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 administrar, recuperar y rotar las credenciales de las bases de datos, las credenciales de las aplicaciones, OAuth los tokens, las claves de API y otros 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: se recomienda 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 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:

- Administrar secretos
- Control del acceso a sus secretos
- Obtener secretos
- Rotar secretos de
- Monitorear secretos
- Supervisión 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 de AWS Secrets Manager. Para controlar sus costos, consulte the section called “Monitoreo de los costos de 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 de AWS 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 obtener más información, consulte Precios de AWS CloudTrail.

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 SDKs
- 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 “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos 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 SDKs

AWS SDKs Constan de bibliotecas y códigos de muestra para varios lenguajes de programación y plataformas. SDKs Incluyen tareas como la firma criptográfica de las solicitudes, la gestión de errores y el reintento automático de las solicitudes. Para descargar e instalar cualquiera de ellas SDKs, consulte Herramientas para Amazon Web Services.

AWS SDKs Utilizan automáticamente el punto de enlace 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 puedes realizar llamadas directas a la API de consulta HTTPS de Secrets Manager, te recomendamos que utilices una de ellas SDKs en su lugar. El SDK realiza muchas tareas de gran utilidad que, de otro modo, tendría que realizar de forma manual. Por ejemplo, firman SDKs automáticamente sus solicitudes y convierten las respuestas en una estructura sintácticamente adecuada a su idioma.

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. Los puntos finales de Secrets Manager son puntos de enlace de doble pila, lo que significa que admiten tanto como. IPv4 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 IPv6 y, luego, IPv4 en secuencia, por lo que si no lo tiene IPv6 habilitado, puede pasar algún tiempo antes de que se agote el tiempo de espera de la llamada y vuelva a IPv4 intentarlo. Para solucionar este problema, puedes deshabilitarlo 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. Para obtener información sobre el uso de direcciones de doble pila en Secrets Manager, consulteIPv4 y IPv6 acceso.

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.	us-east-1	secretsmanager.us-east-1.amazonaws.com	HTTPS HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo
(Norte de Virginia)		secretsmanager-fips.us-east-1.amazonaws.com	
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

Nombre de la región	Región	Punto de conexión	Protocolo	
Asia-Pacífico (Malasia)	ap-southeast-5	secretsmanager.ap-southeast-5.amazonaws.com	HTTPS	
Asia-Pacífico (Melbourne)	ap-southeast-4	secretsmanager.ap-southeast-4.amazonaws.com	HTTPS	
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 (Tailandia)	ap-southeast-7	secretsmanager.ap-southeast-7.amazonaws.com	HTTPS	

Nombre de la región	Región	Punto de conexión	Protocolo
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
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

Nombre de la región	Región	Punto de conexión	Protocolo
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
México (central)	mx-central-1	secretsmanager.mx-central-1.amazonaws.com	HTTPS
Medio Oriente (Baréin)	me-south-1	secretsmanager.me-south-1.amazonaws.com	HTTPS
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

Nombre de la región	Región	Punto de conexión	Protocolo	
AWS GovCloud (Estados Unidos-Oeste)	us-gov-west-1	secretsmanager.us-gov-west-1.amazonaws.com secretsmanager-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS	

AWS Secrets Manager mejores prácticas

Secrets Manager proporciona un número de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no suponen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, plantéeselas como consideraciones útiles en lugar de como normas.

Tenga en cuenta las siguientes prácticas recomendadas para almacenar y administrar secretos:

- Guarde las credenciales y otra información confidencial en AWS Secrets Manager
- Encontrar secretos sin protección en su código
- Elija una clave de cifrado para su secreto
- Utilice el almacenamiento en caché para recuperar los secretos
- Rotar sus secretos de
- Mitigar los riesgos del uso de la CLI
- Limitar el acceso a los secretos
- Replicar secretos
- Monitorear secretos
- Ejecute su infraestructura en redes privadas

Guarde las credenciales y otra información confidencial en AWS Secrets Manager

Secrets Manager puede ayudarle a mejorar su posición de seguridad y el cumplimiento, y reducir el riesgo de acceso no autorizado a su información confidencial. Secrets Manager cifra los secretos en reposo mediante claves de cifrado que usted posee y almacena en AWS Key Management Service (AWS KMS). Al recuperar un secreto, Secrets Manager lo descifra y lo transmite de forma segura a través de TLS a su entorno local. Para obtener más información, consulte Crear secretos.

Encontrar secretos sin protección en su código

CodeGuru Reviewer se integra con Secrets Manager para usar un detector de secretos que encuentra secretos desprotegidos en el código. El detector de secretos busca contraseñas

codificadas, cadenas de conexión a bases de datos, nombres de usuario y mucho más. Para obtener más información, consulte the section called “ CodeGuru Revisor de Amazon”.

Amazon Q puede escanear su base de código en busca de vulnerabilidades de seguridad y problemas de calidad del código para mejorar el estado de sus aplicaciones a lo largo del ciclo de desarrollo. Para obtener más información, consulte Escaneo del código con Amazon Q en la Guía del usuario de Amazon Q Developer.

Elija una clave de cifrado para su secreto

En la mayoría de los casos, recomendamos usar la clave `aws/secretsmanager` AWS administrada para cifrar los secretos. No se aplica ningún cargo por su uso.

Para poder acceder a un secreto desde otra cuenta o aplicar una política de claves a la clave de cifrado, utilice una clave administrada por el cliente para cifrar el secreto.

- En la política de claves, asigne el valor `secretsmanager.<region>.amazonaws.com` a la clave de `kms:ViaService`condición. Esto limita el uso de la clave solo a las solicitudes de Secrets Manager.
- Para limitar aún más el uso de la clave solo a las solicitudes de Secrets Manager con el contexto correcto, utilice las claves o valores del contexto de cifrado de Secrets Manager como condición a fin de utilizar la clave de KMS creando lo siguiente:
  - Un operador de condición de cadena en una política de claves o de IAM
  - Una restricción de la concesión en una concesión

Para obtener más información, consulte the section called “Cifrado y descifrado de secretos”.

Utilice el almacenamiento en caché para recuperar los secretos

Para utilizar sus secretos de la manera más eficaz, le recomendamos que use uno de los siguientes componentes de almacenamiento en caché de Secrets Manager compatibles para almacenar en caché sus secretos y actualizarlos solo cuando sea necesario:

- Java con almacenamiento en caché del cliente
- Python con almacenamiento en caché del cliente
- .NET con almacenamiento en caché del cliente

- Go con almacenamiento en caché del cliente
- Rust con almacenamiento en caché del cliente
- AWS Parámetros y secretos de la extensión Lambda
- the section called “Amazon EKS”
- Úselo the section called “Agente de Secrets Manager” para estandarizar el consumo de información confidencial de Secrets Manager en entornos como AWS Lambda Amazon Elastic Container Service, Amazon Elastic Kubernetes Service y Amazon Elastic Compute Cloud.

Rotar sus secretos de

Si no cambia sus secretos durante un largo período de tiempo, los secretos se vuelven más propensos a ser comprometidos. Con Secrets Manager, puede configurar la rotación automática con una frecuencia máxima de cuatro horas. Secrets Manager ofrece dos estrategias de rotación: Un solo usuario y Usuarios alternativos. Para obtener más información, consulte Rotar secretos de .

Mitigar los riesgos del uso de la CLI

Cuando se utiliza AWS CLI para invocar AWS operaciones, se introducen esos comandos en una consola de comandos. La mayoría de los intérpretes de comandos ofrecen características que podrían comprometer sus secretos, como el registro y la posibilidad de ver el último comando introducido. Antes de utilizar la AWS CLI para introducir información confidencial, asegúrese de the section called “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos AWS Secrets Manager”.

Limitar el acceso a los secretos

En las declaraciones de política de IAM que controlan el acceso a sus secretos, utilice el principio de acceso de privilegio mínimo. Puede utilizar los roles y políticas de IAM, las políticas de recursos y el control de acceso basado en atributos (ABAC). Para obtener más información, consulte the section called “Autenticación y control de acceso”.

Temas

- Bloquear el acceso amplio a los secretos
- Tener cuidado con las condiciones de dirección IP en las políticas

- Limitar solicitudes con condiciones del punto de conexión de VPC

Bloquear el acceso amplio a los secretos

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 sobre Zelkova, consulte Cómo utilizar el AWS razonamiento automatizado para ayudarle a lograr una seguridad a gran escala en el blog de AWS seguridad.

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"
            }
        }
    }
}
```

Tener cuidado con las condiciones de dirección IP en las políticas

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 adjuntas a un secreto una política que restringe AWS las acciones a las solicitudes del rango de direcciones IP de la red corporativa, tus solicitudes como usuario de IAM que invocan la solicitud de la red corporativa funcionan según lo esperado. Sin embargo, si habilita otros servicios para que accedan al secreto en su nombre, por ejemplo, cuando habilita la rotación con una función Lambda, esa función llama a las operaciones de Secrets Manager desde un espacio AWS de direcciones interno. 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 “Limitar solicitudes con condiciones del punto de conexión de VPC”.

Limitar solicitudes con condiciones del punto de conexión de 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 VPCs”.

- `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 AWS servicios se pueden ejecutar con un punto final 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 the section called “Puntos de conexión de VPC (AWS PrivateLink)”.

Replicar secretos

Secrets Manager puede replicar automáticamente sus datos secretos en varias AWS regiones para cumplir con sus requisitos de resiliencia o recuperación ante desastres. Para obtener más información, consulte Replicación multirregional.

Monitorear secretos

Secrets Manager le permite auditar y supervisar los secretos mediante la integración con los servicios de AWS registro, supervisión y notificación. Para obtener más información, consulte:

- the section called “Inicia sesión con AWS CloudTrail ”

- the section called “Supervise con CloudWatch”
- the section called “Supervisión de secretos para la conformidad”
- the section called “Monitoreo de los costos de Secrets Manager”
- the section called “Detecte amenazas con GuardDuty”

Ejecute su infraestructura en redes privadas

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. Para obtener más información, consulte the section called “Puntos de conexión de VPC (AWS PrivateLink)”.

AWS Secrets Manager tutoriales

Temas

Encuentre secretos desprotegidos en su código con Amazon Reviewer CodeGuru

Amazon CodeGuru Reviewer es un servicio que utiliza el análisis de programas y el aprendizaje automático para detectar posibles defectos difíciles de encontrar para los desarrolladores y ofrece sugerencias para mejorar el código de Java y Python. CodeGuru Reviewer se integra con Secrets Manager para encontrar secretos desprotegidos en tu código. Para conocer los tipos de secretos que puede encontrar, consulte Tipos de secretos detectados por CodeGuru Reviewer en la Guía del usuario de Amazon CodeGuru Reviewer.

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 ”

Mueva los 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 usar el secreto en tiempo de ejecución, por ejemplo, en este tutorial que usas*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 a seguir a continuación”.

Pasos:

- Paso 1: Crear el secreto
- Paso 2: Actualización del código
- Paso 3: Actualizar el secreto
- Pasos a seguir a continuación

Paso 1: Crear el secreto

El primer paso es copiar el secreto codificado existente en Secrets Manager. Si el secreto está relacionado con un AWS recurso, 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 Almacenar un secreto nuevo.
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:

      API key

         Introducirlo como pares clave/valor:

ClientID : *my_client_id*

ClientSecret : *wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

OAuth token

Introducirlo como texto no cifrado:

AKIAI44QH8DHBEXAMPLE

Digital certificate

Introducirlo como texto no cifrado:

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

Private key

Introducirlo como texto no cifrado:

```
---- 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 *RoleToRetrieveSecretAtRuntime* permite recuperar el secreto, y luego elija 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. 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 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 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 en https://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 a seguir a continuación

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 Amazon CodeGuru Reviewer.
- Puede mejorar el rendimiento y reducir los costos almacenando secretos en caché. Para obtener más información, consulte Obtener 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 Replicación multirregional.
- En este tutorial, *RoleToRetrieveSecretAtRuntime* solo concedió 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 “Políticas basadas en recursos”.
- En este tutorial, otorgaste el permiso *RoleToRetrieveSecretAtRuntime* mediante la política de recursos del secreto. Para ver otras formas de conceder permiso, consulte the section called “Políticas basadas en 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
- Pasos a seguir a continuación

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.

Creación de un secreto

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:
      - Bases de datos de Amazon RDS
      - Base de datos de Amazon DocumentDB
      - Almacenamiento 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 *RoleToRetrieveSecretAtRuntime* permite recuperar el secreto, y luego elija 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 obtener instrucciones, consulte the section called "Puntos de conexión de VPC (AWS PrivateLink)".

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 añade»SecretsManager"al principio del nombre de la función.
    e. Para la estrategia de rotación, elija un solo usuario.
    f. Seleccione Save.

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, consulta Solución de problemas de rotación los CloudWatch registros para ver la función de rotación.

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

Pasos a seguir a continuación

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 Obtener secretos.
- Puede elegir un programa de rotación diferente. Para obtener más información, consulte the section called “Programación de rotación”.
- 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 EC2 instancia de Amazon 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 AWS Lambda función 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
- Requisitos previos
- Paso 1: cree un usuario de base de datos de Amazon RDS
- Paso 2: cree un secreto para las credenciales del usuario
- Paso 3: pruebe el secreto rotado
- Paso 4: limpie los recursos
- Pasos a seguir a continuación

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 the section called “Autenticación y control de acceso”.

Requisitos previos

Para este tutorial, necesita lo siguiente:

- Requisito previo A: Amazon VPC
- Prerrequisito 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 computadora local se conecte a la instancia EC2

Requisito previo A: Amazon VPC

En este paso, crea una VPC en la que puede 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 enlace dentro de la VPC, se asegura de que las llamadas desde la función Lambda a Secrets Manager y Amazon RDS no abandonen la infraestructura. 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 Creación de 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 Creación de 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.

Prerrequisito 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 EC2 instancia para un host bastión

1. Abre la EC2 consola de Amazon 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 Iniciar 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, selecciona Administrar credenciales maestras en AWS Secrets Manager.
5. En la sección Conectividad, en Recurso informático, elija Conectarse a un recurso EC2 informático y, a continuación, en EC2 Instance, elija**`SecretsManagerTutorialInstance`**.
6. Elija Creación de base de datos.

Requisito previo D: Permita que su computadora local se conecte a la instancia EC2

En este paso, configura la EC2 instancia 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 el ordenador local se conecte a la instancia EC2

1. Abre la EC2 consola de Amazon en https://console.aws.amazon.com/ec2/.
2. En la EC2 instancia SecretsManagerTutorialInstance, en la pestaña Seguridad, en Grupos de seguridad, seleccione**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. Para SSH Hostname, introduce la dirección IP pública de la instancia de Amazon EC2 .

Para encontrar la dirección IP en la EC2 consola de Amazon, selecciona la instancia SecretsManagerTutorialInstance. Copia la dirección IP en Public IPv4 DNS.

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

iii. Para SSH Keyfile, elija el archivo de pares 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 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). La función de rotación prueba el secreto actualizado mediante SELECT, por lo que **`appuser`** debe tener ese privilegio como mínimo.

```
CREATE DATABASE myDB;
CREATE USER 'appuser'@'%' IDENTIFIED BY 'EXAMPLE-PASSWORD';
```

```
GRANT SELECT 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 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 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 se utiliza CloudFormation para crear recursos como la función de rotación de Lambda y un rol de ejecución que ejecuta la función Lambda. Cuando CloudFormation termine, el cartel pasará a ser Secreto y su rotación está programada. 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 en https://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 SecretsManagerTutorialy, a continuación, seleccione 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 EC2 instancia que tiene acceso a Internet, elimine los siguientes recursos que creó en este tutorial y sus 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.
- EC2 Instancia de Amazon. Para obtener instrucciones, consulta Terminar una instancia en la Guía del EC2 usuario de Amazon.
- 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 a seguir a continuación

- 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 AWS Lambda función 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 the section called “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 SecretsManagerTutorialy, a continuación, seleccione 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). La función de rotación prueba el secreto actualizado mediante SELECT, por lo que **`dbuser`** debe tener ese privilegio como mínimo.

```
CREATE USER 'dbuser'@'%' IDENTIFIED BY 'EXAMPLE-PASSWORD';
GRANT SELECT 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 rota el secreto, lo que significa que la contraseña se genera mediante programación; ningún humano 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 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 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 se utiliza CloudFormation para crear recursos como la función de rotación de Lambda y un rol de ejecución que ejecuta la función Lambda. Cuando CloudFormation termine, el cartel pasará a ser Secreto y su rotación está programada. 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 en https://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 SecretsManagerTutorialy, a continuación, seleccione 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 a seguir a continuación

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

Crea un AWS Secrets Manager secreto

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

> **Tip**
>
> Para las credenciales del usuario administrador de Amazon RDS y Amazon Redshift, se recomienda utilizar secretos administrados. El secreto gestionado se crea a través del servicio de gestión y, a continuación, se puede utilizar la rotación gestionada.

Cuando se usa la consola para almacenar 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 la política SecretsManagerReadWrite administrada.

Secrets Manager genera una entrada de CloudTrail registro al crear un secreto. Para obtener más información, consulte the section called “Inicia 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 Elegir tipo de secreto, haga lo siguiente:

    a. En Secret type (Tipo de secreto), haga una de estas cosas:

    - Para almacenar credenciales de base de datos, elija el tipo de credenciales de base de datos que desea almacenar. A continuación, elija la Base de datos y, luego, introduzca las Credenciales.
    - Para almacenar claves de API, tokens de acceso y credenciales que no son para bases de datos, elija Otro tipo de secreto.

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. Presentamos algunos ejemplos:

API key

Introducirlo como pares clave/valor:

`ClientID` : *`my_client_id`*

`ClientSecret` : *`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`*

OAuth token

Introducirlo como texto no cifrado:

`AKIAI44QH8DHBEXAMPLE`

Digital certificate

Introducirlo como texto no cifrado:

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

Private key

Introducirlo como texto no cifrado:

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

b. 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”.

c. 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 secretos pueden contener de 1 a 512 caracteres alfanuméricos y los caracteres /_+ =.@-.

   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 “Políticas basadas en recursos”.

   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 Replicación multirregional.

   e. Elija 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 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 “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos AWS Secrets Manager”.

Example Crear un secreto a partir de credenciales de base de datos en 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"
}
```

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\"}"
```

AWS SDK

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

¿Qué hay en un secreto de Secrets Manager?

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"
}
```

Si desea activar la rotación automática para un secreto de base de datos, este debe contener la información de conexión a la base de datos en la estructura JSON correcta. Para obtener más información, consulte the section called “Estructura JSON de un secreto”.

Metadatos

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, los dos secretos son diferentes ARNs debido a estos caracteres. Los usuarios con acceso al secreto anterior no acceden automáticamente al secreto nuevo porque ARNs son diferentes.
- El nombre del secreto, una descripción, una política de recursos y las etiquetas.
- El ARN de una clave de cifrado, AWS KMS key que Secrets Manager utiliza para cifrar y descifrar el valor 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 Rotar secretos de .

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 para 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 “Versiones de un secreto”.

Puedes usar un secreto entre varios Regiones de AWS si lo replicas. 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 Replicación multirregional.

Consulte Administrar secretos.

Versiones de un secreto

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:

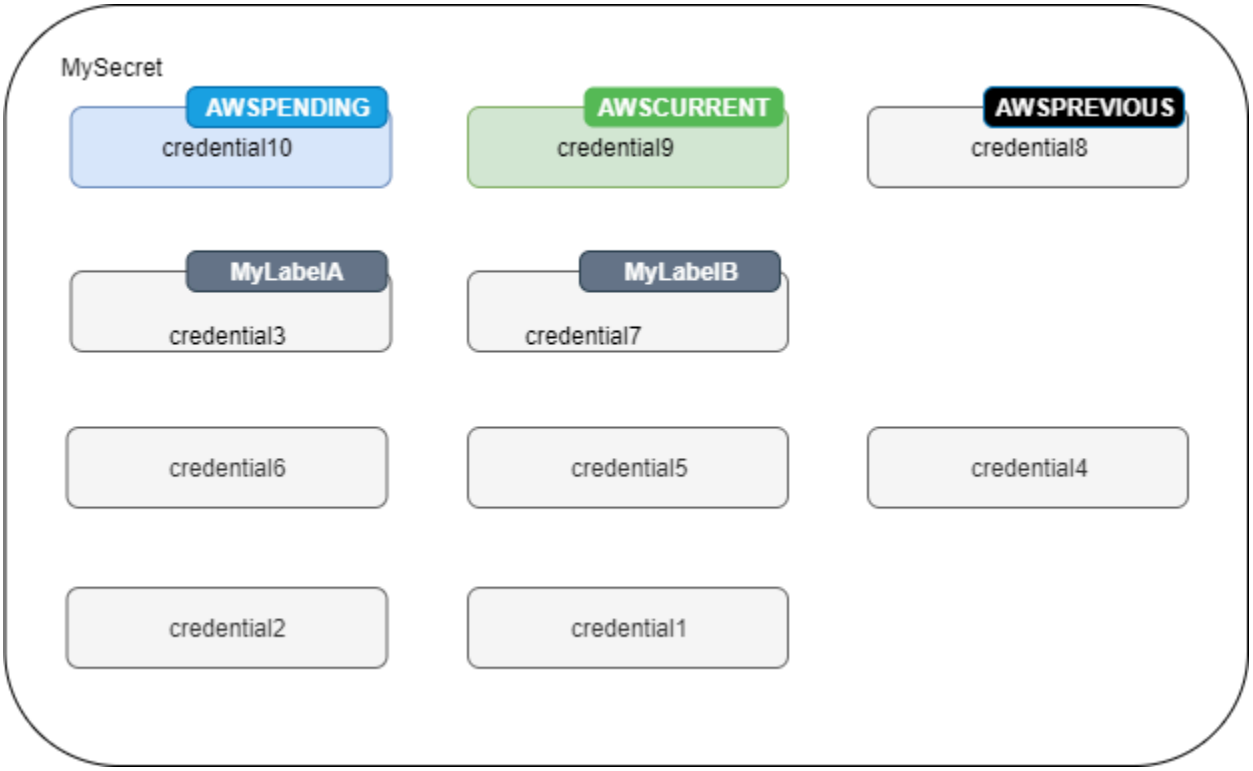
- La versión actual: `AWSCURRENT`
- La versión anterior — `AWSPREVIOUS`
- La versión pendiente (durante la rotación): `AWSPENDING`

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.

También puede etiquetar las versiones con sus propias etiquetas llamando `update-secret-version-stage`al 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 figura se muestra un secreto que contiene versiones AWS etiquetadas y versiones etiquetadas por el cliente. Las versiones sin etiquetas se consideran obsoletas y Secrets Manager las eliminará en algún momento.



Estructura JSON de AWS Secrets Manager secretos

En un secreto de Secrets Manager, puede almacenar cualquier texto o binario con un tamaño máximo de 65 536 bytes.

Si usa the section called “Rotación con función de Lambda”, un secreto debe contener los campos JSON específicos que la función de rotación espera. Por ejemplo, en el caso de un secreto que contiene credenciales de base de datos, la función de rotación se conecta a la base de datos para actualizar las credenciales, por lo que el secreto debe contener la información de conexión a la base de datos.

Si utiliza la consola para editar la rotación de un secreto de base de datos, el secreto debe contener pares clave-valor JSON específicos que identifiquen la base de datos. Secrets Manager utiliza estos campos para consultar la base de datos y encontrar la VPC correcta para almacenar una función de rotación.

Los nombres de clave JSON distinguen entre mayúsculas y minúsculas.

Temas

Credenciales de Amazon RDS y Aurora

Para utilizar las plantillas de funciones de rotación que proporciona Secrets Manager, utilice la siguiente estructura JSON. Puede agregar más pares clave/valor; por ejemplo, para contener información de conexión de bases de datos de réplicas de otras regiones.

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": "<ARN of the elevated secret>",
  "dbInstanceIdentifier": <optional: ID of the instance. Alternately, use
 dbClusterIdentifier. Required for configuring rotation in the console.>",
  "dbClusterIdentifier": <optional: ID of the cluster. Alternately, use
 dbInstanceIdentifier. Required for configuring rotation in the console.>"
}
```

