El tipo de entidad que se registra. Para Secrets Manager, el valor de esta dimensión es `Resource`.
`Resource`	El tipo de recurso que se está ejecutando. Para Secrets Manager, el valor de esta dimensión es `SecretCount` .
`Class`	Ninguna.

Entre las solicitudes de la API Secrets Manager que puede supervisar mediante CloudWatch métricas se incluyen `GetSecretValue` `DescribeSecretListSecrets`,, y otras. Para buscar métricas, en la CloudWatch consola, selecciona Todas las métricas y, a continuación, en el cuadro de búsqueda, introduce el término de búsqueda, por ejemplo**`secrets`**.

Crear alarmas para supervisar las métricas de Secrets Manager

Puede crear una CloudWatch alarma que envíe un mensaje de Amazon SNS cuando el valor de la métrica cambie y haga que la alarma cambie de estado. Una alarma vigila una métrica durante el periodo especificado y realiza acciones en función del valor de la métrica relativo a un determinado umbral durante una serie de periodos de tiempo. Las alarmas invocan acciones únicamente en caso de cambios de estado sostenidos. CloudWatch las alarmas no invocan acciones simplemente porque se encuentran en un estado determinado; el estado debe haber cambiado y se ha mantenido durante un número específico de períodos.

Para obtener más información, consulte Uso de CloudWatch alarmas de Amazon y Creación de una CloudWatch alarma basada en la detección de anomalías.

Amazon CloudWatch Synthetics canarios

CloudWatch Los canarios de Amazon Synthetics son scripts configurables que se ejecutan de forma programada para supervisar los puntos de conexión y las API. Los Canaries siguen las mismas rutas y realizan las mismas acciones que un cliente, lo que le permite verificar continuamente su experiencia de cliente incluso cuando no tiene tráfico de clientes en sus aplicaciones.

Para ver un ejemplo de cómo integrar Secrets Manager, consulte Integración del valor controlado con otros servicios de AWS.

Monitoreo de los secretos de AWS Secrets Manager programados para su eliminación mediante Amazon CloudWatch

Puede utilizar una combinación de AWS CloudTrail, Amazon CloudWatch Logs y Amazon Simple Notification Service (Amazon SNS) para crear una alarma que le avise de cualquier intento de acceder a un secreto que esté pendiente de eliminación. Si recibe una notificación de una alarma de este tipo, es posible que prefiera cancelar la eliminación del secreto para disponer de más tiempo y poder determinar si realmente desea eliminarlo. Es posible que finalmente el secreto se restaure porque siga siendo necesario. Por otro lado, también es posible que necesite actualizar el usuario con los detalles del nuevo secreto que desee usar.

Los siguientes procedimientos explican cómo puede recibir una notificación cuando una solicitud de la operación `GetSecretValue` genera un mensaje de error específico que se escribe en los archivos de registro de CloudTrail. Se pueden realizar otras operaciones de API en el secreto sin activar la alarma. Esta alarma de CloudWatch detecta un uso que podría indicar que una persona o aplicación está utilizando credenciales obsoletas.

Antes de empezar con estos procedimientos, debe activar CloudTrail en la cuenta y Región de AWS donde tenga pensado monitorear las solicitudes de API de AWS Secrets Manager. Para obtener instrucciones, vaya a Creación de un registro de seguimiento por primera vez en la Guía del usuario de AWS CloudTrail.

Paso 1: Configurar el envío de archivos de registro de CloudTrail a CloudWatch Logs.

Debe configurar la entrega de sus archivos de registro de CloudTrail a CloudWatch Logs. Esto se hace para que CloudWatch Logs pueda monitorearlos con objeto de que las solicitudes a la API de Secrets Manager recuperen un secreto pendiente de eliminación.

Configurar la entrega de archivos de registro de CloudTrail a CloudWatch Logs

1. Abra la consola de CloudTrail en https://console.aws.amazon.com/cloudtrail/.
2. En la barra de navegación superior, elija la región de AWS para monitorear los secretos.
3. En el panel de navegación izquierdo, elija Trails (Registros de seguimiento) y, a continuación, elija el nombre del registro de seguimiento que va a configurar para CloudWatch.
4. En la página Trails Configuration (Configuración de registros de seguimiento), desplácese hacia abajo hasta la sección CloudWatch Logs y, a continuación, elija el icono de edición (✏).
5. Para New or existing log group, escriba un nombre del grupo de registros, como **`CloudTrail/MyCloudWatchLogGroup`**.
6. En IAM role (rol de IAM), puede usar el rol predeterminado, llamado CloudTrail_CloudWatchLogs_Role. Ese rol tiene una política de rol predeterminada con los permisos necesarios para enviar eventos de CloudTrail al grupo de registros.
7. Elija Continue (Continuar) para guardar la configuración.
8. En la página AWS CloudTrail will deliver CloudTrail events associated with API activity in your account to your CloudWatch Logs log group (CTlong entregará eventos de CloudTrail asociados con la actividad de la API en su cuenta a su grupo de registro de CloudWatch Logs), elija Allow (Permitir).

Paso 2: Crear la alarma de CloudWatch

Si desea recibir una notificación cuando una operación de la API `GetSecretValue` de Secrets Manager solicite acceso a un secreto pendiente de eliminación, debe crear una alarma de CloudWatch y configurar la notificación.

Para crear una alarma de CloudWatch

1. Inicie sesión en la consola de CloudWatch en https://console.aws.amazon.com/cloudwatch/.

2. En la barra de navegación superior, elija la región de AWS donde desee monitorear los secretos.
3. En el panel de navegación izquierdo, elija Logs (Registros).
4. En la lista Log Groups (Grupos de registros), seleccione la casilla situada junto al grupo que creó en el procedimiento anterior; por ejemplo, CloudTrail/MyCloudWatchLogGroup. A continuación, elija Create Metric Filter.
5. En Filter Pattern, escriba o pegue lo siguiente:

   ```
   { $.eventName = "GetSecretValue" && $.errorMessage = "*secret because it was marked
    for deletion*" }
   ```

   Elija Assign Metric (Asignar métrica).
6. En la página Create Metric Filter and Assign a Metric, haga lo siguiente:

   a. En Metric Namespace (Espacio de nombres de métrica), escriba **`CloudTrailLogMetrics`**.

   b. En Nombre de métrica, escriba **`AttemptsToAccessDeletedSecrets`**.

   c. Elija Show advanced metric settings y, a continuación, si es necesario para Metric Value, escriba **1**.

   d. Elija Create Filter.
7. En el cuadro de filtro, elija Create Alarm.
8. En la ventana Create Alarm, haga lo siguiente:

   a. En Name (Nombre), escriba **`AttemptsToAccessDeletedSecretsAlarm`**.

   b. Whenever: (Donde:), para is: (es:), elija **>=** y, a continuación, escriba **1**.

   c. Junto a Send notification to:, realice una de las siguientes acciones:

      - Para crear y utilizar un nuevo tema de Amazon SNS, elija New list (Nueva lista) y, a continuación, escriba un nuevo nombre de tema. En Email list:, escriba al menos una dirección de correo electrónico. Puede escribir varias direcciones de correo electrónico separándolas con comas.
      - Para utilizar un tema de Amazon SNS existente, elija el nombre del tema que desea usar. Si no existe ninguna lista, elija Select list (Seleccionar lista).

   d. Elija Create Alarm (Crear alarma).