MariaDB

```
{
```

```
  "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": "<optional: ARN of the elevated secret. Required for the the section called “Usuarios alternativos”.>",
  "dbInstanceIdentifier": <optional: ID of the instance. Alternately, use dbClusterIdentifier. Required for configuring rotation in the console.>",
  "dbClusterIdentifier": <optional: ID of the cluster. Alternately, use dbInstanceIdentifier. Required for configuring rotation in the console.>"
}
```

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>,
  "masterarn": "<optional: ARN of the elevated secret. Required for the the section called “Usuarios alternativos”.>",
  "dbInstanceIdentifier": <optional: ID of the instance. Alternately, use dbClusterIdentifier. Required for configuring rotation in the console.>",
  "dbClusterIdentifier": <optional: ID of the cluster. Alternately, use dbInstanceIdentifier. Required for configuring rotation in the console.>"
}
```

Oracle

```
{
  "engine": "oracle",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name>",
  "port": <TCP port number. If not specified, defaults to 1521>,
  "masterarn": "<optional: ARN of the elevated secret. Required for the the section called “Usuarios alternativos”.>",
```

```
  "dbInstanceIdentifier": <optional: ID of the instance. Alternately, use
 dbClusterIdentifier. Required for configuring rotation in the console.>",
  "dbClusterIdentifier": <optional: ID of the cluster. Alternately, use
 dbInstanceIdentifier. Required for configuring rotation in the console.>"
}
```

Postgres

```
{
  "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": "<optional: ARN of the elevated secret. Required for the the section
 called “Usuarios alternativos”.>",
  "dbInstanceIdentifier": <optional: ID of the instance. Alternately, use
 dbClusterIdentifier. Required for configuring rotation in the console.>",
  "dbClusterIdentifier": <optional: ID of the cluster. Alternately, use
 dbInstanceIdentifier. Required for configuring rotation in the console.>"
}
```

SQLServer

```
{
  "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": "<optional: ARN of the elevated secret. Required for the the section
 called “Usuarios alternativos”.>",
  "dbInstanceIdentifier": <optional: ID of the instance. Alternately, use
 dbClusterIdentifier. Required for configuring rotation in the console.>",
  "dbClusterIdentifier": <optional: ID of the cluster.Alternately, use
 dbInstanceIdentifier.  Required for configuring rotation in the console.>"
}
```

Credenciales de Amazon Redshift

Para utilizar las plantillas de funciones de rotación que proporciona Secrets Manager, utilice la siguiente estructura JSON. Puede agregar más pares clave/valor; por ejemplo, para contener información de conexión de bases de datos de réplicas de otras regiones.

```
{
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "dbClusterIdentifier": "<optional: database ID. Required for configuring rotation in
 the console.>"
  "port": <optional: TCP port number. If not specified, defaults to 5439>
  "masterarn": "<optional: ARN of the elevated secret. Required for the the section
 called “Usuarios alternativos”.>"
}
```

Credenciales de Amazon Redshift sin servidor

Para utilizar las plantillas de funciones de rotación que proporciona Secrets Manager, utilice la siguiente estructura JSON. Puede agregar más pares clave/valor; por ejemplo, para contener información de conexión de bases de datos de réplicas de otras regiones.

```
{
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "namespaceName": "<optional: namespace name, Required for configuring rotation in the
 console.> "
  "port": <optional: TCP port number. If not specified, defaults to 5439>
  "masterarn": "<optional: ARN of the elevated secret. Required for the the section
 called “Usuarios alternativos”.>"
}
```

Credenciales de Amazon DocumentDB

Para utilizar las plantillas de funciones de rotación que proporciona Secrets Manager, utilice la siguiente estructura JSON. Puede agregar más pares clave/valor; por ejemplo, para contener información de conexión de bases de datos de réplicas de otras regiones.

```
{
  "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>,
  "masterarn": "<optional: ARN of the elevated secret. Required for the the section
 called “Usuarios alternativos”.>",
  "dbClusterIdentifier": "<optional: database cluster ID. Alternately, use
 dbInstanceIdentifier. Required for configuring rotation in the console.>"
  "dbInstanceIdentifier": "<optional: database instance ID. Alternately, use
 dbClusterIdentifier. Required for configuring rotation in the console.>"
}
```

Estructura secreta de Amazon Timestream para InfluxDB

Para rotar los secretos de Timestream, puede utilizar las plantillas de rotación the section called “Amazon Timestream para InfluxDB”.

Para obtener más información, consulte Cómo utiliza los secretos Amazon Timestream para InfluxDB en la Guía para desarrolladores de Amazon Timestream.

Los secretos de Timestream deben estar en la estructura JSON correcta para poder utilizar las plantillas de rotación. Para obtener más información, consulte Qué hay en el secreto en la Guía para desarrolladores de Amazon Timestream.

ElastiCache Credenciales de Amazon

El siguiente ejemplo muestra la estructura JSON de un secreto que almacena las ElastiCache credenciales.

```
{
  "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.

Credenciales de Active Directory

AWS Directory Service usa secretos para almacenar las credenciales de Active Directory. Para obtener más información, consulte Cómo unir sin problemas una instancia de Amazon EC2 Linux a su Active Directory de AD gestionado en la Guía de AWS Directory Service administración. La unión sin problemas de dominios requiere los nombres de claves de los siguientes ejemplos. Si no utiliza la unión de dominios fluida, puede cambiar los nombres de las claves del secreto mediante variables de entorno, tal y como se describe en el código de la plantilla de la función de rotación.

Para rotar los secretos de Active Directory, puede usar las plantillas de rotación de Active Directory.

Active Directory credential

```
{
  "awsSeamlessDomainUsername": "<username>",
  "awsSeamlessDomainPassword": "<password>"
}
```

Si desea rotar el secreto, incluya el ID del directorio del dominio.

```
{
  "awsSeamlessDomainDirectoryId": "d-12345abc6e",
  "awsSeamlessDomainUsername": "<username>",
  "awsSeamlessDomainPassword": "<password>"
}
```

Si el secreto se usa junto con un secreto que contiene un keytab, debes incluir el secreto keytab. ARNs

```
{
  "awsSeamlessDomainDirectoryId": "d-12345abc6e",
  "awsSeamlessDomainUsername": "<username>",
  "awsSeamlessDomainPassword": "<password>",
```

```
  "directoryServiceSecretVersion": 1,
  "schemaVersion": "1.0",
  "keytabArns": [
    "<ARN of child keytab secret 1>,
    "<ARN of child keytab secret 2>,
    "<ARN of child keytab secret 3>,
  ],
  "lastModifiedDateTime": "2021-07-19 17:06:58"
}
```

Active Directory keytab

Para obtener información sobre el uso de archivos keytab para autenticarse en las cuentas de Active Directory en Amazon EC2, consulte Implementación y configuración de la autenticación de Active Directory con SQL Server 2017 en Amazon Linux 2.

```
{
  "awsSeamlessDomainDirectoryId": "d-12345abc6e",
  "schemaVersion": "1.0",
  "name": "< name>",
  "principals": [
    "aduser@MY.EXAMPLE.COM",
    "MSSQLSvc/test:1433@MY.EXAMPLE.COM"
  ],
  "keytabContents": "<keytab>",
  "parentSecretArn": "<ARN of parent secret>",
  "lastModifiedDateTime": "2021-07-19 17:06:58"
  "version": 1
}
```

Gestiona tus secretos con AWS Secrets Manager

Temas

Actualizar el valor de un AWS Secrets Manager secreto

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 “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos 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"
  ```

AWS SDK

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”.

Generar una contraseña con Secrets Manager

Un patrón habitual de uso de Secrets Manager consiste en generar una contraseña en Secrets Manager y, a continuación, utilizarla en la base de datos o el servicio. Para ello, tiene los siguientes métodos:

- AWS CloudFormation — VerAWS CloudFormation.
- AWS CLI — Ver `get-random-password`.
- AWS SDKs — Ver `GetRandomPassword`.

Restaurar un secreto a una versión anterior

Puede revertir un secreto a una versión anterior moviendo las etiquetas adjuntas a las versiones secretas mediante la AWS CLI. Para obtener información sobre cómo Secrets Manager almacena versiones de secretos, consulte the section called “Versiones de un secreto”.

En el siguiente `update-secret-version-stage`ejemplo, se AWSCURRENT mueve la etiqueta provisional a la versión anterior de un secreto, lo que revierte el secreto a la versión anterior. Para encontrar el ID de la versión anterior, use `list-secret-version-ids` o consulte las versiones en la consola de Secrets Manager.

En este ejemplo, la versión con la etiqueta es a1b2c3d4-5678-90ab-cdef- y la versión con la AWSCURRENT etiqueta es a1b2c3d4-5678-90ab-cdef-. EXAMPLE11111 AWSPREVIOUS EXAMPLE22222 En este ejemplo, mueves AWSCURRENT la etiqueta de la versión 11111 a la 22222. Como la AWSCURRENT etiqueta se elimina de una versión, la AWSPREVIOUS mueve `update-secret-version-stage` automáticamente a esa versión (11111). El efecto es que las AWSPREVIOUS versiones AWSCURRENT y se intercambian.

```
aws secretsmanager update-secret-version-stage \
  --secret-id MyTestSecret \
  --version-stage AWSCURRENT \
  --move-to-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
  --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

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 utilizar

la clave gestionada por el cliente Clave administrada de AWS aws/secretsmanager, o puede utilizar una clave gestionada por el cliente. En la mayoría de los casosaws/secretsmanager, se recomienda su uso, sin coste alguno. Si necesita acceder al secreto desde otra persona Cuenta de AWS o si quiere utilizar su propia clave 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 gestionadaaws/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 las versiones AWSCURRENT, AWSPENDING y AWSPREVIOUS con la nueva clave. Para evitar que descubra el secreto, Secrets Manager mantiene todas las versiones existentes cifradas con la clave anterior. Esto significa que puede descifrar las versiones AWSCURRENT, AWSPENDING y AWSPREVIOUS con la clave anterior o con la nueva clave. Si no tiene permiso kms:Decrypt para usar la clave anterior, al cambiar la clave de cifrado, Secrets Manager no podrá descifrar las versiones secretas para volver a cifrarlas. En este caso, las versiones existentes no se vuelven a cifrar.

Para que solo AWSCURRENT se pueda descifrar con la nueva clave de cifrado, cree una nueva versión del secreto con la nueva clave. Luego, para poder descifrar la versión secreta de AWSCURRENT, 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 “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos 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"
```

Modificar un AWS Secrets Manager secreto

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 campo AWS CLI, llama a describe-secret y, a continuación, revisa 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 CloudTrail registro cuando se modifica un secreto. Para obtener más información, consulte the section called “Inicia 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 “Políticas basadas en recursos”.

- 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 Replicación multirregional.
- 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."
```

AWS SDK

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”.

Encuentra 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 CloudTrail registro al enumerar los secretos. Para obtener más información, consulte the section called “Inicia sesión con AWS CloudTrail ”.

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

Filtros de búsqueda

Si no utiliza ningún filtro, Secrets Manager divide la cadena de búsqueda en palabras y, a continuación, busca coincidencias en todos los atributos. Esta búsqueda no distingue entre mayúsculas y minúsculas. Por ejemplo, la búsqueda de **`My_Secret`** une secretos con las palabras my (mi) o secret (secreto) en el nombre, la descripción o las etiquetas.

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`

Administrado por

Busca secretos gestionados por servicios ajenos a AWS, por ejemplo, CyberArk o HashiCorp.

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`.

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 Encuentra secretos gestionados por otros AWS servicios

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>"
```

AWS SDK

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

Eliminar un AWS Secrets Manager secreto

Debido a la naturaleza crítica de los secretos, AWS Secrets Manager intencionalmente dificulta la eliminación de un secreto. 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, se eliminan todas las etiquetas de ensayo de la versión mediante el AWS SDK AWS CLI o el SDK. Esto marca la versión como obsoleta y permite que Secrets Manager elimine automáticamente la versión en segundo plano.

Si no sabes si una aplicación sigue usando un secreto, puedes crear una CloudWatch alarma de Amazon que te avise de cualquier intento de acceder a un secreto durante el período de recuperación. Para obtener más información, consulte Supervise cuándo se accede a los AWS Secrets Manager secretos cuya eliminación está programada.

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

Secrets Manager genera una entrada de CloudTrail registro cuando eliminas un secreto. Para obtener más información, consulte the section called “Inicia 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 en https://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 en https://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. Puede recuperar el secreto `restore-secret`hasta la fecha y la hora en el campo de DeletionDate respuesta. 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
```

AWS SDK

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”.

Restaura un AWS Secrets Manager secreto

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 CloudTrail registro cuando restauras un secreto. Para obtener más información, consulte the section called “Inicia 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
```

AWS SDK

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

Etiqueta: AWS Secrets Manager secretos

Secrets Manager define una etiqueta como un rótulo que consta de una clave definida por el usuario y un valor opcional. Puedes usar etiquetas para facilitar la administración, la búsqueda y el filtrado de los secretos y otros recursos de tu AWS cuenta. 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 “Controlar el acceso a los secretos mediante etiquetas”.

Puedes encontrar secretos mediante etiquetas en la consola AWS CLI, y SDKs. AWS también proporciona la herramienta Resource Groups para crear una consola personalizada que consolide y organice los recursos en función de sus 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 laAWS Guía de referencia general. Las etiquetas distinguen entre mayúsculas y minúsculas.

Secrets Manager genera una entrada de CloudTrail registro al etiquetar o quitar la etiqueta de un secreto. Para obtener más información, consulte the section called “Inicia 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"]'
```

AWS SDK

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

Replica AWS Secrets Manager secretos en todas las regiones

Puede replicar sus datos secretos en varias Regiones de AWS para admitir aplicaciones distribuidas en esas regiones y cumplir con los requisitos de baja latencia y acceso regional. 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.

Puedes replicar los secretos en todas las AWS regiones habilitadas. Sin embargo, si utilizas Secrets Manager en AWS regiones especiales, como AWS GovCloud (US) las regiones de China, solo podrás configurar los secretos y las réplicas dentro de esas AWS regiones especializadas. No puedes replicar un secreto de AWS las regiones habilitadas en una región especializada ni replicar secretos de una región especializada en 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”. Si desea utilizar la replicación para mejorar la resiliencia de su carga de trabajo, consulte Arquitectura de recuperación ante desastres (DR) en AWS la parte I: Estrategias de recuperación en la nube.

Secrets Manager genera una entrada de CloudTrail registro al replicar un secreto. Para obtener más información, consulte the section called “Inicia 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 está cifrada con la clave AWS gestionadaaws/secretsmanager.

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

Example Crear un secreto y replicarlo

En el siguiente ejemplo, se crea un secreto y se lo replica en eu-west-3. La réplica se cifra con Clave administrada de AWS aws/secretsmanager.

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

Promociona una réplica secreta a una secreta independiente en AWS Secrets Manager

Un secreto de réplica es un secreto que se replica desde un elemento principal en otro. 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 CloudTrail registro cuando promocionas un secreto. Para obtener más información, consulte the section called “Inicia sesión con AWS CloudTrail ”.

Para promover un secreto de réplica (consola)

1. Inicie sesión en 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
```

AWS SDK

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”.

Impedir AWS Secrets Manager la replicación

Como los secretos se pueden replicar utilizando `ReplicateSecretToRegions` o cuando se crean con `CreateSecret`, si quiere impedir que los usuarios repliquen los secretos, le recomendamos que evite las acciones que contengan el parámetro `AddReplicaRegions`. Puede usar una declaración `Condition` en sus políticas de permisos para permitir solo las acciones que no agreguen regiones de réplica. Consulte los siguientes ejemplos de políticas para ver las declaraciones de condiciones que puede utilizar.

Example Impedir el permiso de replicación

El siguiente ejemplo de política muestra cómo permitir todas las acciones que no agreguen regiones de réplica. Esto impide que los usuarios repliquen los secretos mediante `ReplicateSecretToRegions` y `CreateSecret`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:*",
      "Resource": "*",
      "Condition": {
        "Null": {
          "secretsmanager:AddReplicaRegions": "true"
        }
      }
    }
  ]
}
```

Example Habilite el permiso de replicación solo en regiones específicas

En la siguiente política, se muestra cómo permitir todas las operaciones siguientes:

- Crear secretos sin replicación
- Crear secretos replicándolos solo en regiones de Estados Unidos y Canadá
- Replicar secretos solo en regiones de Estados Unidos y Canadá

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret",
        "secretsmanager:ReplicateSecretToRegions"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringLike": {
          "secretsmanager:AddReplicaRegions": [
            "us-*",
            "ca-*"
          ]
        }
      }
    }
  ]
}
```

Solucionar problemas de replicación AWS Secrets Manager

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 AWS regiones. en la Guía de referencia de administración de AWS cuentas.

Obtenga secretos de AWS Secrets Manager

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

Puede recuperar valores secretos mediante:

- Obtener un valor secreto de Secrets Manager con Java
- Obtener un valor secreto de Secrets Manager con Python
- Obtenga un valor secreto de Secrets Manager usando .NET
- Obtener un valor secreto de Secrets Manager con Go
- Obtener un valor secreto de Secrets Manager con Rust
- AWS Secrets Manager Secretos de uso en Amazon Elastic Kubernetes Service
- Usa AWS Secrets Manager secretos en AWS Lambda las funciones
- AWS Secrets Manager Agente
- Obtenga un valor secreto de Secrets Manager con el AWS SDK de C++
- Obtenga un valor secreto de Secrets Manager con el JavaScript AWS SDK
- Obtén un valor secreto de Secrets Manager con el SDK de Kotlin AWS
- Obtenga un valor secreto de Secrets Manager con el AWS SDK de PHP
- Obtenga un valor secreto de Secrets Manager con el AWS SDK de Ruby
- Obtenga un valor secreto mediante el AWS CLI
- Obtenga un valor secreto con la AWS consola
- Usa AWS Secrets Manager secretos en AWS Batch
- Obtén un AWS Secrets Manager secreto en un AWS CloudFormation recurso
- Usa AWS Secrets Manager secretos en los GitHub trabajos
- Usa AWS Secrets Manager secretos en AWS IoT Greengrass
- Usa AWS Secrets Manager secretos en Parameter Store

Obtener un valor secreto de Secrets Manager con Java

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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 conectarse a una base de datos mediante las credenciales de un secreto, puede utilizar los controladores de conexión SQL de Secrets Manager, que incluyen el controlador JDBC básico. Esto también utiliza el almacenamiento en caché del lado del cliente, por lo que puede reducir el costo de llamar a Secrets Manager. APIs

Temas

- Obtener un valor secreto de Secrets Manager mediante Java con almacenamiento en caché del cliente
- Conexión a una base de datos SQL mediante JDBC con credenciales en un secreto de AWS Secrets Manager
- Obtenga un valor secreto de Secrets Manager con el AWS SDK de Java

Obtener un valor secreto de Secrets Manager mediante Java con almacenamiento en caché del cliente

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. Como llamar a Secrets Manager conlleva un coste APIs, el uso de una memoria caché puede reducir los costes. Para conocer todas las formas en las que puede recuperar secretos, consulte Obtener 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.

Para descargar el código fuente, consulte el componente de 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 “Java con almacenamiento en caché del cliente”.

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, 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.

getMaxCacheTamaño

```
public int getMaxCacheSize()
```

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

setMaxCacheTamaño

```
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 Con el cliente

```
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 withMaxCacheTamaño

```
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.

introducción

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

Deriva el objeto a partir del objeto almacenado en caché.

Devuelve el objeto que se devolverá de la caché

Conexión a una base de datos SQL mediante JDBC 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 en 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). O bien, en la llamada. AWS CLIget-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 administrado de 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 configurar 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
```

Obtenga un valor secreto de Secrets Manager con el AWS SDK de Java

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de los AWS SDKs. 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.

- 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` o `BatchGetSecretValue`.

En los siguientes ejemplos de código, se muestra cómo utilizar `GetSecretValue`.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
import software.amazon.awssdk.services.secretsmanager.model.SecretsManagerException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 *
 * We recommend that you cache your secret values by using client-side caching.
 *
 * Caching secrets improves speed and reduces your costs. For more information,
 * see the following documentation topic:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/retrieving-secrets.html
```

```
 */
public class GetSecretValue {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <secretName>\s

                Where:
                    secretName - The name of the secret (for example, tutorials/
MyFirstSecret).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String secretName = args[0];
        Region region = Region.US_EAST_1;
        SecretsManagerClient secretsClient = SecretsManagerClient.builder()
                .region(region)
                .build();

        getValue(secretsClient, secretName);
        secretsClient.close();
    }

    public static void getValue(SecretsManagerClient secretsClient, String secretName)
 {
        try {
            GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                    .secretId(secretName)
                    .build();

            GetSecretValueResponse valueResponse =
 secretsClient.getSecretValue(valueRequest);
            String secret = valueResponse.secretString();
            System.out.println(secret);

        } catch (SecretsManagerException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
```

```
    }
}
```

Obtener un valor secreto de Secrets Manager con Python

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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.

Temas

- Obtener un valor secreto de Secrets Manager mediante Python con almacenamiento en caché del cliente
- Obtenga un valor secreto de Secrets Manager con el AWS SDK de Python
- Obtenga un lote de valores secretos de Secrets Manager con el AWS SDK de Python

Obtener un valor secreto de Secrets Manager mediante Python con almacenamiento en caché del cliente

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. Como llamar a Secrets Manager conlleva un coste APIs, el uso de una memoria caché puede reducir los costes. Para conocer todas las formas en las que puede recuperar secretos, consulte Obtener 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 el componente de 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 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 “Python con almacenamiento en caché del cliente”.

```
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 el 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

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 el 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
- introducción

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

introducción

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')
```

Obtenga un valor secreto de Secrets Manager con el AWS SDK de Python

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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 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`.

En los siguientes ejemplos de código, se muestra cómo utilizar `GetSecretValue`.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
"""
Purpose

Shows how to use the AWS SDK for Python (Boto3) with AWS
Secrets Manager to get a specific of secrets that match a
specified name
"""
import boto3
import logging

from get_secret_value import GetSecretWrapper

# Configure logging
logging.basicConfig(level=logging.INFO)


def run_scenario(secret_name):
    """
    Retrieve a secret from AWS Secrets Manager.

    :param secret_name: Name of the secret to retrieve.
    :type secret_name: str
    """
    try:
        # Validate secret_name
        if not secret_name:
            raise ValueError("Secret name must be provided.")
        # Retrieve the secret by name
        client = boto3.client("secretsmanager")
        wrapper = GetSecretWrapper(client)
        secret = wrapper.get_secret(secret_name)
        # Note: Secrets should not be logged.
        return secret
    except Exception as e:
        logging.error(f"Error retrieving secret: {e}")
        raise

class GetSecretWrapper:
    def __init__(self, secretsmanager_client):
        self.client = secretsmanager_client
```

```
    def get_secret(self, secret_name):
        """
        Retrieve individual secrets from AWS Secrets Manager using the get_secret_value
API.
        This function assumes the stack mentioned in the source code README has been
successfully deployed.
        This stack includes 7 secrets, all of which have names beginning with
"mySecret".

        :param secret_name: The name of the secret fetched.
        :type secret_name: str
        """
        try:
            get_secret_value_response = self.client.get_secret_value(
                SecretId=secret_name
            )
            logging.info("Secret retrieved successfully.")
            return get_secret_value_response["SecretString"]
        except self.client.exceptions.ResourceNotFoundException:
            msg = f"The requested secret {secret_name} was not found."
            logger.info(msg)
            return msg
        except Exception as e:
            logger.error(f"An unknown error occurred: {str(e)}.")
            raise
```

Obtenga un lote de valores secretos de Secrets Manager con el AWS SDK de Python

En el siguiente ejemplo de código se muestra cómo obtener un lote de valores secretos de Secrets Manager.

Permisos necesarios:

- `secretsmanager:BatchGetSecretValue`
- Permiso `secretsmanager:GetSecretValue` para cada uno de los secretos que desea recuperar.
- Si utiliza filtros, también debe tenerlos `secretsmanager:ListSecrets`.

Si desea ver un ejemplo de política de permisos, consulte the section called “Ejemplo: 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.

```
class BatchGetSecretsWrapper:
    def __init__(self, secretsmanager_client):
        self.client = secretsmanager_client


    def batch_get_secrets(self, filter_name):
        """
        Retrieve multiple secrets from AWS Secrets Manager using the
 batch_get_secret_value API.
        This function assumes the stack mentioned in the source code README has been
 successfully deployed.
        This stack includes 7 secrets, all of which have names beginning with
 "mySecret".

        :param filter_name: The full or partial name of secrets to be fetched.
        :type filter_name: str
        """
        try:
            secrets = []
            response = self.client.batch_get_secret_value(
                Filters=[{"Key": "name", "Values": [f"{filter_name}"]}]
            )
            for secret in response["SecretValues"]:
                secrets.append(json.loads(secret["SecretString"]))
            if secrets:
                logger.info("Secrets retrieved successfully.")
            else:
                logger.info("Zero secrets returned without error.")
            return secrets
        except self.client.exceptions.ResourceNotFoundException:
            msg = f"One or more requested secrets were not found with filter:
 {filter_name}"
```

```
            logger.info(msg)
            return msg
        except Exception as e:
            logger.error(f"An unknown error occurred:\n{str(e)}.")
            raise
```

Obtenga un valor secreto de Secrets Manager usando .NET

En las aplicaciones, puedes recuperar tus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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.

Temas

- Obtener un valor secreto de Secrets Manager mediante .NET con almacenamiento en caché del cliente
- Obtenga un valor secreto de Secrets Manager utilizando el SDK para .NET

Obtener un valor secreto de Secrets Manager mediante .NET con almacenamiento en caché del cliente

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. Como llamar a Secrets Manager conlleva un coste APIs, el uso de una memoria caché puede reducir los costes. Para conocer todas las formas en las que puede recuperar secretos, consulte Obtener 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.
- El AWS SDK para .NET. Consulte the section called “AWS SDKs”.

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

- Realice 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 que recupera 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)

El siguiente ejemplo de código muestra un método que recupera un secreto denominado *MySecret* y establece la duración de la actualización de la caché 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 “.NET con almacenamiento en caché del cliente”.

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, como por ejemplo, para utilizar una región o un punto de conexión específicos.

Parámetros

secretsManager

El del que AmazonSecretsManagerClientse van a recuperar 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 caché mediante un cliente Secrets Manager creado con el proporcionado AmazonSecretsManagerClienty unthe section called "SecretCacheConfiguration". 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 AmazonSecretsManagerClientpara recuperar 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; }
```

Los de AmazonSecretsManagerClientlos que recuperar 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

Obtenga un valor secreto de Secrets Manager utilizando el SDK para .NET

En las aplicaciones, puedes recuperar tus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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 .NET, utilice el componente de almacenamiento en caché basado en .NET de Secrets Managero o llame al SDK directamente con `GetSecretValue` o `BatchGetSecretValue`.

Los siguientes ejemplos de código muestran cómo utilizar `GetSecretValue`.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
    using System;
    using System.IO;
    using System.Threading.Tasks;
    using Amazon.SecretsManager;
    using Amazon.SecretsManager.Model;

    /// <summary>
```

```
    /// This example uses the Amazon Web Service Secrets Manager to retrieve
    /// the secret value for the provided secret name.
    /// </summary>
    public class GetSecretValue
    {
        /// <summary>
        /// The main method initializes the necessary values and then calls
        /// the GetSecretAsync and DecodeString methods to get the decoded
        /// secret value for the secret named in secretName.
        /// </summary>
        public static async Task Main()
        {
            string secretName = "<<{{MySecretName}}>>";
            string secret;

            IAmazonSecretsManager client = new AmazonSecretsManagerClient();

            var response = await GetSecretAsync(client, secretName);

            if (response is not null)
            {
                secret = DecodeString(response);

                if (!string.IsNullOrEmpty(secret))
                {
                    Console.WriteLine($"The decoded secret value is: {secret}.");
                }
                else
                {
                    Console.WriteLine("No secret value was returned.");
                }
            }
        }

        /// <summary>
        /// Retrieves the secret value given the name of the secret to
        /// retrieve.
        /// </summary>
        /// <param name="client">The client object used to retrieve the secret
        /// value for the given secret name.</param>
        /// <param name="secretName">The name of the secret value to retrieve.</param>
        /// <returns>The GetSecretValueReponse object returned by
        /// GetSecretValueAsync.</returns>
        public static async Task<GetSecretValueResponse> GetSecretAsync(
```

```
            IAmazonSecretsManager client,
            string secretName)
        {
            GetSecretValueRequest request = new GetSecretValueRequest()
            {
                SecretId = secretName,
                VersionStage = "AWSCURRENT", // VersionStage defaults to AWSCURRENT if
unspecified.
            };

            GetSecretValueResponse response = null;

            // For the sake of simplicity, this example handles only the most
            // general SecretsManager exception.
            try
            {
                response = await client.GetSecretValueAsync(request);
            }
            catch (AmazonSecretsManagerException e)
            {
                Console.WriteLine($"Error: {e.Message}");
            }

            return response;
        }

        /// <summary>
        /// Decodes the secret returned by the call to GetSecretValueAsync and
        /// returns it to the calling program.
        /// </summary>
        /// <param name="response">A GetSecretValueResponse object containing
        /// the requested secret value returned by GetSecretValueAsync.</param>
        /// <returns>A string representing the decoded secret value.</returns>
        public static string DecodeString(GetSecretValueResponse response)
        {
            // Decrypts secret using the associated AWS Key Management Service
            // Customer Master Key (CMK.) Depending on whether the secret is a
            // string or binary value, one of these fields will be populated.
            if (response.SecretString is not null)
            {
                var secret = response.SecretString;
                return secret;
            }
            else if (response.SecretBinary is not null)
```

```
            {
                var memoryStream = response.SecretBinary;
                StreamReader reader = new StreamReader(memoryStream);
                string decodedBinarySecret =
System.Text.Encoding.UTF8.GetString(Convert.FromBase64String(reader.ReadToEnd()));
                return decodedBinarySecret;
            }
            else
            {
                return string.Empty;
            }
        }
    }
```

Obtener un valor secreto de Secrets Manager con Go

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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.

Temas

- Obtener un valor secreto de Secrets Manager mediante Go con almacenamiento en caché del cliente
- Obtén un valor secreto de Secrets Manager con el AWS SDK de Go

Obtener un valor secreto de Secrets Manager mediante Go con almacenamiento en caché del cliente

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. Como llamar a Secrets Manager conlleva un coste APIs, el uso de una memoria caché puede reducir los costes. Para conocer todas las formas en las que puede recuperar secretos, consulte Obtener 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 for Go. Consulte the section called “AWS SDKs”.

Para descargar el código fuente, consulte el cliente de almacenamiento en GitHub caché Secrets Manager Go activado.

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
- escriba CacheConfig
- tipo 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 "Go con almacenamiento en caché del cliente".

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 SecretsManager cliente a partir de una nueva sesión. Se inicializa CacheConfig con 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 secreta de la caché para el identificador secreto dado. 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 secreta de la caché para el ID secreto y la etapa 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 secreto de la caché para el ID secreto dado. 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 secreto de la caché para el ID secreto y la etapa de versión determinados. Devuelve el valor binario del secreto y un error si la operación no pudo llevarse a cabo.

escriba 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
}
```

tipo 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.

Obtén un valor secreto de Secrets Manager con el AWS SDK de Go

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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 Go, utilice el componente de almacenamiento en caché basado en Go de Secrets Manager o llame directamente al SDK con `GetSecretValue` o `BatchGetSecretValue`.

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
 // Use this code snippet in your app.
 // If you need more information about configurations or implementing the sample code,
visit the AWS docs:
 // https://aws.github.io/aws-sdk-go-v2/docs/getting-started/

 import (
   "context"
   "log"

   "github.com/aws/aws-sdk-go-v2/aws"
   "github.com/aws/aws-sdk-go-v2/config"
   "github.com/aws/aws-sdk-go-v2/service/secretsmanager"
 )

 func main() {
   secretName := "<<{{MySecretName}}>>"
   region := "<<{{MyRegionName}}>>"

   config, err := config.LoadDefaultConfig(context.TODO(), config.WithRegion(region))
   if err != nil {
     log.Fatal(err)
   }

   // Create Secrets Manager client
   svc := secretsmanager.NewFromConfig(config)

   input := &secretsmanager.GetSecretValueInput{
     SecretId:     aws.String(secretName),
     VersionStage: aws.String("AWSCURRENT"), // VersionStage defaults to AWSCURRENT if
unspecified
   }

   result, err := svc.GetSecretValue(context.TODO(), input)
```

```
    if err != nil {
      // For a list of exceptions thrown, see
      // https://<<{{DocsDomain}}>>/secretsmanager/latest/apireference/
API_GetSecretValue.html
      log.Fatal(err.Error())
    }

    // Decrypts secret using the associated KMS key.
    var secretString string = *result.SecretString

    // Your code goes here.
  }
```

Obtener un valor secreto de Secrets Manager con Rust

En las aplicaciones, puede recuperar sus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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.

Temas

- Obtener un valor secreto de Secrets Manager mediante Rust con almacenamiento en caché del cliente
- Obtenga un valor secreto de Secrets Manager con el AWS SDK de Rust

Obtener un valor secreto de Secrets Manager mediante Rust con almacenamiento en caché del cliente

Cuando recupera un secreto, puede utilizar el componente de almacenamiento en caché basado en Rust de Secrets Manager para utilizarlo más adelante. Recuperar un secreto almacenado en la memoria caché es más rápido que recuperarlo desde Secrets Manager. Como llamar a Secrets Manager conlleva un coste APIs, el uso de una memoria caché puede reducir los costes. Para conocer todas las formas en las que puede recuperar secretos, consulte Obtener secretos.

La caché sigue la política de primero en entrar, primero en salir (FIFO), por lo que cada vez que la caché tiene que descartar un secreto, descarta el más antiguo. De forma predeterminada, la caché actualiza secretos cada hora. Puede configurar las siguientes opciones:

- `max_size`: el número máximo de secretos en caché que se deben mantener antes de desalojar los secretos a los que no se ha accedido recientemente.
- `ttl`: el tiempo que se considera válido un elemento almacenado en caché antes de requerir una actualización del estado del secreto.

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 los rasgos proporcionados para modificar la caché.

Para utilizar el componente, debe disponer de un entorno de desarrollo Rust 2021 con `tokio`. Para obtener más información, consulte Comenzar en el sitio web del lenguaje de programación Rust.

Para descargar el código fuente, consulte el componente de cliente de almacenamiento en caché basado en Rust de Secrets Manager en. GitHub

Para instalar el componente de almacenamiento en caché, utilice el siguiente comando.

```
cargo add aws_secretsmanager_caching
```

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

Example Recuperación de un secreto

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

```
use aws_secretsmanager_caching::SecretsManagerCachingClient;
use std::num::NonZeroUsize;
use std::time::Duration;

let client = match SecretsManagerCachingClient::default(
    NonZeroUsize::new(10).unwrap(),
    Duration::from_secs(60),
```

```
)
.await
{
    Ok(c) => c,
    Err(_) => panic!("Handle this error"),
};

let secret_string = match client.get_secret_value("MyTest", None, None).await {
    Ok(s) => s.secret_string.unwrap(),
    Err(_) => panic!("Handle this error"),
};

// Your code here
```

Example Creación de instancias de caché con una configuración y un cliente personalizados

En el siguiente ejemplo se muestra cómo configurar la caché y, a continuación, obtener el valor secreto de un secreto denominado*MyTest*.

```
let config = aws_config::load_defaults(BehaviorVersion::latest())
    .await
    .into_builder()
    .region(Region::from_static("us-west-2"))
    .build();

let asm_builder = aws_sdk_secretsmanager::config::Builder::from(&config);

let client = match SecretsManagerCachingClient::from_builder(
    asm_builder,
    NonZeroUsize::new(10).unwrap(),
    Duration::from_secs(60),
)
.await
{
    Ok(c) => c,
    Err(_) => panic!("Handle this error"),
};

let secret_string = client
    .get_secret_value("MyTest", None, None)
    .await
    {
        Ok(c) => c.secret_string.unwrap(),
```

```
        Err(_) => panic!("Handle this error"),
    };

// Your code here
```

Obtenga un valor secreto de Secrets Manager con el AWS SDK de Rust

En las aplicaciones, puedes recuperar tus secretos llamando `GetSecretValue` o `BatchGetSecretValue` en cualquiera de las AWS SDKs. 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 las aplicaciones de Rust, utilice el componente de almacenamiento en caché basado en Rust de Secrets Manager o llame al SDK directamente con GetSecretValue o. BatchGetSecretValue

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
async fn show_secret(client: &Client, name: &str) -> Result<(), Error> {
    let resp = client.get_secret_value().secret_id(name).send().await?;

    println!("Value: {}", resp.secret_string().unwrap_or("No value!"));

    Ok(())
}
```

AWS Secrets Manager Secretos de uso en Amazon Elastic Kubernetes Service

Para mostrar los secretos de AWS Secrets Manager (ASCP) como archivos montados en los pods de Amazon EKS, puede utilizar el proveedor de AWS secretos y configuración del controlador CSI de Kubernetes Secrets Store. El ASCP funciona con Amazon Elastic Kubernetes Service 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 desea montar en Amazon EKS. El ASCP usa JMESPath sintaxis para consultar los pares clave-valor de tu secreto. El ASCP también funciona con parámetros

del almacén de parámetros. El ASCP ofrece dos métodos de autenticación con Amazon EKS. El primer enfoque utiliza las funciones de IAM para cuentas de servicio (IRSA). El segundo enfoque utiliza identidades de pod. Cada enfoque tiene sus ventajas y casos de uso.

ASCP con funciones de IAM para cuentas de servicio (IRSA)

El ASCP con funciones de IAM para cuentas de servicio (IRSA) le permite montar datos secretos a AWS Secrets Manager partir de archivos en sus Amazon EKS Pods. Este enfoque es adecuado cuando:

- Tienes que guardar los secretos como archivos en tus Pods.
- Está utilizando Amazon EKS versión 1.17 o posterior con grupos de EC2 nodos de Amazon.
- Desea recuperar pares clave-valor específicos de secretos con formato JSON.

Para obtener más información, consulte the section called “Integre ASCP con IRSA para Amazon EKS”.

ASCP con Pod Identity

ASCP con EKS Pod Identity

El método ASCP with Pod Identity mejora la seguridad y simplifica la configuración para acceder a los secretos en Amazon EKS. Este enfoque resulta beneficioso cuando:

- Necesita una gestión de permisos más detallada a nivel de pod.
- Está utilizando Amazon EKS versión 1.24 o posterior.
- Desea mejorar el rendimiento y la escalabilidad.

Para obtener más información, consulte the section called “Integre ASCP con Pod Identity para Amazon EKS”.

Elección del enfoque correcto

Tenga en cuenta los siguientes factores al decidir entre el ASCP con IRSA y el ASCP con Pod Identity:

- Amazon EKSversion: Pod Identity requiere Amazon EKS 1.24+, mientras que el controlador CSI funciona con Amazon EKS 1.17+.

- Requisitos de seguridad: Pod Identity ofrece un control más detallado a nivel de pod.
- Rendimiento: Por lo general, Pod Identity funciona mejor en entornos de gran escala.
- Complejidad: Pod Identity simplifica la configuración al eliminar la necesidad de cuentas de servicio independientes.

Elija el método que mejor se adapte a sus requisitos específicos y al entorno de Amazon EKS.

Instalación de ASCP para Amazon EKS

En esta sección se explica cómo instalar el proveedor de AWS secretos y configuración para Amazon EKS. Con ASCP, puede montar los secretos de Secrets Manager y los parámetros desde AWS Systems Manager archivos en Amazon EKS Pods.

Requisitos previos

- Un clúster de Amazon EKS
  - Versión 1.24 o posterior para Pod Identity
  - Versión 1.17 o posterior para IRSA
- El AWS CLI instalado y configurado
- kubectl instalado y configurado para su clúster de Amazon EKS
- Helm (versión 3.0 o posterior)

Instale y configure 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.

Durante la instalación, puede configurar el ASCP para que utilice un punto de conexión FIPS. Para obtener una lista de puntos de enlace , consulte the section called “Puntos de conexión de Secrets Manager”.

Instalar el ASCP mediante Helm

1. Para asegurarse de que el repositorio apunta a los gráficos más recientes, utilice `helm repo update.`
2. Agregue el gráfico de controladores CSI de Secrets Store.