Paso 3: Probar la alarma de CloudWatch

Para probar la alarma, cree un secreto y prográmelo para su eliminación. A continuación, intente recuperar el valor secreto. Al poco tiempo recibirá un correo electrónico en la dirección que haya configurado en la alarma. Es un aviso sobre el uso de un secreto programado para su eliminación.

Validación de la conformidad en AWS Secrets Manager

Su responsabilidad en cuanto a conformidad al usar servicios de Secrets Manager está determinada por la confidencialidad de sus datos, los objetivos de conformidad de su compañía y las leyes y regulaciones aplicables. AWS proporciona los siguientes recursos para ayudarle con la conformidad:

- Guías de inicio rápido de seguridad y conformidad: estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre arquitectura para seguridad y conformidad de HIPAA : en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones conformes con HIPAA.
- AWSRecursos de conformidad de: este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- AWS Config evalúa en qué medida las configuraciones de los recursos cumplen las prácticas internas, las directrices del sector y la normativa. Para obtener más información, consulte the section called “Auditoría de secretos para la conformidad”.
- AWS Security Hub proporciona una vista integral de su estado de seguridad en AWS que lo ayuda a verificar la conformidad con los estándares y las prácticas recomendadas del sector de seguridad. Para obtener más información sobre el uso de Security Hub para evaluar los recursos de Secrets Manager, consulte Controles de AWS Secrets Manager en la Guía del usuario de AWS Security Hub.
- IAM Access Analyzer analiza las políticas, incluidas las declaraciones de condición de una política, que permiten a una entidad externa acceder a un secreto. Para obtener más información, consulte Vista previa del acceso con las API de Access Analyzer.
- AWS Systems Manager proporciona manuales de procedimientos predefinidos para Secrets Manager. Para obtener más información, consulte Referencia del manual de procedimientos de Systems Manager Automation para Secrets Manager.

AWS Secrets Manager ha sido sometido a auditorías para los siguientes estándares y puede se parte de su solución cuando necesite obtener una certificación de conformidad.

AWS ha ampliado el programa de conformidad con la Ley de portabilidad y responsab ilidad de seguros médicos (HIPAA) para incluir a AWS Secrets Manager como servicio

	compatible con HIPAA. Si ha formalizado un Contrato de asociación empresarial (BAA, por sus siglas en inglés) con AWS, puede utilizar Secrets Manager para crear aplicaciones compatibles con HIPAA. AWS ofrece además un documento técnico dedicado a HIPAA para los clientes que deseen informarse acerca del modo de aprovechar AWS para procesar y almacenar información relacionada con el estado. Para obtener más información, consulte Conformidad con HIPAA.
	AWS Secrets Manager dispone de declaración de conformidad para el estándar de seguridad de datos del sector de tarjetas de pago (PCI DSS) versión 3.2 de nivel de proveedor de servicios 1. Los clientes que utilizan los productos y servicios de AWS para almacenar, procesar o transmitir datos de titulares de tarjetas pueden utilizar AWS Secrets Manager al administrar su propia certificación de conformidad con PCI DSS. Para obtener más información acerca de PCI DSS, incluido cómo solicitar una copia del Paquete de conformidad con PCI de AWS, consulte PCI DSS Nivel 1.
	AWS Secrets Manager ha obtenido las certificaciones de conformidad ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, e ISO 9001. Para obtener más información, consulte ISO 27001, ISO 27017, ISO 27018 e ISO 9001.
	Los informes de control de organizaciones y sistemas (SOC) son informes de análisis independientes de terceros que muestran cómo Secrets Manager logra los controles y objetivos clave de conformidad. La finalidad de estos informes es ayudarle a usted y a sus auditores a entender los controles de AWS que se han establecido como soporte a las operaciones y a la conformidad. Para obtener más información, consulte Conformidad con SOC.
	El Federal Risk and Authorization Management Program (FedRAMP) es un amplio programa gubernamental que ofrece un enfoque estandarizado para la supervisión continua, la autorización y la evaluación de la seguridad de servicios y productos en la nube. El Programa FedRAMP también proporciona autorizaciones provisionales para servicios y regiones en East/West (Este/Oeste) y GovCloud para consumir datos gubernamentales o regulados. Para obtener más información, consulte Conformidad con FedRAMP.

	La Guía de requisitos de seguridad (SRG, por sus siglas en inglés) de informática en la nube del Departamento de Defensa (DoD, por sus siglas en inglés) proporciona un proceso estandarizado de evaluación y autorización para que los proveedores de servicios de nube (CSP) obtengan una autorización provisional del DoD, de modo que puedan servir a los clientes del DoD. Para obtener más información, consulte Recursos de DoD SRG
	El Programa de Asesores Registrados de Seguridad de la Información (IRAP, por sus siglas en inglés) permite a los clientes del gobierno australiano validar que existen controles apropiados y determinar el modelo de responsabilidad adecuado para cumplir los requisitos del Manual de Seguridad de la Información (ISM, por sus siglas en inglés) del gobierno australiano producido por el Centro Australiano de Ciberseguridad (ACSC, por sus siglas en inglés). Para obtener más información, consulte Recursos de IRAP.
	Amazon Web Services (AWS) obtuvo la certificación OSPAR (Informe de Auditoría del Proveedor de Servicios Subcontratados). La alineación de AWS con las Directrices de la Asociación de Bancos de Singapur (ABS) sobre objetivos y procedimientos de control para proveedores de servicios externos (Directrices ABS) demuestra a los clientes el compromiso de AWS de satisfacer las altas expectativas para los proveedores de servicios de nube establecidas por la industria de servicios financieros en Singapur. Para obtener más información, consulte Recursos OSPAR.

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact.

Auditoría de secretos de AWS Secrets Manager para la conformidad mediante AWS Config

Puede utilizar AWS Config para evaluar los secretos y valorar en qué medida cumplen con sus prácticas internas, las directrices del sector y las normativas. Puede definir los requisitos internos de seguridad y cumplimiento para los secretos mediante reglas de AWS Config. Luego, AWS Config puede identificar los secretos que no se ajusten a las reglas. También puede realizar un seguimiento de los cambios de los metadatos de los secretos, la configuración de rotación, la clave KMS utilizada para cifrar el secreto, la función de rotación de Lambda y las etiquetas asociadas a un secreto.

Puede recibir notificaciones de Amazon SNS sobre las configuraciones de los secretos. Por ejemplo, puede recibir notificaciones de Amazon SNS sobre una lista de secretos no configurados para la rotación que le permite controlar las prácticas recomendadas de seguridad para la rotación de secretos.

Si tiene secretos en varias Cuentas de AWS y Regiones de AWS en la organización, puede agregar esos datos de configuración y cumplimiento.

Para agregar una nueva regla para los secretos

- Siga las instrucciones que aparecen en Uso de las reglas administradas de AWS Config y, a continuación, elija una de las siguientes reglas:
  - `secretsmanager-rotation-enabled-check`: verifica si se ha configurado la rotación para los secretos almacenados en Secrets Manager.
  - `secretsmanager-scheduled-rotation-success-check`: verifica si la última rotación correcta se encuentra dentro de la frecuencia de rotación configurada. La frecuencia mínima para la verificación es diariamente.
  - `secretsmanager-secret-periodic-rotation`: verifica si los secretos se rotaron dentro de la cantidad de días especificada.
  - `secretsmanager-secret-unused`: verifica si se accedió a los secretos dentro de la cantidad de días especificada.
  - `secretsmanager-using-cmk`: verifica si los secretos se cifran mediante Clave administrada de AWS `aws/secretsmanager` o una clave administrada por el cliente que creó en AWS KMS.