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

3. Instale el gráfico. Para configurar la limitación, agregue el siguiente indicador: `--set-json 'k8sThrottlingParams={"qps": "number of queries per second", "burst": "number of queries per second"}'`.

```
helm install -n kube-system csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver
```

4. Agregue el gráfico del ASCP.

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

5. Instale el gráfico. Para utilizar un punto de conexión FIPS, agregue el siguiente indicador: `--set useFipsEndpoint=true`.

```
helm install -n kube-system secrets-provider-aws aws-secrets-manager/secrets-store-csi-driver-provider-aws
```

Instalarlo mediante el YAML del repositorio

- Use 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
```

Verifica las instalaciones

Para verificar las instalaciones del clúster EKS, el controlador CSI de Secrets Store y el complemento ASCP, siga estos pasos:

1. Compruebe el clúster de EKS:

```
eksctl get cluster --name clusterName
```

Este comando debería devolver información sobre el clúster.

2. Compruebe la instalación del controlador CSI de Secrets Store:

```
kubectl get pods -n kube-system -l app=secrets-store-csi-driver
```

Deberías ver Pods ejecutándose con nombres como`csi-secrets-store-secrets-store-csi-driver-xxx`.

3. Verifica la instalación del complemento ASCP:

YAML installation

```
$ kubectl get pods -n kube-system -l app=csi-secrets-store-provider-aws
```

Ejemplo de salida:

```
NAME                                     READY   STATUS    RESTARTS   AGE
csi-secrets-store-provider-aws-12345      1/1     Running   0          2m
```

Helm installation

```
$  kubectl get pods -n kube-system -l app=secrets-store-csi-driver-provider-aws
```

Ejemplo de salida:

```
NAME                                                READY   STATUS    RESTARTS
 AGE
secrets-provider-aws-secrets-store-csi-driver-provider-67890       1/1
 Running   0          2m
```

Deberías ver Pods en el `Running` estado.

Después de ejecutar estos comandos, si todo está configurado correctamente, debería ver que todos los componentes se ejecutan sin errores. Si tienes algún problema, es posible que tengas que comprobarlo consultando los registros de los pods específicos que están teniendo problemas.

Solución de problemas

1. Para comprobar los registros del proveedor de ASCP, ejecuta:

   ```
   kubectl logs -n kube-system -l app=csi-secrets-store-provider-aws
   ```

2. Compruebe el estado de todos los pods del `kube-system` espacio de nombres:

   ```
   kubectl -n kube-system get pods
   ```

   ```
   kubectl -n kube-system logs pod/PODID
   ```

   Todos los pods relacionados con el controlador CSI y el ASCP deben estar en estado «En ejecución».

3. Compruebe la versión del controlador CSI:

   ```
   kubectl get csidriver secrets-store.csi.k8s.io -o yaml
   ```

   Este comando debería devolver información sobre el controlador CSI instalado.

Recursos adicionales

Para obtener más información sobre el uso de ASCP con Amazon EKS, consulte los siguientes recursos:

- Uso de Pod Identity con Amazon EKS
- Uso del proveedor de AWS secretos y configuración
- AWS Secrets Store CSI está activado GitHub

Utilice el CSI del proveedor de AWS secretos y configuración con Pod Identity para Amazon EKS

La integración del proveedor de AWS secretos y configuración con el agente de identidad del pod para Amazon Elastic Kubernetes Service proporciona una seguridad mejorada, una configuración simplificada y un rendimiento mejorado para las aplicaciones que se ejecutan en Amazon EKS. Pod Identity simplifica la autenticación de IAM para Amazon EKS al recuperar secretos de Secrets Manager o parámetros de Parameter Store AWS Systems Manager .

Amazon EKS Pod Identity agiliza el proceso de configuración de los permisos de IAM para las aplicaciones de Kubernetes al permitir que los permisos se configuren directamente a través de las interfaces de Amazon EKS, lo que reduce el número de pasos y elimina la necesidad de cambiar entre los servicios de Amazon EKS e IAM. Pod Identity permite el uso de una sola función de IAM en varios clústeres sin actualizar las políticas de confianza y admite etiquetas de sesión de funciones para un control de acceso más detallado. Este enfoque no solo simplifica la administración de políticas al permitir la reutilización de las políticas de permisos en todos los roles, sino que también mejora la seguridad al permitir el acceso a AWS los recursos en función de las etiquetas coincidentes.

Funcionamiento

1. Pod Identity asigna una función de IAM al pod.
2. ASCP usa este rol para autenticarse con. Servicios de AWS
3. Si está autorizado, el ASCP recupera los secretos solicitados y los pone a disposición del pod.

Para obtener más información, consulte Cómo funciona Amazon EKS Pod Identity en la Guía del usuario de Amazon EKS.

Requisitos previos

> **Important**
>
> Pod Identity solo es compatible con Amazon EKS en la nube. No es compatible con Amazon EKS Anywhere ni con los clústeres de Kubernetes autogestionados en las instancias de Amazon. Red Hat OpenShift Service en AWS EC2

- Clúster Amazon EKS (versión 1.24 o posterior)

- Acceso a un clúster AWS CLI de Amazon EKS a través de `kubectl`
- Acceso a dos Cuentas de AWS (para acceso entre cuentas)

Instalación del Amazon EKS Pod Identity Agent

Para usar Pod Identity con su clúster, debe instalar el complemento Amazon EKS Pod Identity Agent.

Para instalar el Pod Identity Agent

- Instala el complemento Pod Identity Agent en tu clúster:

```
eksctl create addon \
  --name eks-pod-identity-agent \
  --cluster clusterName \
  --region region
```

Configura un ASCP con Pod Identity

1. Crea una política de permisos que conceda `secretsmanager:GetSecretValue` `secretsmanager:DescribeSecret` permisos a los secretos a los que el Pod necesita acceder. Para ver una política de ejemplo, consulte the section called “Ejemplo: Permiso para leer y describir secretos individuales”.
2. Cree una función de IAM que pueda asumir el director de servicio de Amazon EKS para Pod Identity:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "pods.eks.amazonaws.com"
        },
        "Action": [
          "sts:AssumeRole",
          "sts:TagSession"
        ]
      }
    ]
```

```
  }
```

Adjunte la política de IAM al rol:

```
aws iam attach-role-policy \
  --role-name MY_ROLE \
  --policy-arn POLICY_ARN
```

3. Crea una asociación de identidad de Pod. Para ver un ejemplo, consulte Crear una asociación de identidad de pod y Crear una asociación de identidad de pod en la Guía del usuario de Amazon EKS.
4. Crea una `SecretProviderClass` que especifique qué secretos montar en el pod:

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

La diferencia clave SecretProviderClass entre IRSA y Pod Identity es el parámetro `usePodIdentity` opcional. Es un campo opcional que determina el enfoque de autenticación. Si no se especifica, se utilizan de forma predeterminada las funciones de IAM para las cuentas de servicio (IRSA).

- Para usar EKS Pod Identity, utilice cualquiera de estos valores:. `"true", "True", "TRUE", "t", "T"`
- Para usar IRSA de forma explícita, establézcalo en cualquiera de estos valores:`"false", "False", "FALSE", "f", or "F"`.

5. Despliega el módulo que contiene los secretos que se encuentran debajo: `/mnt/secrets-store`

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

6. 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 más información sobre la creación de un punto de conexión, consulte Puntos de conexión de VPC de tipo interfaz en la Guía del usuario de AWS Identity and Access Management .

Compruebe el montaje secreto

Para comprobar que el secreto está montado correctamente, ejecute el siguiente comando:

```
kubectl exec -it $(kubectl get pods | awk '/pod-identity-deployment/{print $1}' | head
 -1) -- cat /mnt/secrets-store/MySecret
```

Para configurar Amazon EKS Pod Identity para acceder a los secretos de Secrets Manager

1. Crea una política de permisos que conceda `secretsmanager:GetSecretValue` `secretsmanager:DescribeSecret` permisos a los secretos a los que el Pod necesita acceder. Para ver una política de ejemplo, consulte the section called “Ejemplo: Permiso para leer y describir secretos individuales”.
2. Crea un secreto en Secrets Manager, si aún no lo tienes.

Solución de problemas

Para ver la mayoría de los errores, describe la implementación del Pod.

Ver los mensajes de error del contenedor

1. Obtén una lista de los nombres de los 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, *PODID* usa el ID de pod de los pods que encontraste 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, usa el siguiente comando para *PODID* usar el ID del pod csi-secrets-store-provider-aws.

  ```
  kubectl -n kube-system get pods
  ```

```
kubectl -n kube-system logs pod/PODID
```

Utilice el CSI del proveedor de AWS secretos y configuración con funciones de IAM para cuentas de servicio (IRSA)

Temas

Requisitos previos

- Clúster Amazon EKS (versión 1.17 o posterior)
- Acceso a un clúster AWS CLI de Amazon EKS a través de `kubectl`

Configurar el control de acceso

La ASCP recupera la identidad del pod de Amazon EKS y la cambia por una función de IAM. Los permisos se establecen en una política de IAM para ese rol de IAM. Cuando el ASCP asume el rol de IAM, 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.

Para conceder a tu Amazon EKS Pod acceso a los secretos de Secrets Manager

1. Crea una política de permisos que conceda `secretsmanager:GetSecretValue` `secretsmanager:DescribeSecret` permisos a los secretos a los que el Pod necesita acceder. Para ver una política de ejemplo, consulte the section called “Ejemplo: Permiso para leer y describir secretos individuales”.
2. Cree un proveedor OpenID Connect (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 en la Guía del usuario de Amazon EKS.
3. Cree un rol de IAM para la cuenta de servicio y adjunte la política. Para obtener más información, consulte Crear un rol de IAM para su cuenta de servicio en la Guía del usuario de Amazon EKS.

4. 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 más información sobre la creación de un punto de conexión, consulte Puntos de conexión de VPC de tipo interfaz en la Guía del usuario de AWS Identity and Access Management .

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 the section called “SecretProviderClass”. El `SecretProviderClass` contiene una lista de los secretos que hay que montar y el nombre de archivo con el que montarlos. `SecretProviderClass`Debe estar en el mismo espacio de nombres que el Amazon EKS Pod al que hace referencia.

Monta los secretos como archivos

Las siguientes instrucciones muestran cómo montar los secretos como archivos utilizando los archivos YAML de ejemplo .yaml y ExampleSecretProviderClass.yaml. ExampleDeployment

Montar secretos en Amazon EKS

1. Aplícalos al pod: `SecretProviderClass`

```
kubectl apply -f ExampleSecretProviderClass.yaml
```

2. Despliega tu pod:

```
kubectl apply -f ExampleDeployment.yaml
```

3. El ASCP monta los archivos.

Solución de problemas

Para ver la mayoría de los errores, describe la implementación del Pod.

Ver los mensajes de error del contenedor

1. Obtén una lista de los nombres de los 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, *podId* usa el ID de pod de los pods que encontraste 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, usa el siguiente comando para *podId* usar el ID del pod csi-secrets-store-provider-aws.

```
kubectl -n kube-system get pods
kubectl -n kube-system logs Pod/podId
```

- Compruebe que el **`SecretProviderClass`** CRD esté instalado:

```
kubectl get crd secretproviderclasses.secrets-store.csi.x-k8s.io
```

Este comando debe devolver información sobre la definición de recurso `SecretProviderClass` personalizada.

- Compruebe que se ha creado el SecretProviderClass objeto.

```
kubectl get secretproviderclass SecretProviderClassName -o yaml
```

AWS Ejemplos de código de proveedores de secretos y configuraciones

Ejemplos de autenticación y control de acceso ASCP

Ejemplo: política de IAM que permite al servicio Amazon EKS Pod Identity (pods.eks.amazonaws.com) asumir el rol y etiquetar la sesión:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

SecretProviderClass

Se debe utilizar YAML para describir qué secretos hay que montar en Amazon EKS mediante el ASCP. Para ver ejemplos, consulta the section called “SecretProviderClass uso”.

SecretProviderClass Estructura de YAML

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: name
spec:
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    usePodIdentity:
    preferredAddressType:
    objects:
```

El campo de parámetros contiene los detalles de la solicitud de montaje:

region

(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 añade una sobrecarga a la hora de montar solicitudes, por

lo que te recomendamos que indiques la región para los clústeres que utilizan un gran número 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 Conmutación por error secreta multirregional.

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`.

usePodIdentity

(Opcional) Determina el enfoque de autenticación. Si no se especifica, el valor predeterminado es Roles de IAM para cuentas de servicio (IRSA) (IRSA).

- Para usar EKS Pod Identity, utilice cualquiera de estos valores: `"true"` «,,`"True"`, `"TRUE"` o. `"t"` `"T"`
- Para usar IRSA de forma explícita, establézcalo en cualquiera de estos valores: `"false""False","FALSE","f"`, o `"F"` «=.

preferredAddressType

(Opcional) Especifica el tipo de dirección IP preferido para la comunicación con los terminales de Pod Identity Agent. El campo solo se aplica cuando se utiliza la función EKS Pod Identity y se ignorará cuando se utilicen las funciones de IAM para las cuentas de servicio. Los valores no distinguen entre mayúsculas y minúsculas. Los valores válidos son:

- `"ipv4"`, `"IPv4"` «, o `"IPV4"` — Forzar el uso del terminal Pod Identity Agent IPv4
- `"ipv6"`,`"IPv6"`, o `"IPV6"` — Forzar el uso del IPv6 punto final del Pod Identity Agent
- no especificado: utilice la selección automática del punto final, pruebe primero el IPv4 punto final y vuelva al IPv6 punto final si IPv4 falla

objetos

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

Obligatorio. Especifica el nombre del secreto o parámetro que se va a obtener. Para Secrets Manager, este es el `SecretId`parámetro y puede ser el nombre descriptivo o el ARN completo del secreto. Para SSM Parameter Store, es el `Name`del parámetro y puede ser el nombre o el ARN completo del parámetro.

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 del Amazon EKS Pod. 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 “Versiones

de un secreto". 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`.

path

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

El nombre del archivo que se va a montar en el Amazon EKS Pod. 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 Conmutación por error a un secreto diferente.

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 Conmutación por error secreta multirregional.

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 “Versiones de un secreto”.

Crea una SecretProviderClass configuración básica para montar secretos en tus Amazon EKS Pods.

Pod Identity

SecretProviderClass para usar un secreto en el mismo clúster de Amazon EKS:

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets-manager
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "mySecret"
        objectType: "secretsmanager"
    usePodIdentity: "true"
```

IRSA

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: deployment-aws-secrets
spec:
  provider: aws
  parameters:
    objects: |
        - objectName: "MySecret"
          objectType: "secretsmanager"
```

SecretProviderClass uso

Utilice estos ejemplos para crear SecretProviderClass configuraciones para diferentes escenarios.

Ejemplo: Montar secretos por nombre o ARN

En este ejemplo se muestra cómo montar tres tipos diferentes de secretos:

- Un secreto especificado por el ARN completo
- Un secreto especificado por su nombre
- 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-2:777788889999:secret:MySecret2-
d4e5f6"
      - objectName: "MySecret3"
        objectType: "secretsmanager"
      - objectName: "MySecret4"
        objectType: "secretsmanager"
        objectVersionLabel: "AWSCURRENT"
```

Ejemplo: monta pares clave-valor a partir de un secreto

En este ejemplo se muestra cómo montar pares clave-valor específicos a partir de un secreto con formato JSON:

```
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-2:777788889999:secret:MySecret-
a1b2c3"
        jmesPath:
            - path: username
              objectAlias: dbusername
            - path: password
              objectAlias: dbpassword
```

Ejemplo: ejemplos de configuración de conmutación por error

Estos ejemplos muestran cómo configurar la conmutación por error para los secretos.

Conmutación por error secreta multirregional

En este ejemplo se muestra cómo configurar la conmutación por error automática para un secreto replicado en varias regiones:

```
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"
```

Conmutación por error a un secreto diferente

En este ejemplo se muestra cómo configurar la conmutación por error a un secreto diferente (no a una réplica):

```
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-1:777788889999:secret:MySecret-
a1b2c3"
        objectAlias: "MyMountedSecret"
        failoverObject:
          - objectName: "arn:aws:secretsmanager:us-
east-2:777788889999:secret:MyFailoverSecret-d4e5f6"
```

Recursos adicionales

Para obtener más información sobre el uso de ASCP con Amazon EKS, consulte los siguientes recursos:

- Uso de Pod Identity con Amazon EKS
- Uso del proveedor de AWS secretos y configuración
- AWS Secrets Store CSI está activado GitHub

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. Como llamar a Secrets Manager conlleva un coste APIs, el uso de una memoria caché puede reducir los costes. 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 de Lambda se ejecuta en una VPC, debe crear un punto de conexión de VPC para que la extensión pueda realizar llamadas a Secrets Manager. Para obtener más información, consulte the section called “Puntos de conexión de VPC (AWS PrivateLink)”.

Permisos necesarios:

- El rol de ejecución de Lambda debe tener el permiso `secretsmanager:GetSecretValue` 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`, el rol de ejecución también necesitará el `kms:Decrypt` permiso para la clave de KMS.

Para usar la extensión AWS Lambda Parameters and Secrets

1. Añada la capa de AWS denominada Extensión de Lambda para parámetros y secretos de AWS a su función. Para obtener instrucciones, consulte Adición de capas a las funciones en la Guía para desarrolladores de Lambda. Si utiliza AWS CLI para añadir la capa, necesitará el ARN de

la extensión. Para obtener una lista ARNs, consulte AWS Parámetros y secretos de la extensión Lambda ARNs en la Guía del AWS Systems Manager usuario.

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 se cifra con una clave gestionada por el cliente en lugar de con Clave administrada de AWS `aws/secretsmanager`, el rol de ejecución también necesitará el `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 obtener más información, consulte the section called "Controlar el acceso a los secretos mediante etiquetas".
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_EXTENSION_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_EXTENSION_CACHE_SIZE` es 0. El valor predeterminado es de 300 segundos.

AWS Secrets Manager Agente

El AWS Secrets Manager agente es un servicio HTTP del lado del cliente que puede utilizar para estandarizar el consumo de información confidencial de Secrets Manager en entornos como Amazon Elastic Container Service, AWS Lambda Amazon Elastic Kubernetes Service y Amazon Elastic

Compute Cloud. El Agente de Secrets Manager puede recuperar y almacenar en caché los secretos de la memoria para que sus aplicaciones puedan consumirlos directamente desde el caché. Esto significa que puede obtener los secretos que su aplicación necesita del servidor local en lugar de tener que realizar llamadas a Secrets Manager. El agente de Secrets Manager solo puede realizar solicitudes de lectura a Secrets Manager; no puede modificar los secretos.

El agente de Secrets Manager utiliza las AWS credenciales que usted proporciona en su entorno para realizar llamadas a Secrets Manager. El Agente de Secrets Manager ofrece protección contra la falsificación de solicitudes del lado del servidor (SSRF) para ayudar a mejorar la seguridad del secreto. Puede configurar el Agente de Secrets Manager estableciendo el número máximo de conexiones, el tiempo de vida (TTL), el puerto HTTP del servidor local y el tamaño de la caché.

Como el Agente de Secrets Manager utiliza una caché en memoria, se restablece cuando se reinicia el Agente de Secrets Manager. El Agente de Secrets Manager actualiza periódicamente el valor secreto almacenado en caché. La actualización se produce cuando se intenta leer un secreto del Agente de Secrets Manager después de que el TTL haya caducado. La frecuencia de actualización predeterminada (TTL) es de 300 segundos y puede cambiarla mediante un Archivo de configuración que se pasa al Agente de Secrets Manager mediante el argumento de la línea de comandos `--config`. El Agente de Secrets Manager no incluye la invalidación del caché. Por ejemplo, si un secreto rota antes de que caduque la entrada del caché, el Agente de Secrets Manager podría devolver un valor secreto obsoleto.

El Agente de Secrets Manager devuelve los valores secretos en el mismo formato que la respuesta de `GetSecretValue`. Los valores secretos no se cifran en caché.

Para descargar el código fuente, consulte https://github.com/aws/aws-secretsmanager-agenten GitHub.

Temas

- Paso 1: crear el binario del Agente de Secrets Manager
- Paso 2: instalar el Agente de Secrets Manager
- Paso 3: recuperar secretos con el Agente de Secrets Manager
- Actualice los secretos a la fuerza con RefreshNow
- Configuración del Agente de Secrets Manager
- Registro
- Consideraciones de seguridad

Paso 1: crear el binario del Agente de Secrets Manager

Para crear el binario del Agente de Secrets Manager de forma nativa, necesita las herramientas de desarrollo estándar y las herramientas de Rust. Como alternativa, puede realizar una compilación cruzada para los sistemas que lo admitan, o puede usar Rust de forma cruzada para realizar una compilación cruzada.

RPM-based systems

1. En los sistemas basados en RPM, como el AL2 023, puede instalar las herramientas de desarrollo mediante el grupo de herramientas de desarrollo.

   ```
   sudo yum -y groupinstall "Development Tools"
   ```

2. Siga las instrucciones de Instalar Rust en la documentación de Rust.

   ```
   curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
   . "$HOME/.cargo/env"
   ```

3. Cree el agente mediante el comando cargo build:

   ```
   cargo build --release
   ```

   Encontrará el ejecutable en `target/release/aws-secrets-manager-agent`.

Debian-based systems

1. En sistemas basados en Debian, como Ubuntu, puede instalar las herramientas de desarrollador mediante el paquete build-essential.

   ```
   sudo apt install build-essential
   ```

2. Siga las instrucciones de Instalar Rust en la documentación de Rust.

   ```
   curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
   . "$HOME/.cargo/env"
   ```

3. Cree el agente mediante el comando cargo build:

```
cargo build --release
```

Encontrará el ejecutable en `target/release/aws-secrets-manager-agent`.

Windows

Para compilar en Windows, siga las instrucciones de Configurar el entorno de desarrollo en Windows para Rust en la documentación de Microsoft Windows.

Cross-compile natively

En las distribuciones en las que está disponible el paquete mingw-w64, como Ubuntu, puede realizar compilaciones cruzadas de forma nativa.

```
# Install the cross compile tool chain
sudo add-apt-repository universe
sudo apt install -y mingw-w64

# Install the rust build targets
rustup target add x86_64-pc-windows-gnu

# Cross compile the agent for Windows
cargo build --release --target x86_64-pc-windows-gnu
```

Encontrará el ejecutable en `target/x86_64-pc-windows-gnu/release/aws-secrets-manager-agent.exe`.

Cross compile with Rust cross

Si las herramientas de compilación cruzada no están disponibles de forma nativa en el sistema, puede utilizar el proyecto cruzado de Rust. Para obtener más información, consulte cross. https://github.com/cross-rs/

> **Important**
>
> Recomendamos 32 GB de espacio en disco para el entorno de compilación.

```
# Install and start docker
```

```
sudo yum -y install docker
sudo systemctl start docker
sudo systemctl enable docker # Make docker start after reboot

# Give ourselves permission to run the docker images without sudo
sudo usermod -aG docker $USER
newgrp docker

# Install cross and cross compile the executable
cargo install cross
cross build --release --target x86_64-pc-windows-gnu
```

Paso 2: instalar el Agente de Secrets Manager

Según el tipo de procesamiento, tiene varias opciones para instalar el Agente de Secrets Manager.

Amazon EKS, Amazon EC2, and Amazon ECS

Instalar el Agente de Secrets Manager

1. Use el script de `install` que se proporciona en el repositorio.

   El script genera un token SSRF aleatorio al inicio y lo almacena en el archivo`/var/run/awssmatoken`. El grupo `awssmatokenreader` que crea el script de instalación puede leer el token.

2. Para permitir que la aplicación lea el archivo de token, debe añadir, al grupo `awssmatokenreader`, la cuenta de usuario con la que se ejecuta la aplicación. Por ejemplo, puedes conceder permisos para que tu aplicación lea el archivo token con el siguiente comando usermod, que *<APP_USER>* es el ID de usuario con el que se ejecuta la aplicación.

   ```
   sudo usermod -aG awssmatokenreader <APP_USER>
   ```

Docker

Puede ejecutar el Agente de Secrets Manager como un contenedor lateral junto con la aplicación mediante Docker. Luego, su aplicación puede recuperar los secretos del servidor HTTP local que proporciona el Agente de Secrets Manager. Para obtener más información sobre Docker, consulte la documentación de Docker.

Crear un contenedor lateral para el Agente de Secrets Manager con Docker

1. Cree un Dockerfile para el contenedor lateral del Agente de Secrets Manager. En el siguiente ejemplo, se crea un contenedor de Docker con el binario del Agente de Secrets Manager.

```
# Use the latest Debian image as the base
FROM debian:latest

# Set the working directory inside the container
WORKDIR /app

# Copy the Secrets Manager Agent binary to the container
COPY secrets-manager-agent .

# Install any necessary dependencies
RUN apt-get update && apt-get install -y ca-certificates

# Set the entry point to run the Secrets Manager Agent binary
ENTRYPOINT ["./secrets-manager-agent"]
```

2. Cree un Dockerfile para su aplicación cliente.
3. Cree un archivo de Docker Compose para ejecutar ambos contenedores, asegurándose de que utilizan la misma interfaz de red. Esto es necesario porque el Agente de Secrets Manager no acepta solicitudes desde fuera de la interfaz localhost. El siguiente ejemplo muestra un archivo de Docker Compose en el que la clave `network_mode` adjunta el contenedor `secrets-manager-agent` al espacio de nombres de red del contenedor `client-application`, lo que les permite compartir la misma interfaz de red.

> **Important**
>
> Debe cargar AWS las credenciales y el token de la SSRF para que la aplicación pueda utilizar el agente Secrets Manager. Consulte lo siguiente:
>
> - Gestionar el acceso en la Guía del usuario de Amazon Elastic Kubernetes Service
> - Rol de IAM en las tareas de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service

```
version: '3'
services:
```

```
    client-application:
    container_name: client-application
    build:
        context: .
        dockerfile: Dockerfile.client
    command: tail -f /dev/null  # Keep the container running


    secrets-manager-agent:
    container_name: secrets-manager-agent
    build:
        context: .
        dockerfile: Dockerfile.agent
    network_mode: "container:client-application"  # Attach to the client-
application container's network
    depends_on:
        - client-application
```

4. Copie el binario `secrets-manager-agent` en el mismo directorio que contiene sus archivos Dockerfiles y Docker Compose.
5. Cree y ejecute los contenedores en función de los Dockerfiles proporcionados mediante el siguiente comando `docker-compose`.

```
docker-compose up --build
```

6. En su contenedor de clientes, ahora puede usar el Agente de Secrets Manager para recuperar secretos. Para obtener más información, consulte the section called “Paso 3: recuperar secretos con el Agente de Secrets Manager”.

AWS Lambda

Puedes empaquetar el Secrets Manager Agent como una AWS Lambda extensión. A continuación, puede añadirla a la función de Lambda como una capa y llamar al Agente de Secrets Manager desde la función de Lambda para obtener los secretos.

Las siguientes instrucciones muestran cómo obtener un nombre secreto *MyTest* mediante el script `secrets-manager-agent-extension.sh` de ejemplo https://github.com/aws/aws-secretsmanager-agentpara instalar el agente Secrets Manager como una extensión de Lambda.

> **Note**
>
> El script de ejemplo usa el comando `curl`, que se incluye en los tiempos de ejecución basados en Amazon Linux 2023, como Python 3.12 y Node.js 20. Si utiliza un entorno de tiempo de ejecución basado en Amazon Linux 2, como Python 3.11 o Node.js 18, primero debe instalar `curl` en la imagen de contenedor de Lambda. Para obtener instrucciones, consulte Cómo puedo utilizar los paquetes binarios nativos de AMI de Amazon Linux 2 con Lambda en AWS Re:post.

Crear una extensión de Lambda que empaquete el Agente de Secrets Manager

1. Cree una función de Lambda de Python que consulte `http://localhost:2773/secretsmanager/get?secretId=MyTest` para obtener el secreto. Asegúrese de implementar la lógica de reintento en el código de la aplicación para adaptarse a los retrasos en la inicialización y el registro de la extensión de Lambda.
2. Desde la raíz del paquete de códigos del Agente de Secrets Manager, ejecute los siguientes comandos para probar la extensión de Lambda.

```
AWS_ACCOUNT_ID=<AWS_ACCOUNT_ID>
LAMBDA_ARN=<LAMBDA_ARN>

# Build the release binary
cargo build --release --target=x86_64-unknown-linux-gnu

# Copy the release binary into the `bin` folder
mkdir -p ./bin
cp ./target/x86_64-unknown-linux-gnu/release/aws_secretsmanager_agent ./bin/
secrets-manager-agent

# Copy the `secrets-manager-agent-extension.sh` script into the `extensions`
 folder.
mkdir -p ./extensions
cp aws_secretsmanager_agent/examples/example-lambda-extension/secrets-manager-
agent-extension.sh ./extensions

# Zip the extension shell script and the binary
zip secrets-manager-agent-extension.zip bin/* extensions/*

# Publish the layer version
```

```
LAYER_VERSION_ARN=$(aws lambda publish-layer-version \
    --layer-name secrets-manager-agent-extension \
    --zip-file "fileb://secrets-manager-agent-extension.zip" | jq -r
 '.LayerVersionArn')

# Attach the layer version to the Lambda function
aws lambda update-function-configuration \
    --function-name $LAMBDA_ARN \
    --layers "$LAYER_VERSION_ARN"
```

3. Invoque la función de Lambda para comprobar que el secreto se está recuperando correctamente.

Paso 3: recuperar secretos con el Agente de Secrets Manager

Para usar el agente, debe llamar al punto de conexión local del Agente de Secrets Manager e incluir el nombre o el ARN del secreto como parámetro de consulta. De forma predeterminada, el Agente de Secrets Manager recupera la versión `AWSCURRENT` del secreto. Para recuperar una versión diferente, puede configurar `versionStage` o `versionId`.

Para ayudar a proteger al Agente de Secrets Manager, debe incluir un encabezado de token SSRF como parte de cada solicitud: `X-Aws-Parameters-Secrets-Token`. El Agente de Secrets Manager rechaza las solicitudes que no tengan este encabezado o que tengan un token SSRF no válido. Puede personalizar el nombre del encabezado de SSRF en Archivo de configuración.

El agente Secrets Manager usa el AWS SDK para Rust, que usa la cadena de proveedores de credenciales predeterminada. La identidad de estas credenciales de IAM determina los permisos que tiene el Agente de Secrets Manager para recuperar los secretos.

Permisos necesarios:

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

Para obtener más información, consulte Referencia de permisos.

> **Important**
>
> Tras introducir el valor secreto en el Agente de Secrets Manager, cualquier usuario con acceso al entorno informático y al token SSRF podrá acceder al secreto desde la memoria

caché del Agente de Secrets Manager. Para obtener más información, consulte the section called “Consideraciones de seguridad”.

curl

El siguiente ejemplo de curl muestra cómo obtener un secreto del Agente de Secrets Manager. El ejemplo se basa en la presencia de la SSRF en un archivo, que es donde se almacena mediante el script de instalación.

```
curl -v -H \
    "X-Aws-Parameters-Secrets-Token: $(</var/run/awssmatoken)" \
    'http://localhost:2773/secretsmanager/get?secretId=<YOUR_SECRET_ID>'; \
    echo
```

Python

El siguiente ejemplo de Python muestra cómo obtener un valor secreto del Agente de Secrets Manager. El ejemplo se basa en la presencia de la SSRF en un archivo, que es donde se almacena mediante el script de instalación.

```
import requests
import json

# Function that fetches the secret from Secrets Manager Agent for the provided
 secret id.
def get_secret():
    # Construct the URL for the GET request
    url = f"http://localhost:2773/secretsmanager/get?secretId=<YOUR_SECRET_ID>"

    # Get the SSRF token from the token file
    with open('/var/run/awssmatoken') as fp:
        token = fp.read()

    headers = {
        "X-Aws-Parameters-Secrets-Token": token.strip()
    }

    try:
        # Send the GET request with headers
        response = requests.get(url, headers=headers)
```

```
        # Check if the request was successful
        if response.status_code == 200:
            # Return the secret value
            return response.text
        else:
            # Handle error cases
            raise Exception(f"Status code {response.status_code} - {response.text}")

    except Exception as e:
        # Handle network errors
        raise Exception(f"Error: {e}")
```

Actualice los secretos a la fuerza con RefreshNow

Secrets Manager Agent utiliza una caché en memoria para almacenar valores secretos, que actualiza periódicamente. De forma predeterminada, esta actualización se produce cuando se solicita un secreto una vez transcurrido el tiempo de vida (TTL), normalmente cada 300 segundos. Sin embargo, este enfoque a veces puede dar como resultado valores secretos obsoletos, especialmente si un secreto se cambia antes de que caduque la entrada de la caché.

Para solucionar esta limitación, Secrets Manager Agent admite un parámetro llamado `refreshNow` en la URL. Puede utilizar este parámetro para forzar una actualización inmediata del valor de un secreto, omitiendo la caché y asegurándose de disponer de la mayor cantidad de up-to-date información.

Comportamiento predeterminado (sin`refreshNow`)

- Utiliza valores en caché hasta que caduque el TTL
- Actualiza los secretos solo después del TTL (por defecto, 300 segundos)
- Puede devolver valores obsoletos si los secretos rotan antes de que caduque la caché

Comportamiento con `refreshNow=true`

- Omite la memoria caché por completo
- Recupera el último valor secreto directamente de Secrets Manager
- Actualiza la caché con el valor nuevo y restablece el TTL
- Garantiza que siempre obtendrá el valor secreto más actualizado

Al usar el `refreshNow` parámetro, puede asegurarse de trabajar siempre con los valores secretos más actuales, incluso en situaciones en las que sea necesaria la rotación frecuente del secreto.

refreshNowcomportamiento de los parámetros

refreshNow es `true`.

Si Secrets Manager Agent no puede recuperar el secreto de Secrets Manager, devuelve un error y no actualiza la caché.

refreshNowconfigurado como `false` o no especificado

Secrets Manager Agent sigue su comportamiento predeterminado:

- Si el valor en caché es más reciente que el TTL, Secrets Manager Agent devuelve el valor en caché.
- Si el valor en caché es anterior al TTL, Secrets Manager Agent llama a Secrets Manager.

Mediante el parámetro RefreshNow

Para usar el refreshNow parámetro, inclúyalo en la URL de la solicitud GET de Secrets Manager Agent.

Example Ejemplo: solicitud GET de Secrets Manager Agent con el parámetro RefreshNow

> **Important**
>
> El valor predeterminado de `refreshNow` es `false`. Cuando se establece en`true`, anula el TTL especificado en el archivo de configuración del agente Secrets Manager y realiza una llamada de API a Secrets Manager.

curl

El siguiente ejemplo de curl muestra cómo forzar a Secrets Manager Agent a actualizar el secreto. El ejemplo se basa en la presencia de la SSRF en un archivo, que es donde se almacena mediante el script de instalación.

```
curl -v -H \
    "X-Aws-Parameters-Secrets-Token: $(</var/run/awssmatoken)" \
```

```
    'http://localhost:2773/secretsmanager/get?
secretId=<YOUR_SECRET_ID>&refreshNow=true' \
    echo
```

Python

El siguiente ejemplo de Python muestra cómo obtener un valor secreto del Agente de Secrets Manager. El ejemplo se basa en la presencia de la SSRF en un archivo, que es donde se almacena mediante el script de instalación.

```
import requests
import json

# Function that fetches the secret from Secrets Manager Agent for the provided
 secret id.
def get_secret():
    # Construct the URL for the GET request
    url = f"http://localhost:2773/secretsmanager/get?
secretId=<YOUR_SECRET_ID>&refreshNow=true"

    # Get the SSRF token from the token file
    with open('/var/run/awssmatoken') as fp:
        token = fp.read()

    headers = {
        "X-Aws-Parameters-Secrets-Token": token.strip()
    }

    try:
        # Send the GET request with headers
        response = requests.get(url, headers=headers)

        # Check if the request was successful
        if response.status_code == 200:
            # Return the secret value
            return response.text
        else:
            # Handle error cases
            raise Exception(f"Status code {response.status_code} - {response.text}")

    except Exception as e:
        # Handle network errors
        raise Exception(f"Error: {e}")
```

Configuración del Agente de Secrets Manager

Para cambiar la configuración del Agente de Secrets Manager, cree un archivo de configuración TOML y, a continuación, realice una llamada `./aws-secrets-manager-agent --config config.toml`.

La siguiente lista muestra las opciones que puede configurar para el Agente de Secrets Manager.

- log_level: el nivel de detalle indicado en los registros del Agente de Secrets Manager: DEBUG, INFO, WARN, ERROR o NONE. El valor predeterminado es INFO.
- http_port: el puerto del servidor HTTP local, en el rango de 1024 a 65535. El valor predeterminado es 2773.
- región: la AWS región que se utilizará para las solicitudes. Si no se especifica ninguna región, el Agente de Secrets Manager determina la región a partir del SDK. Para obtener más información, consulte Especifique las credenciales y regiones predeterminadas en la Guía para desarrolladores del SDK de AWS para Rust.
- ttl_seconds: el TTL en segundos de los elementos en caché, en el rango de 0 a 3600. El valor predeterminado es 300. 0 indica que no hay almacenamiento en caché.
- cache_size: el número máximo de secretos que se pueden almacenar en la memoria caché, entre 1 y 1000. El valor predeterminado es 1000.
- ssrf_headers: lista de nombres de encabezados que el Agente de Secrets Manager comprueba en busca del token SSRF. El valor predeterminado es «X-Aws-Parameters-Secrets-Token». X-Vault-Token
- ssrf_env_variables: una lista de nombres de variables de entorno que el Agente de Secrets Manager comprueba en busca del token SSRF. La variable de entorno puede contener el token o una referencia al archivo del token, como en: `AWS_TOKEN=file:///var/run/awssmatoken`. El valor AWS_TOKEN predeterminado es '', _TOKEN AWS_SESSION».
- path_prefix: el prefijo URI que se utiliza para determinar si la solicitud es una solicitud basada en una ruta. El valor predeterminado es “/v1/”.
- max_conn: el número máximo de conexiones desde clientes HTTP que permite el Agente de Secrets Manager, entre 1 y 1000. El valor predeterminado es 800.

Registro

El Agente de Secrets Manager registra los errores localmente en el archivo `logs/secrets_manager_agent.log`. Cuando la aplicación llama al Agente de Secrets Manager para

obtener un secreto, esas llamadas aparecen en el registro local. No aparecen en los CloudTrail registros.

El Agente de Secrets Manager crea un nuevo archivo de registro cuando el archivo alcanza los 10 MB y almacena hasta cinco archivos de registro en total.

El registro no va a Secrets Manager, CloudTrail, o CloudWatch. Las solicitudes para obtener secretos del Agente de Secrets Manager no aparecen en esos registros. Cuando el agente de Secrets Manager llama a Secrets Manager para obtener un secreto, esa llamada se graba CloudTrail con una cadena de agente de usuario que contiene`aws-secrets-manager-agent`.

Puede configurar el registro en Archivo de configuración.

Consideraciones de seguridad

En el caso de una arquitectura de agente, el dominio de confianza es el lugar donde se puede acceder al punto de conexión del agente y al token SSRF, que suele ser todo el host. El dominio de confianza del Agente de Secrets Manager debe coincidir con el dominio en el que están disponibles las credenciales de Secrets Manager para mantener la misma postura de seguridad. Por ejemplo, en Amazon, EC2 el dominio de confianza del agente de Secrets Manager sería el mismo que el dominio de las credenciales cuando se utilizan funciones para Amazon EC2.

Las aplicaciones preocupadas por la seguridad que aún no utilizan una solución de agente con las credenciales de Secrets Manager bloqueadas en la aplicación deberían considerar la posibilidad de utilizar soluciones de almacenamiento en caché AWS SDKs o específicas del idioma. Para obtener más información, consulte Obtener secretos.

Obtenga un valor secreto de Secrets Manager con el AWS SDK de C++

En el caso de las aplicaciones de C++, llame al SDK directamente con GetSecretValueo BatchGetSecretValue.

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
//! Retrieve an AWS Secrets Manager encrypted secret.
/*!
  \param secretID: The ID for the secret.
```

```
  \return bool: Function succeeded.
 */
bool AwsDoc::SecretsManager::getSecretValue(const Aws::String &secretID,
                                            const Aws::Client::ClientConfiguration
 &clientConfiguration) {
    Aws::SecretsManager::SecretsManagerClient
 secretsManagerClient(clientConfiguration);

    Aws::SecretsManager::Model::GetSecretValueRequest request;
    request.SetSecretId(secretID);

    Aws::SecretsManager::Model::GetSecretValueOutcome getSecretValueOutcome =
 secretsManagerClient.GetSecretValue(
            request);
    if (getSecretValueOutcome.IsSuccess()) {
        std::cout << "Secret is: "
                  << getSecretValueOutcome.GetResult().GetSecretString() << std::endl;
    }
    else {
        std::cerr << "Failed with Error: " << getSecretValueOutcome.GetError()
                  << std::endl;
    }

    return getSecretValueOutcome.IsSuccess();
}
```

Obtenga un valor secreto de Secrets Manager con el JavaScript AWS SDK

Para JavaScript las aplicaciones, llame al SDK directamente con `getSecretValue`o `batchGetSecretValue`.

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
import {
  GetSecretValueCommand,
  SecretsManagerClient,
} from "@aws-sdk/client-secrets-manager";

export const getSecretValue = async (secretName = "SECRET_NAME") => {
```

```
  const client = new SecretsManagerClient();
  const response = await client.send(
    new GetSecretValueCommand({
      SecretId: secretName,
    }),
  );
  console.log(response);
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: '584eb612-f8b0-48c9-855e-6d246461b604',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   ARN: 'arn:aws:secretsmanager:us-east-1:xxxxxxxxxxxx:secret:binary-
secret-3873048-xxxxxx',
  //   CreatedDate: 2023-08-08T19:29:51.294Z,
  //   Name: 'binary-secret-3873048',
  //   SecretBinary: Uint8Array(11) [
  //      98, 105, 110, 97, 114,
  //     121,  32, 100, 97, 116,
  //      97
  //   ],
  //   VersionId: '712083f4-0d26-415e-8044-16735142cd6a',
  //   VersionStages: [ 'AWSCURRENT' ]
  // }

  if (response.SecretString) {
    return response.SecretString;
  }

  if (response.SecretBinary) {
    return response.SecretBinary;
  }
};
```

Obtén un valor secreto de Secrets Manager con el SDK de Kotlin AWS

Para las aplicaciones de Kotlin, llama al SDK directamente con GetSecretValueo. BatchGetSecretValue

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
suspend fun getValue(secretName: String?) {
    val valueRequest =
        GetSecretValueRequest {
            secretId = secretName
        }

    SecretsManagerClient { region = "us-east-1" }.use { secretsClient ->
        val response = secretsClient.getSecretValue(valueRequest)
        val secret = response.secretString
        println("The secret value is $secret")
    }
}
```

Obtenga un valor secreto de Secrets Manager con el AWS SDK de PHP

Para aplicaciones de PHP, llame al SDK directamente con `GetSecretValue` o `BatchGetSecretValue`.

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
<?php

  /**
    * Use this code snippet in your app.
    *
    * If you need more information about configurations or implementing the sample
 code, visit the AWS docs:
    * https://aws.amazon.com/developer/language/php/
```

```
    */

  require 'vendor/autoload.php';

  use Aws\SecretsManager\SecretsManagerClient;
  use Aws\Exception\AwsException;

  /**
    * This code expects that you have AWS credentials set up per:
    * https://<<{{DocsDomain}}>>/sdk-for-php/v3/developer-guide/guide_credentials.html
    */

  // Create a Secrets Manager Client
  $client = new SecretsManagerClient([
      'profile' => 'default',
      'version' => '2017-10-17',
      'region' => '<<{{MyRegionName}}>>',
  ]);

  $secret_name = '<<{{MySecretName}}>>';

  try {
      $result = $client->getSecretValue([
          'SecretId' => $secret_name,
      ]);
  } catch (AwsException $e) {
      // For a list of exceptions thrown, see
      // https://<<{{DocsDomain}}>>/secretsmanager/latest/apireference/
API_GetSecretValue.html
      throw $e;
  }

  // Decrypts secret using the associated KMS key.
  $secret = $result['SecretString'];

  // Your code goes here
```

Obtenga un valor secreto de Secrets Manager con el AWS SDK de Ruby

Para aplicaciones de Ruby, llame al SDK directamente con `get_secret_value` o `batch_get_secret_value`.

El siguiente ejemplo de código muestra cómo obtener un valor secreto de Secrets Manager.

Permisos necesarios: `secretsmanager:GetSecretValue`

```
  # Use this code snippet in your app.
  # If you need more information about configurations or implementing the sample code,
 visit the AWS docs:
  # https://aws.amazon.com/developer/language/ruby/

  require 'aws-sdk-secretsmanager'

  def get_secret
    client = Aws::SecretsManager::Client.new(region: '<<{{MyRegionName}}>>')

    begin
      get_secret_value_response = client.get_secret_value(secret_id:
 '<<{{MySecretName}}>>')
    rescue StandardError => e
      # For a list of exceptions thrown, see
      # https://<<{{DocsDomain}}>>/secretsmanager/latest/apireference/
API_GetSecretValue.html
      raise e
    end

    secret = get_secret_value_response.secret_string
    # Your code goes here.
  end
```

Obtenga un valor secreto mediante el AWS CLI

Permisos necesarios: `secretsmanager:GetSecretValue`

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
```

Obtenga un grupo de secretos en un lote utilizando el AWS CLI

Permisos necesarios:

- `secretsmanager:BatchGetSecretValue`
- Permiso `secretsmanager:GetSecretValue` para cada uno de los secretos que desea recuperar.
- Si utiliza filtros, también debe tenerlos `secretsmanager:ListSecrets`.

Si desea ver un ejemplo de política de permisos, consulte the section called “Ejemplo: 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.

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"
```

Obtenga un valor secreto con la AWS consola

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 Valor secreto, seleccione 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”.

Usa AWS Secrets Manager secretos en AWS Batch

AWS Batch le ayuda a ejecutar cargas de trabajo de computación por lotes en el Nube de AWS. Con AWS Batch, puede introducir datos confidenciales en sus trabajos almacenándolos en AWS Secrets Manager secreto y luego haciendo referencia a ellos en su definición de trabajo. Para obtener más información, consulte Especificación de información confidencial mediante Secrets Manager.

Obtén un AWS Secrets Manager secreto en un AWS CloudFormation recurso

Con AWS CloudFormation, puedes recuperar un secreto para usarlo en otro AWS CloudFormation recurso. 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 información sobre cómo crear secretos con AWS CloudFormation, consulteAWS CloudFormation.