Una vez que se guarda la regla, AWS Config evalúa los secretos cada vez que cambian los metadatos de un secreto. Puede configurar AWS Config para que le notifique los cambios. Para obtener más información, consulte Notificaciones que AWS Config envía a un tema de Amazon SNS.

Agregar secretos de las Cuentas de AWS y las Regiones de AWS

Puede configurar el agregador de datos de varias cuentas y regiones de AWS Config para que revise las configuraciones de sus secretos en todas las cuentas y regiones de su organización y, a continuación, revisar las configuraciones secretas y compararlas con las prácticas recomendadas de administración de secretos.

Debe habilitar AWS Config y las reglas administradas de AWS Config específicas de los secretos en todas las cuentas y regiones antes de crear un agregador. Para obtener más información, consulte Utilice CloudFormation StackSets para aprovisionar recursos en varias Cuentas de AWS y Regiones.

Para obtener más información sobre el agregador de AWS Config, consulte Agregación de datos de varias cuentas y regiones y Configuración de un agregador mediante la consola en la Guía para desarrolladores de AWS Config.

Seguridad en AWS Secrets Manager

La seguridad de AWS es nuestra mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y un centro de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

Usted y AWS comparten la responsabilidad de la seguridad. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los programas de conformidad de AWS. Para obtener más información sobre los programas de conformidad que se aplican a AWS Secrets Manager, consulte Servicios de AWS en el ámbito del programa de conformidad.
- Seguridad en la nube: su servicio de AWS determina su responsabilidad. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Para obtener más recursos, consulte Pilar de Seguridad: Well-Architected Framework de AWS.

Temas

- Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager
- Protección de los datos en AWS Secrets Manager
- Cifrado y descifrado secretos en AWS Secrets Manager
- Seguridad de la infraestructura en AWS Secrets Manager
- Resiliencia en AWS Secrets Manager
- TLS postcuántico

Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager

Cuando utilice la AWS Command Line Interface (AWS CLI) para invocar las operaciones de AWS, escriba dichos comandos en un shell de comandos. Por ejemplo, puede utilizar el símbolo del sistema de Windows o Windows PowerShell o el Bash o Z shell, entre otros. Muchos de estos shells de comandos incluyen una funcionalidad diseñada para aumentar la productividad. Sin embargo, esta funcionalidad se puede utilizar para poner en riesgo sus secretos. Por ejemplo, en la mayoría de los shells, puede utilizar la tecla flecha arriba para ver el último comando escrito. La característica de historial de comandos puede ser explotada por cualquier persona que acceda a su sesión no protegida. Además, otras utilidades que funcionan en segundo plano podrían obtener acceso a los parámetros de comandos, con el fin de ayudarle a realizar las tareas con más eficacia. Para reducir estos riesgos, asegúrese de que realiza los pasos siguientes:

- Bloquee siempre el equipo cuando abandona la consola.
- Desinstale o deshabilite las utilidades de la consola que ya no necesita o no usa.
- Asegúrese de que el shell o el programa de acceso remoto, si está utilizando uno, no registren los comandos que se escriben.
- Utilice técnicas para pasar parámetros que no se registren en el historial de comandos del shell. En el siguiente ejemplo, se muestra cómo puede escribir el texto del secreto en un archivo de texto, que después se transfiere al comando de AWS Secrets Manager y se destruye inmediatamente. Esto significa que el texto del secreto no se captura en el historial de shell habitual.

  En el siguiente ejemplo se muestran los comandos de Linux habituales (es posible que su shell necesite unos comandos ligeramente diferentes):

```
$ 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.
```

Después de ejecutar estos comandos, puede usar las flechas de dirección arriba y abajo para desplazarse por el historial de comandos y comprobar que el texto del secreto no se muestra en ninguna línea.

> **Important**
>
> De forma predeterminada, no puede realizar una técnica equivalente en Windows a menos que reduzca primero el tamaño del búfer del historial de comandos a 1.

Para configurar la ventana del símbolo del sistema de Windows de forma que solo tenga un búfer de historial de comandos de un comando

1. Abra un símbolo del sistema de administrador (Run as administrator (Ejecutar como administrador)).
2. Elija el icono en la parte superior izquierda y, a continuación, elija Properties (Propiedades).
3. En la pestaña Opciones, establezca Tamaño del búfer y Número de búferes en **1**, y después elija Aceptar.
4. Siempre que tenga que escribir un comando que no desea que aparezca en el historial, escriba inmediatamente después otro comando como:

   ```
   echo.
   ```

   Esto garantiza la purga del comando confidencial.

Para el shell del símbolo del sistema de Windows, puede descargar la herramienta SysInternals SDelete y después utilizar comandos similares a los siguientes:

```
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.
```

Protección de los datos en AWS Secrets Manager

El modelo de responsabilidad compartida de AWS se aplica a la protección de datos de AWS Secrets Manager. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye la configuración de seguridad y las tareas de administración para el que utiliza Servicios de AWS. Para obtener más información sobre la privacidad de los datos, consulte las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog AWSShared Responsability Model and GDPR en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de Cuenta de AWS y configurar cuentas de usuario individuales con AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes formas:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Secrets Manager admite TLS 1.2 y 1.3 en todas las regiones. Secrets Manager también admite un protocolo de cifrado de red con opción de intercambio de claves postcuántico para TLS (PQTLS) híbrida.
- Firme las solicitudes programáticas a Secrets Manager utilizando un ID de clave de acceso y una clave de acceso secreta asociada a una entidad principal de IAM. O bien puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar las solicitudes.

- Configure la API y el registro de actividad del usuario con AWS CloudTrail. Consulte the section called “Inicio de sesión con AWS CloudTrail ”.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de conexión de FIPS. Consulte the section called “Puntos de conexión de Secrets Manager”.
- Si utiliza la AWS CLI para acceder a Secrets Manager, the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

Cifrado en reposo

Secrets Manager utiliza el cifrado a través de AWS Key Management Service (AWS KMS) para proteger la confidencialidad de los datos en reposo. AWS KMS proporciona un servicio de almacenamiento de claves y de cifrado que utilizan muchos servicios de AWS. Cada secreto de Secrets Manager se cifra con una clave de datos única. Cada clave de datos está protegida mediante una clave de KMS. Puede optar por utilizar el cifrado predeterminado con Clave administrada de AWS de Secrets Manager para la cuenta o puede crear su propia clave administrada por el cliente en AWS KMS. El uso de una clave administrada por el cliente le da un control de autorización más detallado sobre las actividades clave de KMS. Para obtener más información, consulte the section called “Cifrado y descifrado de secretos”.

Cifrado en tránsito

Secrets Manager proporciona puntos de enlace seguros y privados para cifrar datos en tránsito. Los puntos de conexión seguros y privados permiten a AWS proteger la integridad de las solicitudes de la API a Secrets Manager. AWS requiere que las llamadas a la API sean firmadas por el autor de la llamada utilizando certificados X.509 o una clave de acceso secreta de Secrets Manager. Este requisito se indica en Proceso de firma de Signature Versión 4 (Sigv4).