Para recuperar un secreto de una AWS CloudFormation plantilla, utilice una referencia dinámica. Al crear la pila, la referencia dinámica introduce el valor secreto en el AWS CloudFormation recurso, por lo que no es necesario 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 acceder a un secreto de tu AWS cuenta, puedes usar el nombre secreto. Para acceder a un secreto de otra AWS cuenta, usa el ARN del secreto.

clave-json (Opcional)

El nombre de la clave del par clave-valor cuyo valor desea recuperar. Si no especificas un`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 con una barra invertida (`\`) como valor final. AWS CloudFormation no puede resolver esas referencias, lo que provoca una falla en el recurso.

Usa AWS Secrets Manager secretos en los GitHub trabajos

Para usar un secreto en un GitHub trabajo, puedes usar una GitHub acción para recuperar los 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, consulte 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 the section called “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
    name-transformation: (Optional) uppercase|lowercase|none
    parse-json-secrets: (Optional) true|false
```

Parámetros

`secret-ids`

ARN, nombres y prefijos de nombres de los secretos.

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.

`name-transformation`

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. En el caso de las letras del nombre, puede configurar el paso para usar letras minúsculas con `lowercase` o no cambiar el tipo de letra con `none`. El valor predeterminado es `uppercase`.

`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.

Nombre de variable 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 de los secretos para cumplir 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`. Puede configurar la acción para que no transforme las letras minúsculas.

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 establece `parse-json-secrets: true` e introduce un alias en blanco, seguido de una coma y, a continuación, el ID del 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 del 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 del secreto, la acción asigna un nombre a las variables de entorno como si no hubiera especificado un alias.

El siguiente ejemplo 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

En el siguiente ejemplo, se crean 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"
```

Example 4. Uso de letras minúsculas para los nombres de las variables de entorno

En el siguiente ejemplo, se crea una variable de entorno con un nombre en minúscula.

```
- name: Get secrets
  uses: aws-actions/aws-secretsmanager-get-secrets@v2
  with:
    secret-ids: exampleSecretName
    name-transformation: lowercase
```

Variable de entorno creada:

```
examplesecretname: secretValue
```

Usa AWS Secrets Manager secretos en AWS IoT Greengrass

AWS IoT Greengrass es un software que extiende las capacidades de 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 AWS IoT Greengrass dispositivos sin necesidad de codificar contraseñas, identificadores u otros secretos. Puede utilizarla AWS Secrets Manager para almacenar y gestionar de forma segura sus datos secretos en la nube. AWS IoT Greengrass extiende Secrets Manager a los dispositivos AWS IoT

Greengrass principales, de modo que sus conectores y funciones de Lambda puedan usar secretos locales para interactuar con servicios y aplicaciones.

Para integrar un secreto en un AWS IoT Greengrass grupo, debe crear 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 información sobre cómo crear, administrar y usar recursos secretos, consulte Cómo trabajar con recursos secretos en la Guía para AWS IoT desarrolladores.

Para desplegar secretos en el AWS IoT Greengrass núcleo, consulta Cómo implementar secretos en el AWS IoT Greengrass núcleo.

Usa AWS Secrets Manager secretos en Parameter Store

AWS El almacén de parámetros de Systems Manager proporciona un almacenamiento jerárquico y seguro para la gestión de los datos de configuración y la gestión de 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 Hacer referencia a AWS Secrets Manager los secretos de los parámetros del almacén de parámetros en la Guía del AWS Systems Manager usuario.

Rota AWS Secrets Manager los secretos

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. Hay dos tipos de rotación:

- Rotación administrada: para la mayoría de los secretos administrados, se utiliza la rotación administrada, por la cual el servicio se encarga de configurar y administrar la rotación. La rotación administrada no utiliza una función de Lambda.
- the section called “Rotación con función de Lambda”: para otros tipos de secretos, la rotación de Secrets Manager utiliza una función de Lambda para actualizar el secreto y la base de datos o servicio.

Rotación gestionada de AWS Secrets Manager secretos

Algunos servicios ofrecen rotación administrada, que permite que el servicio se encargue de configurar y administrar la rotación. Con la rotación gestionada, no se utiliza una AWS Lambda función para actualizar el secreto y las credenciales de la base de datos.

Los siguientes servicios ofrecen rotación administrada:

- 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 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 Redshift ofrece rotación administrada 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.

> **Tip**
>
> Para los demás tipos de secretos, consulte the section called “Rotación con función de Lambda”.

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

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 rotación.
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 Save.

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 rotación.

```
aws secretsmanager rotate-secret \
    --secret-id MySecret \
    --rotation-rules \
        "{\"ScheduleExpression\": \"cron(0 16 1,15 * ? *)\", \"Duration\": \"2h\"}"
```

Rotación con función de Lambda

Para muchos tipos de secretos, 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.

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.

Durante la rotación, Secrets Manager registra los eventos que indican el estado de rotación. Para obtener más información, consulte the section called “Inicia sesión con AWS CloudTrail ”.

Para rotar un secreto, Secrets Manager llama a una función Lambda según el programa de rotación que haya configurado. 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.

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",
    "RotationToken" : "string"
}
```

Parámetros:

- Step: el paso de rotación (`create_secret`, `set_secret`, `test_secret` o `finish_secret`). Para obtener más información, consulte the section called “Cuatro pasos en una función de rotación”.

- SecretId— El ARN del secreto para girar.
- ClientRequestToken— Un identificador único para la nueva versión del secreto. Este valor ayuda a garantizar la idempotencia. Para obtener más información, consulte PutSecretValue: ClientRequestToken en la referencia de la AWS Secrets Manager API.
- RotationToken— Un identificador único que indica el origen de la solicitud. Necesario para la rotación secreta mediante un rol asumido o la rotación entre cuentas, en la que se rota un secreto de una cuenta mediante una función de rotación de Lambda en otra cuenta. En ambos casos, la función de rotación asume una función de IAM para llamar a Secrets Manager y, a continuación, Secrets Manager utiliza el token de rotación para validar la identidad de la función de IAM.

Si algún paso de la rotación falla, Secrets Manager vuelve a intentar todo el proceso de rotación varias veces.

Temas

- Configurar la rotación automática de secretos de Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB
- Configurar la rotación automática para secretos que no son de bases de datos AWS Secrets Manager
- Configure la rotación automática mediante el AWS CLI
- Estrategias de rotación de la función de Lambda
- Funciones de rotación de Lambda
- AWS Secrets Manager plantillas de funciones de rotación
- Permisos del rol de ejecución de la función de rotación Lambda para AWS Secrets Manager
- Acceso a la red para la función AWS Lambda de rotación
- Solucionar problemas de rotación AWS Secrets Manager

Configurar la rotación automática de secretos de Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB

En este tutorial, se describe cómo configurar the section called “Rotación con función de Lambda” para los secretos de bases de datos. 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.

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.

> **Warning**
>
> Para activar la rotación automática, debe tener permisos para crear el rol de ejecución de IAM para la función de rotación de Lambda y adjuntarle una política de permisos. Necesita ambos permisos, `iam:CreateRole` y `iam:AttachRolePolicy`. Conceder estos permisos permite que una identidad se conceda a sí misma cualquier permiso.

Pasos:

Paso 1: elegir una estrategia de rotación y (opcionalmente) crear un secreto de superusuario

Para obtener información sobre las estrategias que ofrece Secrets Manager, consulte the section called “Estrategias de rotación de la función de Lambda”.

Si elige la estrategia de usuarios alternativos, debe Crear secretos 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. Tenga en cuenta que Amazon RDS Proxy no admite la estrategia de usuarios alternos.

Paso 2: configurar la rotación y crear una función de rotación

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 rotación.

   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 Save.

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 AWS servicios, para evitar el confuso escenario adjunto, se AWS recomienda utilizar tanto la clave de condición como la `aws:SourceArn`clave de condición `aws:SourceAccount`global. 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 obtener más información, consulte the section called “Acceso a la red para la función AWS Lambda de rotación”.

Pasos a seguir a continuación

Consulte the section called “Solución de problemas de rotación”.

Configurar la rotación automática para secretos que no son de bases de datos AWS Secrets Manager

En este tutorial, se describe cómo configurar los secretos de the section called “Rotación con función de Lambda” que no son de bases de datos. 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.

En el caso de los secretos de bases de datos, consulte Rotación automática de secretos de bases de datos (consola).

> **Warning**
>
> Para activar la rotación automática, debe tener permisos para crear el rol de ejecución de IAM para la función de rotación de Lambda y adjuntarle una política de permisos. Necesita ambos permisos, `iam:CreateRole` y `iam:AttachRolePolicy`. Conceder estos permisos permite que una identidad se conceda a sí misma cualquier permiso.

Pasos:

- Paso 1: crear una función de rotación genérica
- Paso 2: escribir el código de la función de rotación
- Paso 3: configurar el secreto para la rotación
- Paso 4: permitir que la función de rotación acceda a Secrets Manager y a la base de datos o al servicio
- Paso 5: permitir que Secrets Manager invoque la función de rotación
- Paso 6: configurar el acceso a la red para la función de rotación
- Pasos a seguir a continuación

Paso 1: crear una función de rotación genérica

Para comenzar, cree una función de rotación de Lambda. No tendrá el código para rotar su secreto, por lo que tendrá que escribirlo en un paso posterior. Para obtener más información sobre cómo funciona una función de rotación, consulte the section called “Funciones de rotación de Lambda”.

En las regiones compatibles, puede utilizarla AWS Serverless Application Repository para crear la función a partir de una plantilla. Para obtener una lista de las regiones admitidas, consulte AWS Serverless Application Repository FAQs. En otras regiones, se crea la función desde cero y se copia el código de la plantilla en la función.

Crear una función de rotación genérica

1. Para determinar si AWS Serverless Application Repository es compatible en su región, consulte los AWS Serverless Application Repository puntos finales y las cuotas en la Referencia AWS general.
2. Realice una de las siguientes acciones:

- Si AWS Serverless Application Repository es compatible en tu región:

  a. En la consola de Lambda, elija Aplicaciones y, a continuación, seleccione Crear aplicación.

  b. En la página Crear aplicación, seleccione la pestaña Aplicación sin servidor.

  c. En el cuadro de búsqueda, en Aplicaciones públicas, escriba **`SecretsManagerRotationTemplate`**.

  d. Seleccione Mostrar aplicaciones que crean roles de IAM personalizados o políticas de recursos.

  e. Elija el mosaico SecretsManagerRotationTemplate.

  f. En la página Revisar, configurar e implementar, en el mosaico Configuración de la aplicación, complete los campos obligatorios.

     - Para el punto de conexión, introduzca el punto de conexión de su región, incluido **`https://`**. Para obtener una lista de puntos de enlace , consulte the section called “Puntos de conexión de Secrets Manager”.
     - Para colocar la función Lambda en una VPC, incluya los identificadores y. vpcSecurityGroup vpcSubnetIds

  g. Elija Implementar.

- Si AWS Serverless Application Repository no es compatible en tu región:

  a. En la consola de Lambda, seleccione Funciones y elija Crear función.

  b. En la página Create function (Crear función), proceda del modo siguiente:

     i. Elija Crear desde cero.

     ii. En Function name (Nombre de la función), ingrese un nombre para la función de rotación.

     iii. En Runtime (Tiempo de ejecución), elija Python 3.9.

     iv. Elija Crear función.

Paso 2: escribir el código de la función de rotación

En este paso, se escribe el código que actualiza el secreto y el servicio o la base de datos para el que está destinado el secreto. Para obtener información sobre lo que hace una función de rotación, incluidos consejos sobre cómo escribir su propia función de rotación, consulte the section called

"Funciones de rotación de Lambda". También puede utilizar Plantillas de función de rotación como referencia.

Paso 3: configurar el secreto para la rotación

En este paso, establecerá una programación de rotación para su secreto y conectará la función de rotación del secreto.

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 rotación.

   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 Función de rotación, elija la función de Lambda que creó en el paso 1.

   f. Seleccione Save.

Paso 4: permitir que la función de rotación acceda a Secrets Manager y a la base de datos o al servicio

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 5: permitir que Secrets Manager invoque la función de rotación

Para permitir que Secrets Manager invoque la función de rotación en el programa de rotación que haya configurado, debe conceder el permiso `lambda:InvokeFunction` a la entidad principal del servicio Secrets Manager en la política de recursos de la función de Lambda.

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 AWS servicios, para evitar el confuso escenario adjunto, se AWS recomienda utilizar tanto la clave de condición como la `aws:SourceArn`clave de condición `aws:SourceAccount`global. 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.

Para adjuntar una política de recursos a una función de Lambda, consulte Uso de políticas basadas en recursos para Lambda.

La siguiente política permite que Secrets Manager invoque la función de Lambda.

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

```
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "secretsmanager.amazonaws.com"
            },
        "Action": "lambda:InvokeFunction",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "123456789012"
            }
        },
        "Resource": "LambdaRotationFunctionARN"
    }
    ]
}
```

Paso 6: configurar el acceso a la red para la función de rotación

En este paso, permites que la función de rotación se conecte tanto a Secrets Manager como al servicio o base de datos a la que pertenece el secreto. La función de rotación debe poder acceder a ambos para poder rotar el secreto. Consulte the section called “Acceso a la red para la función AWS Lambda de rotación”.

Pasos a seguir a continuación

Cuando se configuró la rotación en el paso 3, se estableció un cronograma para rotar el secreto. Si la rotación falla cuando está programada, Secrets Manager la intentará realizar varias veces. También puede iniciar una rotación inmediatamente siguiendo las instrucciones que se indican en Rotar un secreto inmediatamente.

Si la rotación falla, consulte Solución de problemas de rotación.

Configure la rotación automática mediante el AWS CLI

En este tutorial se describe cómo realizar la configuración the section called “Rotación con función de Lambda” mediante el AWS CLI. 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.

También puede usar la consola para configurar la rotación. En el caso de los secretos de bases de datos, consulte Rotación automática de secretos de bases de datos (consola). Para los demás tipos de secretos, consulte Rotación automática para secretos que no son de bases de datos (consola).

Para configurar la rotación mediante el AWS CLI, si va a rotar un secreto de base de datos, primero debe elegir una 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 Lambda pueda acceder a Secrets Manager y a la base de datos o al servicio a través de la red. Por último, configure el secreto para la rotación.

Pasos:

- Requisito previo para los secretos de la base de datos: elegir una estrategia de rotación
- Paso 1: escribir el código de la función de rotación
- Paso 2: Crear la función de Lambda
- Paso 3: configurar el acceso a la red
- Paso 4: configurar el secreto para la rotación
- Pasos a seguir a continuación

Requisito previo para los secretos de la base de datos: elegir una estrategia de rotación

Para obtener información sobre las estrategias que ofrece Secrets Manager, consulte the section called “Estrategias de rotación de la función de Lambda”.

Opción 1: estrategia de usuario único

Si elige la estrategia de usuario único, puede continuar con el paso 1.

Opción 2: estrategia de usuarios alternos

Si elige la estrategia de usuarios alternos, debe:

- Crear un secreto y almacenar en él las credenciales de superusuario de la base de datos. Necesita un secreto con credenciales de superusuario porque la rotación para usuarios alternos clona el primer usuario, y la mayoría de los usuarios no tienen ese permiso.
- Añadir el ARN del secreto de superusuario al secreto original. Para obtener más información, consulte the section called “Estructura JSON de un secreto”.

Tenga en cuenta que Amazon RDS Proxy no admite la estrategia de usuarios alternos.

Paso 1: 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 obtener más información, consulte the section called “Rotación con función de Lambda”. En este paso, se escribe el código que actualiza el secreto y el servicio o la base de datos para el que está destinado el secreto.

Secrets Manager proporciona plantillas para secretos de bases de datos de Amazon RDS, Amazon Aurora, Amazon Redshift y Amazon DocumentDB en Plantillas de función de rotación.

Escribir el código de la función de rotación

1. Realice una de las siguientes acciones:
   - Consultar la lista de plantillas de funciones de rotación. Si hay alguna que coincida con su estrategia de servicio y rotación, copie el código.
   - Para otros tipos de secretos, escriba su propia función de rotación. Para obtener instrucciones, consulte the section called “Funciones de rotación de Lambda”.
2. Guarde el archivo en un archivo ZIP *`my-function.zip`* junto con las dependencias necesarias.

Paso 2: Crear la función de Lambda

En este paso, se crea la función de Lambda mediante el archivo ZIP que creó en el paso 1. También configura el rol de ejecución de Lambda, que 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 obtener más información y ejemplos, consulte the section called “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 llame al servicio para el que está destinado el secreto, por ejemplo, para crear una contraseña nueva.

2. Crear el rol de ejecución de Lambda y aplicar la política de confianza que creó en el paso anterior 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. 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 python3.7 \
  --zip-file fileb://my-function.zip \
  --handler .handler \
  --role arn:aws:iam::123456789012:role/service-role/rotation-lambda-role
```

4. Establezca una política de recursos en la función de Lambda para permitir que Secrets Manager la invoque mediante una llamada a `lambda add-permission`.

```
aws lambda add-permission \
  --function-name my-rotation-function \
  --action lambda:InvokeFunction \
  --statement-id SecretsManager \
  --principal secretsmanager.amazonaws.com \
  --source-account 123456789012
```

Paso 3: configurar el acceso a la red

Para obtener más información, consulte the section called “Acceso a la red para la función AWS Lambda de rotación”.

Paso 4: 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. Para obtener más información, consulte the section called “Programación de rotación”.

```
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\"}"
```

Pasos a seguir a continuación

Consulte the section called “Solución de problemas de rotación”.

Estrategias de rotación de la función de Lambda

Para the section called “Rotación con función de Lambda”, en el caso de los secretos de base de datos, Secrets Manager ofrece dos estrategias de rotación.

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

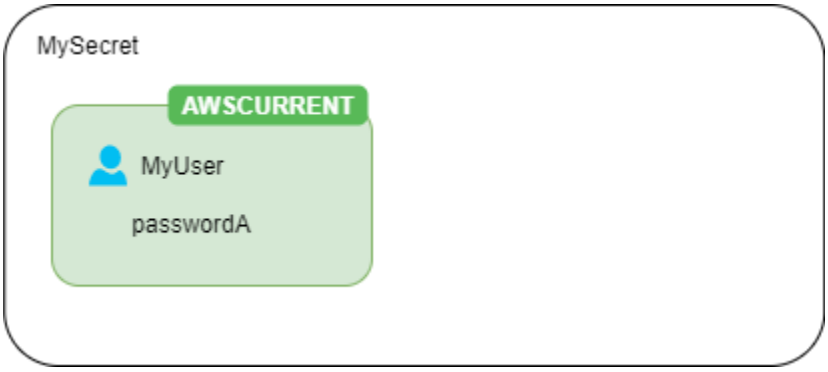
Esta estrategia actualiza las credenciales de dos usuarios en un secreto. Se crea el primer usuario y, durante la primera rotación, la función de rotación lo clona para crear el segundo usuario. Cada vez que el secreto rota, la función de rotación alterna la contraseña de usuario que actualiza. Dado 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).

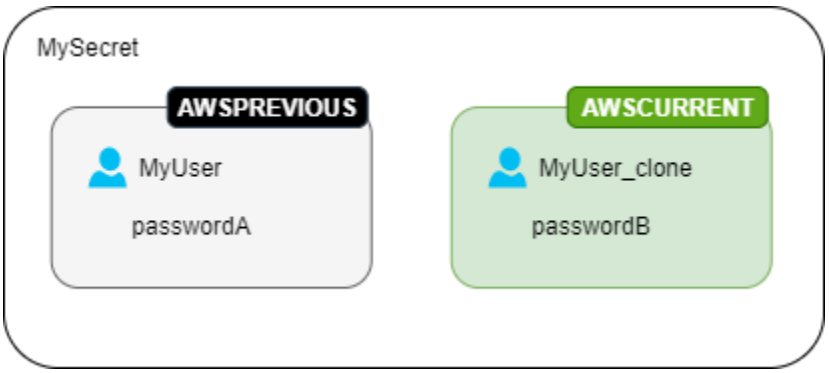
Esa estrategia es adecuada para bases de datos con modelos de permisos en los que un rol es propietario de las tablas de base de datos y un segundo rol tiene permiso para acceder a las tablas de base de datos. También es adecuada para aplicaciones que requieren alta disponibilidad. Si una aplicación 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.

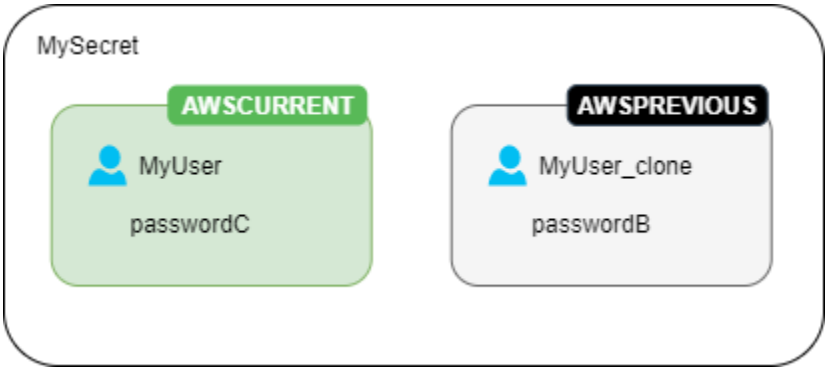
Por ejemplo, si crea un secreto con las credenciales de un usuario de base de datos, el secreto contiene una versión con esas credenciales.



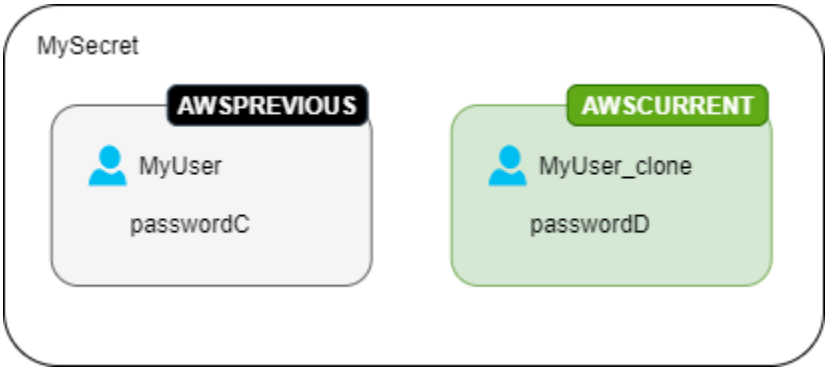
Primera rotación: la función de rotación crea un clon del usuario con una contraseña generada y esas credenciales se convierten en la versión secreta actual.



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



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



Funciones de rotación de Lambda

Enthe section called “Rotación con función de Lambda”, una AWS Lambda función rota el secreto. AWS Secrets Manager utiliza etiquetas de almacenamiento provisional para identificar las versiones secretas durante la rotación.

Si AWS Secrets Manager no proporciona una plantilla de función de rotación para tu tipo secreto, puedes crear una función de rotación personalizada. Sigue estas pautas al escribir la función de rotación:

Prácticas recomendadas para las funciones de rotación personalizadas

- Utilice la plantilla de rotación genérica como punto de partida.
- Tenga cuidado al depurar o registrar sentencias. Pueden escribir información en Amazon CloudWatch Logs. Asegúrese de que los registros no contengan información confidencial.

  Para ver ejemplos de sentencias de registro, consulta el código the section called “Plantillas de función de rotación” fuente.
- Por seguridad, AWS Secrets Manager solo permite que una función de rotación de Lambda gire el secreto directamente. La función de rotación no puede llamar a otra función de Lambda para rotar el secreto.
- Para obtener instrucciones sobre la depuración, consulte Probar y depurar aplicaciones sin servidor.
- Si utilizas bibliotecas y binarios externos, por ejemplo, para conectarte a un recurso, eres responsable de parchearlos y actualizarlos.
- Package la función de rotación y cualquier dependencia en un archivo ZIP, como *`my-function.zip`*.

Cuatro pasos en una función de rotación

Temas

- create_secret: Crear una nueva versión del secreto
- set_secret: Cambiar las credenciales en la base de datos o el servicio
- test_secret: Probar la nueva versión del secreto
- finish_secret: Finalizar la rotación

`create_secret`: Crear una nueva versión del secreto

El método `create_secret` primero comprueba si existe un secreto con una llamada a `get_secret_value` con el valor transmitido de `ClientRequestToken`. Si no hay ningún secreto, crea uno nuevo con `create_secret` y el token como `VersionId`. A continuación, genera un nuevo

valor secreto con `get_random_password`. Luego, llama 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?).

Consejos para escribir su propia función de rotación

- Debe asegurarse de que el nuevo valor secreto solo incluya caracteres válidos para la base de datos o el servicio. Excluya caracteres con el parámetro `ExcludeCharacters`.
- A medida que pruebes la función, usa AWS CLI para ver las etapas de la versión: llama `describe-secret`y mira`VersionIdsToStages`.
- 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.

set_secret: Cambiar las credenciales en la base de datos o el servicio

El método `set_secret` cambia la credencial en la base de datos o el servicio para que coincidan con el nuevo valor secreto en la versión de `AWSPENDING` del secreto.

Consejos para escribir su propia función de rotación

- 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 Query Parameterization Cheat Sheet 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.
- En raras ocasiones, es posible que desee personalizar la función de rotación existente de una base de datos. 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`

test_secret: Probar la nueva versión del secreto

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.

finish_secret: Finalizar la rotación

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. Secrets Manager agrega la etiqueta provisional de `AWSPREVIOUS` a la versión anterior, para que usted conserve la última versión buena conocida del secreto.

El método finish_secret utiliza `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.

Consejos para escribir su propia función de rotación

- No lo elimines `AWSPENDING` antes de este punto ni lo elimines mediante una llamada a la API independiente, ya que eso puede indicarle 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.

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.

AWS Secrets Manager plantillas de funciones de rotación

Para the section called “Rotación con función de Lambda”, Secrets Manager proporciona una cantidad de plantillas de función de rotación. Para utilizar las plantillas, consulte lo siguiente:

- Rotación automática de secretos de bases de datos (consola)
- Rotación automática para secretos que no son de bases de datos (consola)

Las plantillas son compatibles con Python 3.9.

Para escribir su propia función de rotación, consulte Escribir una función de rotación.

Plantillas

- Amazon RDS y Amazon Aurora
  - 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
  - Amazon RDS Microsoft (usuario SQLServer único)
  - Amazon RDS, Microsoft, SQLServer alternancia de usuarios
- Amazon DocumentDB (con compatibilidad con MongoDB)
  - Usuario único de Amazon DocumentDB

- Usuarios alternativos de Amazon DocumentDB
- Amazon Redshift
  - Usuario único de Amazon Redshift
  - Usuarios alternativos de Amazon Redshift
- Amazon Timestream para InfluxDB
  - Usuario único de Amazon Timestream para InfluxDB
  - Usuarios alternos de Amazon Timestream para InfluxDB
- Amazon ElastiCache
- Active Directory
  - Credenciales de Active Directory
  - Teclado de Active Directory
- Otros tipos de secretos

Amazon RDS y Amazon Aurora

Amazon RDS Db2 para un solo usuario

- Nombre de la plantilla: SecretsManager RDSDb2 RotationSingleUser
- Estrategia de rotación: Estrategia de rotación: un solo usuario.
- Estructura de **`SecretString`**: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSDb2RotationSingleUser/lambda _function.py
- Dependencia: python-ibmdb

Usuarios alternos de Amazon RDS Db2

- Nombre de la plantilla: SecretsManager RDSDb2 RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`**: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSDb2RotationMultiUser/lambda _function.py
- Dependencia: python-ibmdb

Un solo usuario de MariaDB en Amazon RDS

- Nombre de la plantilla: SecretsManager RDSMaria DBRotation SingleUser
- Estrategia de rotación: Estrategia de rotación: un solo usuario.
- Estructura de **`SecretString`**: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSMariaDBRotationSingleUser/lambda _function.py
- Dependencia: PyMy SQL 1.0.2. Si utiliza la contraseña sha256 para la autenticación, PyMy SQL [rsa]. Para obtener información sobre el uso de paquetes con código compilado en un entorno de tiempo de ejecución de Lambda, consulte ¿Cómo puedo añadir paquetes de Python con binarios compilados a mi paquete de implementación y hacer que el paquete sea compatible con Lambda? en el Centro de conocimientos de AWS .

Usuarios alternativos de MariaDB en Amazon RDS

- Nombre de la plantilla: SecretsManager RDSMaria DBRotation MultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`**: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSMariaDBRotationMultiUser/lambda _function.py
- Dependencia: PyMy SQL 1.0.2. Si utiliza la contraseña sha256 para la autenticación, PyMy SQL [rsa]. Para obtener información sobre el uso de paquetes con código compilado en un entorno de tiempo de ejecución de Lambda, consulte ¿Cómo puedo añadir paquetes de Python con binarios compilados a mi paquete de implementación y hacer que el paquete sea compatible con Lambda? en el Centro de conocimientos de AWS .

Amazon RDS y Amazon Aurora MySQL para un solo usuario

- Nombre de la plantilla: SecretsManager RDSMy SQLRotation SingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSMySQLRotationSingleUser/lambda _function.py

- Dependencia: PyMy SQL 1.0.2. Si utiliza la contraseña sha256 para la autenticación, PyMy SQL [rsa]. Para obtener información sobre el uso de paquetes con código compilado en un entorno de tiempo de ejecución de Lambda, consulte ¿Cómo puedo añadir paquetes de Python con binarios compilados a mi paquete de implementación y hacer que el paquete sea compatible con Lambda? en el Centro de conocimientos de AWS .

Usuarios alternos de Amazon RDS y Amazon Aurora MySQL

- Nombre de la plantilla: SecretsManager RDSMy SQLRotation MultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSMySQLRotationMultiUser/lambda _function.py
- Dependencia: PyMy SQL 1.0.2. Si utiliza la contraseña sha256 para la autenticación, PyMy SQL [rsa]. Para obtener información sobre el uso de paquetes con código compilado en un entorno de tiempo de ejecución de Lambda, consulte ¿Cómo puedo añadir paquetes de Python con binarios compilados a mi paquete de implementación y hacer que el paquete sea compatible con Lambda? en el Centro de conocimientos de AWS .

Un solo usuario de Oracle en Amazon RDS

- Nombre de la plantilla: SecretsManager RDSOracle RotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSOracleRotationSingleUser/lambda _function.py
- Dependencia: python-oracledb 2.4.1

Usuarios alternativos de Oracle en Amazon RDS

- Nombre de la plantilla: SecretsManager RDSOracle RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.

- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSOracleRotationMultiUser/lambda _function.py
- Dependencia: python-oracledb 2.4.1

Amazon RDS y Amazon Aurora PostgreSQL para un solo usuario

- Nombre de la plantilla: SecretsManager RDSPostgre SQLRotation SingleUser
- Estrategia de rotación: Estrategia de rotación: un solo usuario.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSPostgreSQLRotationSingleUser/lambda _function.py
- Dependencia: PyGre SQL 5.0.7

Usuarios alternos de Amazon RDS y Amazon Aurora PostgreSQL

- Nombre de la plantilla: SecretsManager RDSPostgre SQLRotation MultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSPostgreSQLRotationMultiUser/lambda _function.py
- Dependencia: PyGre SQL 5.0.7

Amazon RDS Microsoft (usuario SQLServer único)

- Nombre de la plantilla: SecretsManager RDSSQLServer RotationSingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSSQLServerRotationSingleUser/lambda _function.py

- Dependencia: Pymssql 2.2.2

Amazon RDS, Microsoft, SQLServer alternancia de usuarios

- Nombre de la plantilla: SecretsManager RDSSQLServer RotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon RDS y Aurora”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRDSSQLServerRotationMultiUser/lambda _function.py
- Dependencia: Pymssql 2.2.2

Amazon DocumentDB (con compatibilidad con MongoDB)

Usuario único de Amazon DocumentDB

- Nombre de la plantilla: SecretsManagerMongo DBRotation SingleUser
- Estrategia de rotación: the section called “Un solo usuario”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon DocumentDB”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerMongoDBRotationSingleUser/lambda _function.py
- Dependencia: Pymongo 3.2

Usuarios alternativos de Amazon DocumentDB

- Nombre de la plantilla: SecretsManagerMongo DBRotation MultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon DocumentDB”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerMongoDBRotationMultiUser/lambda _function.py
- 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”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon Redshift”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRedshiftRotationSingleUser/lambda _function.py
- Dependencia: PyGre SQL 5.0.7

Usuarios alternativos de Amazon Redshift

- Nombre de la plantilla: SecretsManagerRedshiftRotationMultiUser
- Estrategia de rotación: the section called “Usuarios alternativos”.
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Amazon Redshift”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerRedshiftRotationMultiUser/lambda _function.py
- Dependencia: PyGre SQL 5.0.7

Amazon Timestream para InfluxDB

Para usar estas plantillas, consulte Cómo utiliza Amazon Timestream para InfluxDB los secretos en la Guía para desarrolladores de Amazon Timestream.

Usuario único de Amazon Timestream para InfluxDB

- Nombre de plantilla: Influx SecretsManager DBRotation SingleUser
- Estructura de **`SecretString`** esperada: the section called “Estructura secreta de Amazon Timestream para InfluxDB”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerInfluxDBRotationSingleUser/lambda _function.py
- Dependencia: cliente python InfluxDB 2.0

Usuarios alternos de Amazon Timestream para InfluxDB

- Nombre de la plantilla: SecretsManagerInflux DBRotation MultiUser
- Estructura de **`SecretString`** esperada: the section called “Estructura secreta de Amazon Timestream para InfluxDB”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerInfluxDBRotationMultiUser/lambda _function.py
- Dependencia: cliente python InfluxDB 2.0

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 “ ElastiCache Credenciales de Amazon”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerElasticacheUserRotation/lambda _function.py

Active Directory

Credenciales de Active Directory

- Nombre de la plantilla: SecretsManagerActiveDirectoryRotationSingleUser
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Active Directory”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerActiveDirectoryRotationSingleUser/lambda _function.py

Teclado de Active Directory

- Nombre de la plantilla: SecretsManagerActiveDirectoryAndKeytabRotationSingleUser
- Estructura de **`SecretString`** esperada: the section called “Credenciales de Active Directory”.
- Código fuente: https://github.com/aws-samples/ aws-secrets-manager-rotation- lambdas/tree/master/SecretsManagerActiveDirectoryAndKeytabRotationSingleUser/lambda _function.py

- Dependencias: msktutil

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- lambdas/tree/master/SecretsManagerRotationTemplate/lambda _function.py

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

Para the section called “Rotación con función de Lambda”, cuando Secrets Manager utiliza una función de Lambda para rotar un secreto, 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 para secretos que no son de bases de datos (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:

- Ejecute las 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 “Estrategias de rotación de la función de Lambda”.

En el caso de un secreto que contenga credenciales de Amazon RDS, si utiliza la estrategia de usuarios alternativos y Amazon RDS gestiona el secreto del superusuario, también debe permitir que la función de rotación llame en modo de solo lectura APIs en Amazon RDS para que pueda obtener la información de conexión de la base de datos. Te recomendamos que adjuntes la política AWS gestionada de Amazon RDSRead OnlyAccess.

La siguiente política de ejemplo permite a la función:

- Ejecute las 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"
        }
    ]
}
```

Acceso a la red para la función AWS Lambda de rotación

Para the section called “Rotación con función de Lambda”, cuando Secrets Manager usa una función de Lambda para 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 Puntos de conexión de VPC (AWS PrivateLink).

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 su base de datos o servicio se ejecuta en una EC2 instancia de Amazon en una VPC, le recomendamos que configure la función 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.

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 “Rotación con función de Lambda”. 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)
- No hay actividad después de createSecret
- Error: “No se permite el acceso a KMS”
- Error: “Key is missing from secret JSON” (Falta la clave en el JSON del secreto)
- Error: “setSecret: Unable to log into database” (setSecret: no se puede iniciar sesión en la base de datos)
- Error: “No se puede importar el módulo 'lambda_function'”
- Se ha actualizado una función de rotación existente de Python 3.7 a 3.9
- AWS Lambda rotación secreta con error PutSecretValue

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 Puntos de conexión de VPC (AWS PrivateLink).

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:

La red de VPC ACLs no permite la entrada y salida del tráfico HTTPS.

> Para obtener más información, consulte Control del tráfico a las subredes mediante la red ACLs 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 final de la VPC de Secrets Manager no permite que la VPC entre CIDRs 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 the section called “Puntos de conexión de VPC (AWS PrivateLink)”.

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 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.

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 su base de datos o servicio se ejecuta en una EC2 instancia de Amazon en una VPC, le recomendamos que configure la función 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 comando. AWS CLI `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 configuraste la rotación secreta automática antes del 30 de diciembre de 2021, tu función de rotación incluía una versión anterior `libpq` que no es compatible`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 configuraste la rotación secreta automática antes del 20 de diciembre de 2021, es posible que tu función de rotación se base en una plantilla anterior que no era compatibleSSL/TLS. To support connections that use SSL/TLS, por lo que tendrás que volver a crear la función de rotación.

Para determinar cuándo se creó la función de rotación

1. En la consola de Secrets Manager https://console.aws.amazon.com/secretsmanager/, abre tu 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, pega el valor de la clave en el cuadro de búsqueda y, a continuación, selecciona 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 SDKs y 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 AWS Management Console y abra la AWS Lambda consola 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 de Pila de actualizaciones, en Preparar plantilla, elija Editar en Application Composer y, a continuación, en Editar plantilla en Application Composer, elija el botón Editar en Application Composer.
3. En Application Composer, haga lo siguiente:

   a. En el código de la plantilla, en `SecretRotationScheduleHostedRotationLambda`, sustituya el valor para `"functionName": "SecretsManagerTestRotationRDS"` por un nuevo nombre de función, por ejemplo, en JSON, **`"functionName": "SecretsManagerTestRotationRDSupdated"`**

   b. Seleccione Actualizar plantilla.

   c. En el cuadro de diálogo Continuar a AWS CloudFormation, elija Confirmar y continuar a AWS CloudFormation.
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 de Pila de actualizaciones, en Preparar plantilla, elija Editar en Application Composer y, a continuación, en Editar plantilla en Application Composer, elija el botón Editar en Application Composer.
3. En Application Composer, haga lo siguiente:

   a. En la plantilla JSON, para `SecretRotationScheduleHostedRotationLambda`, en `Properties`, en `Parameters`, agregue **`"runtime": "python3.9"`**.

   b. Seleccione Actualizar plantilla.

   c. En el cuadro de diálogo Continuar a AWS CloudFormation, elija Confirmar y continuar a AWS CloudFormation.
4. Continúe con el flujo de trabajo de la AWS CloudFormation pila y, a continuación, seleccione 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.

AWS Lambda rotación secreta con error **PutSecretValue**

Si utiliza un rol asumido o una rotación entre cuentas con Secrets Manager y encuentra un RotationFailed evento en AWS CloudTrail con el mensaje: *LAMBDA_ARN.* Lambda no *SECRET_ARN* creó la versión secreta pendiente *VERSION_ID* para Secret, quite `AWSPENDING` la etiqueta de ensayo y reinicie la rotación, tendrá que actualizar la función de Lambda para usar el parámetro. `RotationToken`

Actualice la función de rotación Lambda para incluir **`RotationToken`**

1. Descargar el código de la función Lambda
   - Abra la consola Lambda
   - En el panel de navegación, elija Funciones
   - Seleccione su función de rotación secreta de Lambda para el nombre de la función
   - Para descargarla, elija una de las siguientes opciones: código de función .zip, AWS SAM archivo o ambos
   - Pulse Aceptar para guardar la función en su máquina local.
2. Editar `Lambda_handler`

   Incluye el parámetro rotation_token en el paso create_secret para la rotación entre cuentas:

```
def lambda_handler(event, context):
    """Secrets Manager Rotation Template

    This is a template for creating an AWS Secrets Manager rotation lambda

    Args:
        event (dict): Lambda dictionary of event parameters. These keys must
 include the following:
            - SecretId: The secret ARN or identifier
            - ClientRequestToken: The ClientRequestToken of the secret version
```

```
            - Step: The rotation step (one of createSecret, setSecret, testSecret,
or finishSecret)
            - RotationToken: the rotation token to put as parameter for
PutSecretValue call

        context (LambdaContext): The Lambda runtime information

    Raises:
        ResourceNotFoundException: If the secret with the specified arn and stage
does not exist

        ValueError: If the secret is not properly configured for rotation

        KeyError: If the event parameters do not contain the expected keys

    """
    arn = event['SecretId']
    token = event['ClientRequestToken']
    step = event['Step']
    # Add the rotation token
    rotation_token = event['RotationToken']

    # Setup the client
    service_client = boto3.client('secretsmanager',
endpoint_url=os.environ['SECRETS_MANAGER_ENDPOINT'])

    # Make sure the version is staged correctly
    metadata = service_client.describe_secret(SecretId=arn)
    if not metadata['RotationEnabled']:
        logger.error("Secret %s is not enabled for rotation" % arn)
        raise ValueError("Secret %s is not enabled for rotation" % arn)
    versions = metadata['VersionIdsToStages']
    if token not in versions:
        logger.error("Secret version %s has no stage for rotation of secret %s." %
(token, arn))
        raise ValueError("Secret version %s has no stage for rotation of secret
%s." % (token, arn))
    if "AWSCURRENT" in versions[token]:
        logger.info("Secret version %s already set as AWSCURRENT for secret %s." %
(token, arn))
        return
    elif "AWSPENDING" not in versions[token]:
        logger.error("Secret version %s not set as AWSPENDING for rotation of
secret %s." % (token, arn))
```

```
        raise ValueError("Secret version %s not set as AWSPENDING for rotation of
 secret %s." % (token, arn))
    # Use rotation_token
    if step == "createSecret":
        create_secret(service_client, arn, token, rotation_token)

    elif step == "setSecret":
        set_secret(service_client, arn, token)

    elif step == "testSecret":
        test_secret(service_client, arn, token)

    elif step == "finishSecret":
        finish_secret(service_client, arn, token)

    else:
        raise ValueError("Invalid step parameter")
```

3. Edita el código `create_secret`

   Revise la `create_secret` función para aceptar y usar el `rotation_token` parámetro:

```
# Add rotation_token to the function
def create_secret(service_client, arn, token, rotation_token):
"""Create the secret

This method first checks for the existence of a secret for the passed in token. If
 one does not exist, it will generate a
new secret and put it with the passed in token.

Args:
service_client (client): The secrets manager service client

arn (string): The secret ARN or other identifier

token (string): The ClientRequestToken associated with the secret version

rotation_token (string): the rotation token to put as parameter for PutSecretValue
 call

Raises:
```

```
ResourceNotFoundException: If the secret with the specified arn and stage does not
 exist

"""
# Make sure the current secret exists
service_client.get_secret_value(SecretId=arn, VersionStage="AWSCURRENT")

# Now try to get the secret version, if that fails, put a new secret
try:
service_client.get_secret_value(SecretId=arn, VersionId=token,
 VersionStage="AWSPENDING")
logger.info("createSecret: Successfully retrieved secret for %s." % arn)
except service_client.exceptions.ResourceNotFoundException:
# Get exclude characters from environment variable
exclude_characters = os.environ['EXCLUDE_CHARACTERS'] if 'EXCLUDE_CHARACTERS' in
 os.environ else '/@"\'\\'
# Generate a random password
passwd = service_client.get_random_password(ExcludeCharacters=exclude_characters)

# Put the secret, using rotation_token
service_client.put_secret_value(SecretId=arn, ClientRequestToken=token,
 SecretString=passwd['RandomPassword'], VersionStages=['AWSPENDING'],
 RotationToken=rotation_token)
logger.info("createSecret: Successfully put secret for ARN %s and version %s." %
 (arn, token))
```

4. Cargue el código de la función Lambda actualizado

   Tras actualizar el código de la función Lambda, cárguelo para rotar su secreto.

Programación de rotación

Secrets Manager rota su secreto durante el periodo de rotación programado que configure. Para configurar la programación y el periodo, utilice una expresión cron() o rate() junto con la duración del periodo. Secrets Manager rota el secreto en cualquier momento durante el periodo de rotación. Se puede rotar un secreto con una frecuencia máxima de cuatro horas en un periodo de rotación de, como mínimo, una hora.

Para activar la rotación, consulte:

- the section called “Rotación administrada”

- the section called “Rotación automática de secretos de bases de datos (consola)”
- the section called “Rotación automática para secretos que no son de bases de datos (consola)”

Las programaciones de rotación de Secrets Manager utilizan la zona horaria UTC.

Periodos de rotación

Un periodo de rotación de Secrets Manager es similar a un periodo de mantenimiento. Se establece el periodo de rotación cuando se quiere rotar el secreto, y Secrets Manager lo hace en algún momento durante ese periodo.

Los periodos de rotación de Secrets Manager siempre comienzan cada hora. En un programa de rotaciones que usa una expresión `rate()` en días, el periodo de rotación se inicia a medianoche. Puede establecer la hora de inicio del periodo de rotación mediante una expresión `cron()`. Para ver ejemplos, consulta the section called “Expresiones cron”.

De forma predeterminada, el periodo de rotación se cierra después de una hora para un programa de rotación en horas, y al final del día para un programa de rotación en días.

Establezca el valor de Duración del periodo para cambiar la duración del periodo de rotación. Puede configurar el intervalo de rotación en, como mínimo, una hora. El periodo de rotación no debe prolongarse hasta el siguiente periodo de rotación. En otras palabras, para un programa de rotación en horas, verifique que el periodo de rotación sea inferior o igual al número de horas entre rotaciones. Para un programa de rotación en días, confirme que la suma de la hora de inicio más la duración del periodo sea inferior o igual a 24 horas.

Expresiones de frecuencia

Las expresiones de tasa de Secrets Manager tienen el siguiente formato, donde *`Value`* es un entero positivo y *`Unit`* puede ser `hourhours`,`day`, o`days`:

```
rate(Value Unit)
```

Se puede rotar un secreto con una frecuencia máxima de cuatro horas. El periodo máximo de rotación es de 999 días. 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.

Expresiones cron

Las expresiones cron de Secrets Manager 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.

Ejemplo de programación	Expression
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 * ? *)`
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 *)`
A partir de enero, cada 11 meses el primer lunes a medianoche.	`cron(0 0 ? 1/11 2#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
Horas	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.
D ay-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. 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

Campos	Valores	Caracteres comodín
		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. Use ***DAY***L para especificar el último día del mes indicado. 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
D ay-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

Rota un AWS Secrets Manager secreto 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. Si no es así, consulte Rotar secretos de .

Para rotar un secreto inmediatamente (consola)

1. Abra la consola de Secrets Manager en https://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 secreto ya debe tener configurada la rotación.