Si utiliza la AWS Command Line Interface (AWS CLI) o cualquiera de los SDK de AWS para realizar llamadas a AWS, usted configura la clave de acceso que se va a utilizar. A continuación, esas herramientas utilizan automáticamente la clave de acceso para firmar las solicitudes por usted. Consulte the section called “Reducción de los riesgos de usar AWS CLI para almacenar sus secretos de AWS Secrets Manager”.

Privacidad del tráfico entre redes

AWS ofrece opciones para mantener la privacidad al enrutar el tráfico a través de rutas de red conocidas y privadas.

Tráfico entre el servicio y las aplicaciones y clientes locales

Tiene dos opciones de conectividad entre su red privada y AWS Secrets Manager:

- Una conexión de Site-to-Site VPN de AWS. Para obtener más información, consulte ¿Qué es AWS Site-to-Site VPN?
- Una conexión AWS Direct Connect. Para obtener más información, consulte ¿Qué es AWS Direct Connect?

Tráfico entre recursos de AWS en la misma región

Si quiere proteger el tráfico entre Secrets Manager y los clientes de API en AWS, configure un AWS PrivateLink para acceder de forma privada a los puntos de conexión de la API de Secrets Manager.

Administración de claves de cifrado

Cuando Secrets Manager necesita cifrar una nueva versión de los datos secretos protegidos, Secrets Manager envía una solicitud a AWS KMS para generar una nueva clave de datos desde la clave de KMS. Secrets Manager utiliza esta clave de datos para el cifrado de sobres. Secrets Manager almacena la clave de datos cifrada con el secreto cifrado. Cuando el secreto necesita ser descifrado, Secrets Manager pregunta a AWS KMS para descifrar la clave de datos. A continuación, Secrets Manager utiliza la clave de datos descifrada para descifrar el secreto cifrado. Secrets Manager nunca almacena la clave de datos en forma no cifrada y elimina la clave de la memoria lo antes posible. Para obtener más información, consulte the section called “Cifrado y descifrado de secretos”.

Cifrado y descifrado secretos en AWS Secrets Manager

Secrets Manager utiliza el cifrado de sobres con AWS KMS claves y claves de datos para proteger cada valor secreto. Cada vez que el valor secreto de un secreto cambia, Secrets Manager solicita una nueva clave de datos AWS KMS para protegerlo. La clave de datos se cifra como una clave de KMS y se almacena en los metadatos del secreto. Para descifrar el secreto, Secrets Manager primero descifra la clave de datos cifrados utilizando la clave KMS in. AWS KMS

Secrets Manager no utiliza la clave KMS para cifrar directamente el valor del secreto. En cambio, utiliza la clave KMS para generar y cifrar una clave de datos simétrica AES (Advanced Encryption Standard) de 256 bits y utiliza la clave de datos para cifrar el valor del secreto. Secrets Manager utiliza la clave de datos de texto simple para cifrar el valor secreto fuera de la memoria y AWS KMS,

a continuación, lo elimina de la memoria. Almacena la copia cifrada de la clave de datos en los metadatos del secreto.

Al crear un secreto, puede elegir cualquier clave de cifrado simétrico gestionada por el cliente en la región Cuenta de AWS y, si lo prefiere, puede utilizar Secrets Manager (`aws/secretsmanager`). Clave administrada de AWS Si eliges el Clave administrada de AWS `aws/secretsmanager` y aún no existe, Secrets Manager lo crea y lo asocia al secreto. Puede utilizar la misma clave KMS o diferentes claves KMS para cada secreto de su cuenta. Es posible que desee utilizar diferentes claves de KMS para establecer permisos personalizados en las claves de un grupo de secretos, o si desea auditar operaciones específicas para esas claves. Secrets Manager solamente admite claves KMS de cifrado simétricas. Si utiliza una clave de KMS en un almacén de claves externo, las operaciones criptográficas en la clave de KMS pueden tardar más y ser menos fiables y duraderas, ya que la solicitud tiene que viajar fuera de AWS.

Para obtener información sobre cómo cambiar la clave de cifrado de un secreto, consulte the section called “Cambiar la clave de cifrado de un secreto”.

Al cambiar la clave de cifrado, Secrets Manager vuelve a cifrar `AWSCURRENT` las `AWSPREVIOUS` versiones con la nueva clave. `AWSPENDING` Para evitar que descubras el secreto, Secrets Manager mantiene todas las versiones existentes cifradas con la clave anterior. Esto significa que puedes descifrar todas `AWSCURRENT` las `AWSPENDING` `AWSPREVIOUS` versiones con la clave anterior o con la nueva clave.

Para que solo se `AWSCURRENT` pueda descifrar con la nueva clave de cifrado, cree una nueva versión del secreto con la nueva clave. Luego, para poder descifrar la versión `AWSCURRENT` secreta, debe tener permiso para usar la nueva clave.

Para encontrar la clave KMS asociada a un secreto, consulte el secreto en la consola o llame ListSecretso DescribeSecret. Cuando el secreto está asociado a Secrets Manager (`aws/secretsmanager`), estas operaciones no devuelven un identificador clave de KMS. Clave administrada de AWS

Temas

- ¿Qué se cifra?
- Procesos de cifrado y descifrado
- Permisos para la clave KMS
- Cómo Secrets Manager utiliza su clave KMS

- Política de clave de la Clave administrada de AWS (aws/secretsmanager)
- Contexto de cifrado en Secrets Manager
- Supervise la interacción de Secrets Manager con AWS KMS

¿Qué se cifra?

Secrets Manager cifra el valor secreto, pero no cifra lo siguiente:

- Nombre y descripción del secreto
- Ajustes de rotación
- ARN de la clave KMS asociada al secreto
- Cualquier AWS etiqueta adjunta

Procesos de cifrado y descifrado

Para cifrar el valor de secreto en un secreto, Secrets Manager utiliza el siguiente proceso.

1. Secrets Manager llama a la AWS KMS GenerateDataKeyoperación con el ID de la clave KMS del secreto y una solicitud de clave simétrica AES de 256 bits. AWS KMS devuelve una clave de datos en texto plano y una copia de esa clave de datos cifrada con la clave KMS.
2. Secrets Manager utiliza la clave de datos de texto sin formato y el algoritmo Advanced Encryption Standard (AES) para cifrar el valor secreto fuera de. AWS KMS Elimina la clave de texto no cifrado de la memoria lo antes posible tras utilizarla.
3. Secrets Manager almacena la clave de datos cifrada en los metadatos del secreto por lo que está disponible para descifrar el valor del secreto. Sin embargo, ninguna de las API de Secrets Manager devuelve el secreto cifrado o la clave de datos cifrada.

Para descifrar un valor de secreto cifrado:

1. Secrets Manager llama a la operación de AWS KMS descifrado y pasa la clave de datos cifrados.
2. AWS KMS utiliza la clave KMS como secreto para descifrar la clave de datos. Devuelve la clave de datos de texto no cifrado.
3. Secrets Manager usa la clave de datos en texto no cifrado para descifrar el valor del secreto. A continuación, elimina la clave de datos de la memoria lo antes posible.

Permisos para la clave KMS

Cuando Secrets Manager utiliza una clave KMS en las operaciones criptográficas, actúa en nombre del usuario que está creando o modificando el valor del secreto. Puede conceder estos permisos en una política de IAM o en una política de claves. Las siguientes operaciones de Secrets Manager requieren AWS KMS permisos.

- CreateSecret
- GetSecretValue
- PutSecretValue
- UpdateSecret
- ReplicateSecretToRegions

Para permitir que la clave KMS se use solo para las solicitudes que se originan en Secrets Manager, en la política de permisos, puede usar la clave de ViaService condición kms: con el `secretsmanager.<Region>.amazonaws.com` valor.

También puede utilizar las claves o los valores en el contexto de cifrado como condición para utilizar la clave KMS para operaciones criptográficas. Por ejemplo, puede utilizar un operador de condición de cadena en un documento de IAM o de políticas de claves, o bien utilizar una restricción de concesión en una concesión. La propagación de la concesión de claves de KMS puede tardar hasta cinco minutos. Para obtener más información, consulte CreateGrant.

Cómo Secrets Manager utiliza su clave KMS

Secrets Manager realiza las siguientes AWS KMS operaciones con su clave KMS.

GenerateDataKey

Secrets Manager llama a la AWS KMS GenerateDataKeyoperación en respuesta a las siguientes operaciones de Secrets Manager.

- CreateSecret— Si el nuevo secreto incluye un valor secreto, Secrets Manager solicita una nueva clave de datos para cifrarlo.
- PutSecretValue— Secrets Manager solicita una nueva clave de datos para cifrar el valor secreto especificado.
- ReplicateSecretToRegions— Para cifrar el secreto replicado, Secrets Manager solicita una clave de datos para la clave de KMS en la región de réplica.

- UpdateSecret— Si cambias el valor secreto o la clave KMS, Secrets Manager solicita una nueva clave de datos para cifrar el nuevo valor secreto.

La RotateSecretoperación no llama `GenerateDataKey` porque no cambia el valor secreto. No obstante, si la función de Lambda que `RotateSecret` invoca cambia el valor del secreto, su llamada a la operación `PutSecretValue` activa una `GenerateDataKey` solicitud.

Decrypt

Secrets Manager llama a la operación Decrypt en respuesta a las siguientes operaciones de Secrets Manager.

- GetSecretValuey BatchGetSecretValue— Secrets Manager descifra el valor secreto antes de devolvérselo a la persona que llama. Para descifrar un valor secreto cifrado, Secrets Manager llama a la operación AWS KMS Decrypt para descifrar la clave de datos cifrados del secreto. A continuación, usa la clave de datos en texto no cifrado para descifrar el valor del secreto cifrado. Para los comandos por lotes, Secrets Manager puede reutilizar la clave descifrada, por lo que no todas las llamadas dan lugar a una `Decrypt` solicitud.
- PutSecretValuey UpdateSecret: la mayoría de `UpdateSecret` las solicitudes `PutSecretValue` y no activan ninguna operación. `Decrypt` Sin embargo, cuando una solicitud `PutSecretValue` o `UpdateSecret` intenta cambiar el valor del secreto en una versión existente de un secreto, Secrets Manager descifra el valor del secreto existente y lo compara con el valor del secreto en la solicitud para confirmar que son iguales. Esta acción garantiza que las operaciones de Secrets Manager son idempotentes. Para descifrar un valor secreto cifrado, Secrets Manager llama a la operación AWS KMS Decrypt para descifrar la clave de datos cifrados del secreto. A continuación, usa la clave de datos en texto no cifrado para descifrar el valor del secreto cifrado.
- ReplicateSecretToRegions— Secrets Manager primero descifra el valor secreto en la región principal antes de volver a cifrar el valor secreto con la clave KMS en la región de réplica.

Encrypt

Secrets Manager llama a la operación Encrypt en respuesta a las siguientes operaciones de Secrets Manager:

- UpdateSecret— Si cambias la clave de KMS, Secrets Manager vuelve a cifrar la clave de datos que protege las `AWSCURRENT` versiones `AWSPENDING` secretas y las versiones secretas con la nueva clave. `AWSPREVIOUS`
- ReplicateSecretToRegions— Secrets Manager vuelve a cifrar la clave de datos durante la replicación mediante la clave KMS de la región de réplica.

DescribeKey

Secrets Manager llama a la DescribeKeyoperación para determinar si se debe incluir la clave KMS al crear o editar un secreto en la consola de Secrets Manager.

Validación del acceso a la clave KMS

Al establecer o cambiar la clave KMS asociada con el secreto, Secrets Manager llama a las operaciones `GenerateDataKey` y `Decrypt` con la clave KMS especificada. Estas llamadas confirman que el intermediario tiene permiso para utilizar la clave KMS para estas operaciones. Secrets Manager descarta los resultados de estas operaciones; no las utiliza en ninguna operación criptográfica.

Puede identificar estas llamadas de validación, ya que el valor del `SecretVersionId`contexto de cifrado de la clave en estas solicitudes es `RequestToValidateKeyAccess`.

> **Note**
>
> En el pasado, las llamadas de validación de Secrets Manager no incluían un contexto de cifrado. Es posible que encuentres llamadas sin contexto de cifrado en AWS CloudTrail los registros más antiguos.

Política de clave de la Clave administrada de AWS (**`aws/secretsmanager`**)

La política clave de Secrets Manager (`aws/secretsmanager`) otorga a los usuarios permiso para usar la clave KMS para operaciones específicas solo cuando Secrets Manager realiza la solicitud en nombre del usuario. Clave administrada de AWS La política de claves no permite a ningún usuario utilizar la clave KMS directamente.

Esta política de claves, como las políticas de todas las Claves administradas por AWS, la establece el servicio. No puede cambiar la política de claves, pero puede verla en cualquier momento. Para obtener más detalles, consulte Ver una política de clave.

Las declaraciones de política de la política de claves tienen el siguiente efecto:

- Permitir a los usuarios de la cuenta utilizar la clave KMS para operaciones criptográficas solo cuando la solicitud proviene de Secrets Manager en su nombre. La clave de condición `kms:ViaService` aplica esta restricción.

- Permite a la AWS cuenta crear políticas de IAM que permiten a los usuarios ver las propiedades clave de KMS y revocar las concesiones.
- Aunque Secrets Manager no utiliza concesiones para obtener acceso a la clave de KMS, la política también permite a Secrets Manager crear concesiones para la clave KMS en nombre del usuario y permite a la cuenta revocar cualquier concesión que permite a Secrets Manager usar la clave KMS. Estos son los elementos estándar del documento de políticas para una Clave administrada de AWS.

La siguiente es una política clave como Clave administrada de AWS ejemplo de Secrets Manager.

```
{
  "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": "*"
    }
  ]
}
```

Contexto de cifrado en Secrets Manager

Un contexto de cifrado es un conjunto de pares de clave-valor que contienen datos no secretos arbitrarios. Al incluir un contexto de cifrado en una solicitud de cifrado de datos, vincula AWS KMS criptográficamente el contexto de cifrado a los datos cifrados. Para descifrar los datos, es necesario pasar el mismo contexto de cifrado.

En sus solicitudes GenerateDataKeyy en las de Decrypt AWS KMS, Secrets Manager utiliza un contexto de cifrado con dos pares de nombre-valor que identifican el secreto y su versión, como se muestra en el siguiente ejemplo. Los nombres no varían, pero los valores de contexto de cifrado combinado serán diferentes para cada valor de secreto.

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
a1b2c3",
    "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

Puede usar el contexto de cifrado para identificar estas operaciones criptográficas en registros y registros de auditoría, como AWS CloudTrailAmazon CloudWatch Logs, y como condición para la autorización en políticas y concesiones.

El contexto de cifrado de Secrets Manager se compone de dos pares de nombre-valor.

- SecretARN: el primer par de nombre-valor identifica el secreto. La clave es `SecretARN`. El valor es el Nombre de recurso de Amazon (ARN) del secreto.

  ```
  "SecretARN": "ARN of an Secrets Manager secret"
  ```

  Por ejemplo, si el ARN del secreto fuera `arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3`, el contexto de cifrado incluiría el siguiente par.

  ```
  "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
  a1b2c3"
  ```

- SecretVersionId— El segundo par nombre-valor identifica la versión del secreto. La clave es `SecretVersionId`. El valor es el ID de la versión.

  ```
  "SecretVersionId": "<version-id>"
  ```

  Por ejemplo, si el ID de versión del secreto fuera `EXAMPLE1-90ab-cdef-fedc-ba987SECRET1`, el contexto de cifrado incluiría el siguiente par.

  ```
  "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
  ```

Cuando estableces o cambias la clave KMS de un secreto, Secrets Manager envía GenerateDataKeyy descifra solicitudes AWS KMS para validar que la persona que llama tiene permiso para usar la clave KMS para estas operaciones. Descarta las respuestas; no las utiliza en el valor del secreto.

En estos solicitudes de validación, el valor de `SecretARN` es el ARN real del secreto, pero el valor `SecretVersionId` es `RequestToValidateKeyAccess`, tal y como se muestra en el siguiente ejemplo de contexto de cifrado. Este valor especial le ayudará a identificar las solicitudes de validación en los registros y las pistas de auditoría.

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
a1b2c3",
    "SecretVersionId": "RequestToValidateKeyAccess"
}
```

> **Note**
>
> En el pasado, las solicitudes de validación de Secrets Manager no incluían un contexto de cifrado. Es posible que encuentres llamadas sin contexto de cifrado en registros antiguos AWS CloudTrail .

Supervise la interacción de Secrets Manager con AWS KMS

Puedes usar AWS CloudTrail Amazon CloudWatch Logs para realizar un seguimiento de las solicitudes que Secrets Manager envía AWS KMS en tu nombre. Para obtener más información acerca del monitoreo del uso de los secretos, consulte Monitorear secretos.

GenerateDataKey

Al crear o cambiar el valor secreto de un secreto, Secrets Manager envía una GenerateDataKeysolicitud a la AWS KMS que se especifica la clave KMS del secreto.

El evento que registra la operación `GenerateDataKey` es similar al siguiente evento de ejemplo. La solicitud la invoca `secretsmanager.amazonaws.com`. Los parámetros incluyen el nombre de recurso de Amazon (ARN) de la clave KMS para el secreto, un especificador de clave que requiere una clave de 256 bits y el contexto de cifrado que identifica el secreto y la versión.

```
{
```

```
    "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"
}
```

Decrypt

Cuando obtienes o cambias el valor secreto de un secreto, Secrets Manager envía una solicitud de descifrado AWS KMS a para descifrar la clave de datos cifrados. Para los comandos por lotes, Secrets Manager puede reutilizar la clave descifrada, por lo que no todas las llamadas dan lugar a una `Decrypt` solicitud.

El evento que registra la operación `Decrypt` es similar al siguiente evento de ejemplo. El usuario principal de su AWS cuenta que accede a la tabla. Los parámetros incluyen la clave de la tabla cifrada (como un bloque de texto cifrado) y el contexto de cifrado que identifica la tabla y la cuenta. AWS AWS KMS obtiene el ID de la clave KMS a partir del texto cifrado.

```
{
    "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"
}
```

Encrypt

Cuando cambias la clave de KMS asociada a un secreto, Secrets Manager envía una solicitud de cifrado a para volver AWS KMS a cifrar las versiones `AWSCURRENTAWSPREVIOUS`, y del `AWSPENDING` secreto con la nueva clave. Cuando replica un secreto en otra región, Secrets Manager también envía una solicitud Encrypt a AWS KMS.

El evento que registra la operación `Encrypt` es similar al siguiente evento de ejemplo. El usuario es el principal de su AWS cuenta que accede a la tabla.

```
{
    "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"
}
```


Seguridad de la infraestructura en AWS Secrets Manager

Como se trata de un servicio administrado, AWS Secrets Manager está protegido por la seguridad de red global de AWS. Para obtener información sobre los servicios de seguridad de AWS y cómo AWS protege la infraestructura, consulte Seguridad en la nube de AWS. Para diseñar su entorno de AWS con las prácticas recomendadas de seguridad de infraestructura, consulte Protección de la infraestructura en Portal de seguridad de AWS Well-Architected Framework.

El acceso a Secrets Manager mediante la red se realiza a través de las API publicadas de AWS con TLS. Las API de Secrets Manager se pueden invocar desde cualquier ubicación de red. Sin embargo, Secrets Manager admite políticas de acceso basadas en recursos, que pueden incluir restricciones en función de la dirección IP de origen. También puede utilizar las políticas de recursos de Secrets Manager para controlar el acceso a los secretos desde los puntos de conexión de nube privada virtual (VPC) específicos o las VPC específicas. Este proceso aísla de manera efectiva el acceso de red a un secreto determinado solo desde la VPC específica de la red de AWS. Para obtener más información, consulte Punto de conexión VPC.

Resiliencia en AWS Secrets Manager

AWS conforma la infraestructura global en torno a regiones y zonas de disponibilidad de AWS. AWS Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se conectan mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad le permiten tener una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre resiliencia y recuperación de desastres, consulte Pilar de fiabilidad: AWS Well-Architected Framework.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte Infraestructura global de AWS.

TLS postcuántico

Secrets Manager admite una opción híbrida de intercambio de claves poscuánticas para el protocolo de cifrado de red seguridad de la capa de transporte (TLS). Puede utilizar esta opción de TLS cuando se conecte a los puntos de enlace de la API de Secrets Manager. Estamos ofreciendo esta característica antes de que se estandaricen los algoritmos postcuánticos para que pueda comenzar a probar el efecto de estos protocolos de intercambio de claves en las llamadas a Secrets Manager. Estas características opcionales de intercambio híbrido postcuántico de claves son al menos tan seguras como el cifrado TLS que utilizamos hoy en día y es muy probable que aporten beneficios de seguridad adicionales. Sin embargo, afectan a la latencia y a la velocidad si las comparamos con los protocolos clásicos de intercambio de claves que se utilizan hoy en día.

Para proteger los datos cifrados hoy frente a posibles ataques futuros, AWS trabaja con la comunidad criptográfica en el desarrollo de algoritmos resistentes a la informática cuántica o postcuánticos. Hemos implementado conjuntos de cifrado de intercambio híbrido postcuántico de claves en los puntos de enlace de Secrets Manager. Estos conjuntos de cifrado híbridos, que combinan elementos clásicos y postcuánticos, garantizan que su conexión TLS sea al menos tan segura como con los conjuntos clásicos de cifrado. Sin embargo, dado que las características de rendimiento y los requisitos de ancho de banda de los conjuntos de cifrado híbridos son diferentes de los mecanismos clásicos de intercambio de claves, le recomendamos que los pruebe en las llamadas a la API.

Secrets Manager admite PQTLS en todas las regiones excepto las de China.

Para configurar el cifrado TLS postcuántico híbrido

1. Agregue el cliente del tiempo de ejecución común de AWS a sus dependencias de Maven. Le recomendamos que utilice la última versión disponible. Por ejemplo, esta declaración agrega la versión 2.20.0.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>aws-crt-client</artifactId>
  <version>2.20.0</version>
</dependency>
```

2. Agregue el SDK de AWS para Java 2.x al proyecto e inicialícelo. Habilite los conjuntos de cifrado postcuántico híbrido en su cliente HTTP.

```
SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
            .postQuantumTlsEnabled(true)
            .build();
```

3. Cree el cliente asíncrono de Secrets Manager.

```
SecretsManagerAsyncClient SecretsManagerAsync = SecretsManagerAsyncClient.builder()
            .httpClient(awsCrtHttpClient)
            .build();
```

   Ahora, cuando llama a las operaciones de la API de Secrets Manager, las llamadas se transmiten al punto de conexión de Secrets Manager mediante TLS postcuántico híbrido.

Para obtener más información acerca del uso de TLS postcuántico, consulte:

- AWS SDK for Java 2.x Guía para desarrolladores y la AWS SDK for Java 2.xentrada de blog publicada.
- Presentación de s2n-tls, una nueva implementación de TLS de código abierto y usandos2n-tls.
- Criptografía postcuántica en el Instituto Nacional de Normalización y Tecnología (NIST).
- Métodos híbridos de encapsulación de claves postcuánticas (PQ KEM) para la capa de seguridad de transporte 1.2 (TLS)..

El TLS postcuántico para Secrets Manager está disponible en todas las Regiones de AWS excepto China.

Solución de problemas de AWS Secrets Manager

Utilice la información que se indica aquí para diagnosticar y solucionar los problemas que puedan surgir cuando trabaje con Secrets Manager.

Para conocer los problemas relacionados con la rotación, consulte the section called “Solución de problemas de rotación de ”.

Temas

Mensajes de “Acceso denegado” cuando envía solicitudes a Secrets Manager

Compruebe que tiene permisos para llamar a la operación y a los recursos que ha solicitado. Un administrador debe conceder permisos asociando una política de IAM a su usuario de IAM o a un grupo del que sea miembro. Si las instrucciones de la política que conceden esos permisos incluyen alguna condición, como la hora del día o restricciones de direcciones IP, también debe cumplir esos requisitos cuando envíe la solicitud. Para obtener más información sobre cómo consultar o modificar políticas para un usuario, grupo o rol de IAM, consulte Trabajar con políticas en la Guía del usuario de IAM. Para obtener más información sobre los permisos necesarios para Secrets Manager, consulte Autenticación y control de acceso.

Si va a firmar las solicitudes a la API manualmente, sin usar los SDK de AWS, compruebe que haya firmado la solicitud correctamente.

“Acceso denegado” para credenciales de seguridad temporales

Compruebe que el usuario o rol de IAM que está utilizando para realizar la solicitud tiene los permisos adecuados. Los permisos de credenciales de seguridad temporales se obtienen de un usuario o un rol de IAM. Esto significa que los permisos están limitados a los que se conceden al usuario o al rol de IAM. Para obtener más información sobre cómo se determinan los permisos de las credenciales de seguridad temporales, consulte Controlar los permisos para credenciales de seguridad tempolrales en la Guía del usuario de IAM.

Compruebe que las solicitudes se han firmado correctamente y que la solicitud tiene el formato correcto. Para obtener más detalles, consulte la documentación del conjunto de herramientas del SDK seleccionado o Uso de credenciales de seguridad temporales para solicitar acceso a los recursos de AWS en la Guía del usuario de IAM.

Compruebe que sus credenciales de seguridad temporales no hayan caducado. Para obtener más información, consulte Solicitud de credenciales de seguridad temporales en la Guía del usuario de IAM.

Para obtener más información sobre los permisos necesarios para Secrets Manager, consulte Autenticación y control de acceso.

Los cambios que realizo no están siempre visibles inmediatamente.

Secrets Manager utiliza un modelo de computación distribuida denominado coherencia final. Cualquier cambio que realice en Secrets Manager (o en otros servicios de AWS) tardará en aparecer en todos los puntos de enlace posibles. Este retraso se debe en parte al tiempo que se tarda en enviar los datos de un servidor a otro, de una zona de replicación a otra y entre regiones de todo el mundo. Secrets Manager también utiliza la caché para mejorar el rendimiento, pero en algunos casos esto puede agregar tiempo. Es posible que el cambio no sea visible hasta que se agoten los datos previamente almacenados.

Diseñe sus aplicaciones globales teniendo en cuenta estos posibles retrasos. Además, asegúrese de que funcionan según lo previsto, incluso cuando un cambio realizado en una ubicación no sea visible inmediatamente en otra.

Para obtener más información sobre cómo otros servicios de AWS se ven afectados por la coherencia final, consulte:

- Administración de la consistencia de los datos en la Guía para desarrolladores de bases de datos de Amazon Redshift
- Modelo de consistencia de datos de Amazon S3 en la Guía del usuario de Amazon Simple Storage Service
- Asegurar la consistencia al usar Amazon S3 y Amazon EMR para ETL Workflows en el blog de big data de AWS
- Consistencia final de Amazon EC2 de Amazon en la Referencia de la API de Amazon EC2

Cuando creo un secreto, recibo el mensaje “No se puede generar una clave de datos con una clave KMS asimétrica”.

Secrets Manager utiliza una clave KMS de cifrado simétrica asociada con un secreto para generar una clave de datos para cada valor de secreto. No puede utilizar una clave KMS asimétrica. Compruebe que está utilizando una clave KMS de cifrado simétrica en lugar de una clave KMS asimétrica. Para obtener instrucciones, consulte Identificar clave KMS simétricas y asimétricas.

Una operación de la AWS CLI o de AWS SDK no puede encontrar mi secreto a partir de un ARN parcial

En muchos casos, Secrets Manager puede encontrar un secreto utilizando parte de un ARN en lugar del ARN completo. No obstante, si el nombre de su secreto termina en un guion seguido de seis caracteres, es posible que Secrets Manager no pueda encontrar el secreto solo con parte de un ARN. En lugar de ello, recomendamos que utilice el ARN completo o el nombre del secreto.

Más información

Secrets Manager incluye seis caracteres de asignación al azar al final del nombre del secreto para garantizar que el ARN del secreto sea único. Si se elimina el secreto original y, a continuación, se crea un secreto nuevo con el mismo nombre, ambos tendrán ARN diferentes debido a estos caracteres. Los usuarios con acceso al secreto anterior no tienen acceso automático al secreto nuevo porque los ARN son diferentes.

Secrets Manager crea un ARN para un secreto con la región, la cuenta, el nombre del secreto y, a continuación, un guion y seis caracteres más, de la siguiente manera:

```
arn:aws:secretsmanager:us-east-2:111122223333:secret:SecretName-abcdef
```

Si el nombre del secreto termina con un guion y seis caracteres, y se utiliza solo una parte del ARN, a Secrets Manager le puede parecer que se está especificando un ARN completo. Por ejemplo, es posible que tenga un secreto denominado `MySecret-abcdef` con el ARN

`arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret-abcdef-nutBrk`

Si llama a la siguiente operación, que solo utiliza parte del ARN del secreto, es posible que Secrets Manager no encuentre el secreto.

```
$ aws secretsmanager describe-secret --secret-id arn:aws:secretsmanager:us-
east-2:111122223333:secret:MySecret-abcdef
```

Este secreto está gestionado por un servicio de AWS, por lo que es necesario utilizar ese servicio de para actualizarlo.

Si aparece este mensaje al intentar modificar un secreto, el secreto solo se puede actualizar mediante el servicio de administración que aparece en el mensaje. Para obtener más información, consulte Secretos gestionados por otros servicios.

Para determinar quién administra un secreto, puede revisar el nombre del secreto. Los secretos gestionados por otros servicios llevan el prefijo ID de ese servicio. O bien, en el AWS CLI, llame a describe-secret y, a continuación, revise el campo `OwningService`.

Cuotas de AWS Secrets Manager

Las API de lectura de Secrets Manager tienen cuotas TPS altas y las API de plano de control que se llaman con menos frecuencia tienen cuotas TPS más bajas. Le recomendamos que evite llamar a `PutSecretValue` or `UpdateSecret` a un ritmo sostenido de más de una vez cada 10 minutos. Cuando llama a `PutSecretValue` o `UpdateSecret` para actualizar el valor del secreto, Secrets Manager crea una nueva versión del secreto. Secrets Manager elimina versiones sin etiquetar cuando hay más de 100, pero no elimina versiones creadas hace menos de 24 horas. Si actualiza el valor secreto más de una vez cada 10 minutos, crea más versiones de las que elimina Secrets Manager y alcanzará la cuota para las versiones de secretos.

Puede operar varias regiones en su cuenta, y cada cuota es específica para cada región.

Cuando una aplicación de una Cuenta de AWS utiliza un secreto que es propiedad de una cuenta diferente, este proceso se denomina solicitud entre cuentas. En el caso de las solicitudes entre cuentas, Secrets Manager limita de forma controlada la cuenta de la identidad que realiza las solicitudes, no la cuenta que es propietaria del secreto. Por ejemplo, si una identidad de la cuenta A utiliza un secreto en la cuenta B, el uso del secreto solo se aplica a las cuotas de la cuenta A.

Cuotas de Secrets Manager

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa combinada de solicitudes de API DeleteResourcePolicy, GetResourcePolicy, PutResourcePolicy y ValidateResourcePolicy	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API DeleteResourcePolicy, GetResourcePolicy, PutResourcePolicy y ValidateResourcePolicy combinadas.
Tasa combinada de solicitudes de API DescribeSecret y GetSecretValue	Cada región admitida: 10 000 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API DescribeSecret

Nombre	Valor predeterminado	Ajusta	Descripción
			y GetSecretValue combinadas.
Tasa combinada de solicitudes de API PutSecretValue, RemoveRegionsFromReplication, ReplicateSecretToRegion, StopReplicationToReplica, UpdateSecret y UpdateSecretVersionStage	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API PutSecretValue, RemoveRegionsFromReplication, ReplicateSecretToRegion, StopReplicationToReplica, UpdateSecret y UpdateSecretVersionStage combinadas.
Tasa combinada de solicitudes de API RestoreSecret	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API RestoreSecret.
Tasa combinada de solicitudes de API RotateSecret y CancelRotateSecret	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API RotateSecret y CancelRotateSecret combinadas.
Tasa combinada de solicitudes de API TagResource y UntagResource	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API TagResource y UntagResource combinadas.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes de API BatchGetSecretValue	Cada región admitida: 100 por segundo	No	Máximo de transacciones por segundo para las solicitudes de la API BatchGetSecretValue.
Tasa de solicitudes de API CreateSecret	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API CreateSecret.
Tasa de solicitudes de API DeleteSecret	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API DeleteSecret.
Tasa de solicitudes de API GetRandomPassword	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API GetRandomPassword.
Tasa de solicitudes de API ListSecretVersionIds	Cada región admitida: 50 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API ListSecretVersionIds.
Tasa de solicitudes de API ListSecrets	Cada región admitida: 100 por segundo	No	Máximo de transacciones por segundo para las solicitudes de API ListSecrets.
Longitud de política basada en recursos	Cada región admitida: 20 480	No	Número máximo de caracteres de una política de permisos basada en recursos asociada a un secreto.

Nombre	Valor predeterminado	Ajusta	Descripción
Tamaño del valor de secreto	Cada región admitida: 65 536 bytes	No	Tamaño máximo de un valor de secreto cifrado. Si el valor de secreto es una cadena, entonces este es el número de caracteres permitido en el valor de secreto.
Secretos	Cada región admitida: 500 000	No	Número máximo de secretos de cada región de AWS de esta cuenta de AWS.
Etiquetas provisionales adjuntas en todas las versiones de un secreto	Cada región admitida: 20	No	Número máximo de etiquetas provisionales asociadas a todas las versiones de un secreto.
Versiones por secreto	Cada región admitida: 100	No	Número máximo de versiones de un secreto.

Agregar reintentos a su aplicación

Su cliente de AWS puede ver que las llamadas a Secrets Manager fallan debido a problemas inesperados en el lado del cliente. O bien las llamadas pueden fallar debido a la limitación de velocidad de Secrets Manager. Cuando supera una cuota de solicitud de API, Secrets Manager realiza una limitación controlada de la solicitud. Rechaza una solicitud que de otro modo sería válida y devuelve un error de throttling. Para ambos tipos de fallos, recomendamos volver a intentar la llamada después de un breve periodo de espera. Esto se denomina estrategia de retroceso y reintento.

Es posible que desee agregar reintentos al código de la aplicación si experimenta los siguientes errores:

Excepciones y errores transitorios

- `RequestTimeout`
- `RequestTimeoutException`
- `PriorRequestNotComplete`
- `ConnectionError`
- `HTTPClientError`

Limitación controlada y limitación de errores y excepciones en el lado del servicio

- `Throttling`
- `ThrottlingException`
- `ThrottledException`
- `RequestThrottledException`
- `TooManyRequestsException`
- `ProvisionedThroughputExceededException`
- `TransactionInProgressException`
- `RequestLimitExceeded`
- `BandwidthLimitExceeded`
- `LimitExceededException`
- `RequestThrottled`
- `SlowDown`

Para obtener más información, así como código de ejemplo, sobre reintentos, retroceso exponencial y fluctuación, consulte los siguientes recursos:

- Retroceso exponencial y fluctuación
- Tiempos de espera, reintentos y retroceso con fluctuación
- Reintentos de error y retroceso exponencial en AWS.

Historial del documento

En la siguiente tabla se describen los cambios importantes en la documentación desde la última versión de AWS Secrets Manager. Para recibir notificaciones sobre los cambios en esta documentación, puede suscribirse a una fuente RSS.

Cambio	Descripción	Fecha
Cambio de Secrets Manager a una política AWS gestionada	La política `SecretsManagerReadWrite` gestionada ahora incluye `redshift-serverless` permisos. Para obtener más información, consulte la política AWS gestionada para AWS Secrets Manager	12 de marzo de 2024

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes en cada versión de la Guía del AWS Secrets Manager usuario antes de febrero de 2024.

Cambio	Descripción	Fecha
Disponibilidad general	Esta es la primera versión pública de Secrets Manager.	4 de abril de 2018

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.