```
$ aws secretsmanager rotate-secret \
    --secret-id MyTestSecret
```

Identificar secretos que no se rotan

Puedes utilizar esta AWS Config herramienta para evaluar tus secretos y comprobar si están rotando de acuerdo con tus normas. Los requisitos internos de seguridad y cumplimiento de los secretos se definen mediante AWS Config reglas. Luego, AWS Config puede identificar los secretos que no se ajustan a sus 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.

Si tiene secretos en varios sitios Cuentas de AWS y Regiones de AWS en su organización, puede agregar esos datos de configuración y cumplimiento. Para obtener más información, consulte Acumulación de datos de varias cuentas y regiones.

Para evaluar si los secretos se están rotando

1. Siga las instrucciones sobre cómo evaluar sus recursos con AWS Config reglas y 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.
2. Si lo desea, AWS Config configúrelo para que le notifique cuando los secretos no sean compatibles. Para obtener más información, consulte el tema Notificaciones que se AWS Config envían a Amazon SNS.

Cancelar la rotación automática en Secrets Manager

Si ha configurado la rotación automática para un secreto y quiere dejar de rotarlo, puede cancelar la rotación.

Cancelar la rotación automática

1. Abra la consola de Secrets Manager en https://console.aws.amazon.com/secretsmanager/.

2. Elija el secreto.
3. En la página de detalles del secreto, en la sección Configuración de rotación, elija Editar rotación.
4. En el cuadro de diálogo Editar configuración de rotación, desactive Rotación automática y, a continuación, seleccione Guardar.

   Secrets Manager conserva la información de configuración de rotación para que pueda utilizarla en el futuro si decide volver a activar la rotación.

AWS Secrets Manager secretos gestionados por otros AWS servicios

Muchos AWS servicios almacenan y utilizan secretos en ellos 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.

> **Note**
>
> Los secretos gestionados solo los puede crear el AWS servicio que los gestiona.

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
```

IDs para los servicios que gestionan 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”
- `pcs` – the section called “AWS Servicio de computación paralela”
- `rds` – the section called “Amazon RDS”
- `redshift` – the section called “Amazon Redshift”

- `sqlworkbench` – the section called “Editor de consultas V2 de Amazon Redshift”

Para buscar secretos gestionados por otros AWS servicios, consulte Buscar secretos gestionados.

Para obtener una lista completa de los servicios que utilizan secretos, consulte Servicios que usan secretos.

AWS servicios que usan AWS Secrets Manager secretos

Obtenga información sobre cómo se integra cada uno de los siguientes Servicios de AWS con Secrets Manager.

- Cómo AWS App Runner usa AWS Secrets Manager
- Cómo usa AWS App2Container AWS Secrets Manager
- ¿Cómo se usa AWS AppConfigAWS Secrets Manager
- Cómo AppFlow usa Amazon AWS Secrets Manager
- ¿Cómo se AWS AppSync usa AWS Secrets Manager
- Cómo Amazon Athena usa AWS Secrets Manager
- Cómo usa Amazon Aurora AWS Secrets Manager
- Cómo los AWS CodeBuild usa AWS Secrets Manager
- Cómo utiliza Amazon Data Firehose AWS Secrets Manager
- ¿Cómo se usa AWS DataSyncAWS Secrets Manager
- Cómo DataZone usa Amazon AWS Secrets Manager
- ¿Cómo se usa AWS Direct ConnectAWS Secrets Manager
- ¿Cómo se usa AWS Directory ServiceAWS Secrets Manager
- Cómo Amazon DocumentDB (con compatibilidad con MongoDB) usa AWS Secrets Manager
- ¿Cómo se usa AWS Elastic BeanstalkAWS Secrets Manager
- Cómo utiliza Amazon Elastic Container Registry AWS Secrets Manager
- Amazon Elastic Container Service
- Cómo ElastiCache usa Amazon AWS Secrets Manager
- ¿Cómo se AWS Elemental Live usa AWS Secrets Manager
- ¿Cómo se AWS Elemental MediaConnect usa AWS Secrets Manager
- ¿ AWS Elemental MediaConvert Cómo se usa AWS Secrets Manager
- ¿Cómo se usa AWS Elemental MediaLiveAWS Secrets Manager
- ¿Cómo se AWS Elemental MediaPackage usa AWS Secrets Manager
- ¿Cómo se AWS Elemental MediaTailor usa AWS Secrets Manager
- La forma en la que Amazon EMR utiliza Secrets Manager

- Cómo EventBridge usa Amazon AWS Secrets Manager
- Cómo FSx usa Amazon AWS Secrets Manager los secretos
- ¿Cómo se usa AWS Glue DataBrewAWS Secrets Manager
- Cómo usa AWS Glue Studio AWS Secrets Manager
- ¿Cómo se usa AWS IoT SiteWiseAWS Secrets Manager
- Cómo usa Amazon Kendra AWS Secrets Manager
- Cómo utiliza Amazon Kinesis Video Streams AWS Secrets Manager
- ¿Cómo se usa AWS Launch WizardAWS Secrets Manager
- Cómo Amazon Lookout for Metrics usa AWS Secrets Manager
- Cómo utiliza Amazon Managed Grafana AWS Secrets Manager
- ¿Cómo se usa AWS Managed ServicesAWS Secrets Manager
- Cómo Amazon Managed Streaming for Apache Kafka usa AWS Secrets Manager
- Cómo utiliza Amazon Managed Workflows for Apache Airflow AWS Secrets Manager
- AWS Marketplace
- ¿Cómo se AWS Migration Hub usa AWS Secrets Manager
- Cómo AWS Panorama usa Secrets Manager
- Cómo utiliza AWS Parallel Computing Service AWS Secrets Manager
- ¿Cómo se AWS ParallelCluster usa AWS Secrets Manager
- Cómo Amazon Q utiliza Secrets Manager
- ¿Cómo se AWS OpsWorks for Chef Automate usa AWS Secrets Manager
- Cómo QuickSight usa Amazon AWS Secrets Manager
- Cómo utiliza Amazon RDS AWS Secrets Manager
- Cómo utiliza Amazon Redshift AWS Secrets Manager
- Amazon Redshift Query Editor v2
- Cómo usa Amazon SageMaker AI AWS Secrets Manager
- ¿Cómo se usa AWS Schema Conversion ToolAWS Secrets Manager
- Cómo utiliza Amazon Timestream para InfluxDB AWS Secrets Manager
- ¿Cómo se usa AWS Toolkit for JetBrainsAWS Secrets Manager
- ¿Cómo AWS Transfer Family usa AWS Secrets Manager los secretos

- ¿Cómo AWS Wickr usa AWS Secrets Manager los secretos

Cómo AWS App Runner usa AWS Secrets Manager

AWS App Runner es un AWS servicio que proporciona una forma rápida, sencilla y rentable de implementar desde el código fuente o una imagen de contenedor directamente a una aplicación web escalable y segura en la AWS nube. No necesita aprender nuevas tecnologías, decidir qué servicio de cómputo usar ni saber cómo aprovisionar y configurar AWS los recursos.

Con App Runner, se puede hacer referencia a secretos y configuraciones en forma de variables de entorno en un servicio cuando se crea un servicio o se actualiza la configuración del servicio. Para obtener más información, consulte Referencing environment variables (Referencia a variables de entorno) y Managing environment variables (Administración de variables de entorno) en la Guía para desarrolladores de AWS App Runner .

Cómo usa AWS App2Container AWS Secrets Manager

AWS App2Container es una herramienta de línea de comandos que le ayuda a seleccionar y cambiar las aplicaciones que se ejecutan en sus centros de datos locales o en máquinas virtuales, de modo que se ejecuten en contenedores gestionados por Amazon ECS, Amazon EKS o AWS App Runner.

App2Container utiliza Secrets Manager para administrar las credenciales para conectar el equipo de trabajo a los servidores de aplicaciones con el fin de ejecutar comandos remotos. Para obtener más información, consulte Administrar los secretos de AWS App2Container en la Guía del AWS usuario de App2Container.

¿Cómo se usa AWS AppConfigAWS Secrets Manager

AWS AppConfig es una capacidad AWS Systems Manager que puede utilizar para crear, administrar e implementar rápidamente configuraciones de aplicaciones. Una configuración puede contener datos de credenciales u otra información confidencial almacenada en Secrets Manager. Al crear un perfil de configuración de formato libre, puede elegir Secrets Manager como origen de los datos de configuración. Para obtener más información, consulte Creating a freeform configuration profile (Creación de un perfil de configuración de formato libre) en la Guía del usuario de AWS AppConfig . Para obtener información sobre cómo AWS AppConfig gestiona los secretos que tienen activada la rotación automática, consulte la rotación de claves de Secrets Manager en la Guía del AWS AppConfig usuario.

Cómo AppFlow usa Amazon AWS Secrets Manager

Amazon AppFlow es un servicio de integración totalmente gestionado que le permite intercambiar datos de forma segura entre aplicaciones de software como servicio (SaaS), como Salesforce, Servicios de AWS y Amazon Simple Storage Service (Amazon S3) y Amazon Redshift.

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.

¿Cómo se AWS AppSync usa AWS Secrets Manager

AWS AppSync proporciona una interfaz GraphQL sólida y escalable para que los desarrolladores de aplicaciones combinen datos de varias fuentes, incluidas Amazon DynamoDB AWS Lambda y HTTP. APIs

AWS AppSync usa las credenciales de un secreto de Secrets Manager para conectarse a Amazon RDS y Aurora. Para obtener más información, consulte Tutorial: Aurora sin servidor en la Guía para desarrolladores de AWS AppSync .

Cómo Amazon Athena usa AWS Secrets Manager

Amazon Athena es un servicio de consultas interactivo que facilita el análisis de datos directamente en Amazon Simple Storage Service (Amazon S3) con SQL estándar.

Los conectores de origen de datos de Amazon Athena pueden utilizar la característica de consulta federada de Athena con secretos de Secrets Manager para consultar datos. Para obtener más información, consulte Uso de consulta federada de Amazon Athena en la Guía del usuario de Amazon Athena.

Cómo usa Amazon Aurora AWS Secrets Manager

Amazon Aurora es un motor de base de datos relacional completamente administrado compatible con MySQL y PostgreSQL.

Para administrar las credenciales de usuario maestro de Aurora, el servicio puede crear un secreto administrado para usted. Se le cobrará ese secreto. Aurora también administra la rotación de estas credenciales. Para obtener más información, consulte Administración de contraseñas con Amazon Aurora y AWS Secrets Manager en la Guía del usuario de Amazon Aurora.

Para obtener otras credenciales de Aurora, consulte Crear secretos.

Al llamar a la API de datos de Amazon RDS, puede transferir las credenciales para la base de datos mediante un secreto en Secrets Manager. Para obtener más información, consulte la sección de Uso de la API de datos para Aurora Serverless en la Guía del usuario de Amazon Aurora.

Cuando utiliza Amazon RDS Query Editor para conectarse a una base de datos, puede almacenar las credenciales de la base de datos en Secrets Manager. Para obtener más información, consulte Uso del editor de consultas en la Guía del usuario de Amazon RDS.

Cómo los AWS CodeBuild usa AWS Secrets Manager

AWS CodeBuild es un servicio de compilación totalmente gestionado en la nube. CodeBuild compila el código fuente, ejecuta pruebas unitarias y produce artefactos listos para su despliegue.

Puede almacenar sus credenciales de registro privado con Secrets Manager. Para obtener más información, consulte Registro privado con AWS Secrets Manager ejemplos CodeBuild en la Guía del AWS CodeBuild usuario.

Cómo utiliza Amazon Data Firehose AWS Secrets Manager

Puede usar Amazon Data Firehose para entregar datos de streaming en tiempo real a varios destinos de streaming. Cuando el destino requiere una credencial o una clave, Firehose recupera un secreto de Secrets Manager en tiempo de ejecución para conectarse al destino. Para obtener más información, consulte Autenticarse con Amazon Data Firehose AWS Secrets Manager en la guía para desarrolladores de Amazon Data Firehose.

¿Cómo se usa AWS DataSyncAWS Secrets Manager

AWS DataSync es un servicio de transferencia de datos en línea que simplifica, automatiza y acelera la transferencia de datos entre sistemas y servicios de almacenamiento. DataSync Discovery le ayuda a acelerar su migración a. AWS

Para recopilar información sobre un sistema de almacenamiento local, 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 DataSync usuario.

Cómo DataZone usa Amazon AWS Secrets Manager

Amazon DataZone es un servicio de administración de datos que le permite catalogar, descubrir, gobernar, compartir y analizar sus datos. Puede usar activos de datos de tablas y vistas de un clúster de Amazon Redshift que se rastrea mediante un trabajo. Rastreador de AWS Glue Para conectarse a Amazon Redshift, debe proporcionar DataZone las credenciales de Amazon en un secreto de Secrets Manager. Para obtener más información, consulte Crear una fuente de datos para una base de datos de Amazon Redshift mediante una nueva AWS Glue conexión en la Guía DataZone del usuario de Amazon.

¿Cómo se usa AWS Direct ConnectAWS Secrets Manager

AWS Direct Connect conecta la red interna a una AWS Direct Connect ubicación a través de un cable Ethernet de fibra óptica estándar. Con esta conexión, puede crear interfaces virtuales directamente a los Servicios de AWS públicos.

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 gestionado con el prefijo. `directconnect` El coste del secreto está incluido en el precio. AWS Direct Connect Para actualizar el secreto, debes usar Secrets Manager AWS Direct Connect en lugar de Secrets Manager. Para obtener más información, consulte Asociar un MACsec CKN/CAK a un LAG en la Guía del AWS Direct Connect usuario.

¿Cómo se usa AWS Directory ServiceAWS Secrets Manager

AWS Directory Service proporciona varias formas de utilizar Microsoft Active Directory (AD) con otros AWS servicios. Puedes unir una EC2 instancia de Amazon a tu directorio utilizando secretos como credenciales. Para obtener más información, consulte lo siguiente en la Guía del usuario de AWS Direct Connect :

- Une sin problemas una EC2 instancia de Linux a tu directorio AWS gestionado de Microsoft AD

- Une sin problemas una EC2 instancia de Linux a tu directorio de AD Connector
- Une sin problemas una EC2 instancia de Linux a tu directorio Simple AD

Cómo Amazon DocumentDB (con compatibilidad con MongoDB) usa AWS Secrets Manager

Amazon DocumentDB (compatible con MongoDB) es un servicio de base de datos de documentos totalmente gestionado que admite cargas de trabajo de MongoDB. Amazon DocumentDB se integra con Secrets Manager para administrar las contraseñas de los usuarios principales de sus clústeres, lo que mejora la seguridad y simplifica la administración de credenciales.

Amazon DocumentDB genera la contraseña, la almacena en Secrets Manager y administra la configuración secreta. De forma predeterminada, Amazon DocumentDB rota el secreto cada siete días, pero puede modificar el programa de rotación si es necesario. Al crear o modificar un clúster de Amazon DocumentDB, puede especificar que debe administrar la contraseña del usuario principal en Secrets Manager. Para obtener más información, consulte Administración de contraseñas con Amazon DocumentDB y Secrets Manager en la Guía para desarrolladores de Amazon DocumentDB.

¿Cómo se usa AWS Elastic BeanstalkAWS Secrets Manager

Con AWS Elastic Beanstalkél, puede implementar y administrar aplicaciones rápidamente en la AWS nube sin tener que conocer la infraestructura en la que se ejecutan esas aplicaciones. Elastic Beanstalk puede lanzar entornos de Docker si se crea una imagen descrita en un Dockerfile o se extrae una imagen de Docker remota. Para autenticarse con el registro en línea que aloja el repositorio privado, Elastic Beanstalk usa un secreto de Secrets Manager. Para obtener más información, consulte Docker configuration en la Guía para desarrolladores de AWS Elastic Beanstalk .

Cómo utiliza Amazon Elastic Container Registry AWS Secrets Manager

Amazon Elastic Container Registry (Amazon ECR) es AWS un servicio gestionado de registro de imágenes de contenedores seguro, escalable y fiable. Puede utilizar la CLI de Docker, o su cliente favorito, para insertar imágenes de sus repositorios y extraerlas desde estos. Para cada registro ascendente que contenga imágenes que desee almacenar en caché en su registro privado de

Amazon ECR, debe crear una regla de caché de extracción. En el caso de los registros originales que requieren autenticación, debe almacenar las credenciales en un secreto de Secrets Manager. Puede crear el secreto de Secrets Manager en las consolas Amazon ECR o Secrets Manager. Para obtener más información, consulte Crear una regla de caché de extracción en la Guía del usuario de Amazon ECR.

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

Amazon ECS admite FSx volúmenes de Windows File Server para contenedores. Amazon ECS utiliza las credenciales almacenadas en un secreto de Secrets Manager para unirse al dominio de Active Directory y adjuntar el FSx sistema de archivos del servidor de archivos de Windows. Para obtener más información, consulte el tutorial: Uso FSx de los sistemas de archivos de Windows File Server con Amazon ECS y FSx para los volúmenes de Windows File Server en la Guía para desarrolladores de Amazon Elastic Container Service.

Puede hacer referencia a imágenes de contenedores en registros privados AWS que no requieran autenticación mediante el uso de un secreto de Secrets Manager con las credenciales del registro. Para obtener más información, consulte Autenticación de registros privados para tareas en la Guía para desarrolladores de Amazon Elastic Container Service.

Cuando se utiliza Amazon ECS Service Connect, Amazon ECS utiliza los secretos administrados de Secrets Manager para almacenar los certificados TLS de AWS Private Certificate Authority . El costo de almacenar el secreto está incluido en el cargo por Amazon ECS. Para actualizar el secreto, debe usar 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.

Cómo ElastiCache usa Amazon AWS Secrets Manager

ElastiCache Puede utilizar una función llamada Control de acceso basado en roles (RBAC) para proteger el clúster. Se pueden almacenar estas credenciales en Secrets Manager. Secrets Manager proporciona una plantilla de rotación para este tipo de secreto. Para obtener más información, consulta Rotación automática de contraseñas para los usuarios en la Guía del ElastiCache usuario de Amazon.

¿Cómo se AWS Elemental Live usa AWS Secrets Manager

AWS Elemental Live es un servicio de vídeo en tiempo real que permite crear salidas en directo para su emisión y distribución en streaming.

AWS Elemental Live usa un ARN secreto para obtener un secreto que contiene una clave de cifrado de Secrets Manager. Elemental Live utiliza la clave de cifrado para cifrar/descifrar el vídeo. Para obtener más información, consulte Cómo MediaConnect funciona la entrega desde AWS Elemental Live a en tiempo de ejecución en la Guía del usuario de Elemental Live.

¿Cómo se AWS Elemental MediaConnect usa AWS Secrets Manager

AWS Elemental MediaConnect es un servicio que permite a las emisoras y otros proveedores de vídeo premium incorporar vídeo en directo de forma fiable Nube de AWS y distribuirlo a varios destinos dentro o fuera del país. Nube de AWS

Puede utilizar el cifrado de clave estática para proteger los orígenes, salidas y concesiones de derechos, y almacenar la clave de cifrado en AWS Secrets Manager. Para obtener más información, consulte Cifrado de clave estática en AWS Elemental MediaConnect en la Guía del usuario de AWS Elemental MediaConnect .

¿ AWS Elemental MediaConvert Cómo se usa AWS Secrets Manager

AWS Elemental MediaConvert es un servicio de procesamiento de vídeo basado en archivos que proporciona un procesamiento de vídeo escalable para propietarios y distribuidores de contenido con bibliotecas multimedia de cualquier tamaño. MediaConvert Para codificar las marcas de agua de Kantar, utilizas Secrets Manager para almacenar tus credenciales de Kantar. Para obtener

más información, consulta Cómo usar Kantar como marca de agua de audio en AWS Elemental MediaConvert las salidas de la Guía del usuario.AWS Elemental MediaConvert

¿Cómo se usa AWS Elemental MediaLiveAWS Secrets Manager

AWS Elemental MediaLive es un servicio de vídeo en tiempo real que permite crear salidas en directo para su emisión y distribución en streaming. Si su organización usa AWS Elemental Link dispositivos con AWS Elemental MediaLive o AWS Elemental MediaConnect, debe implementar el dispositivo y configurarlo. Para obtener más información, consulte Configuración MediaLive como entidad de confianza en la Guía del MediaLive usuario.

¿Cómo se AWS Elemental MediaPackage usa AWS Secrets Manager

AWS Elemental MediaPackage es un servicio just-in-time de empaquetado y creación de vídeos que se ejecuta en. Nube de AWS Con MediaPackage él, puede ofrecer transmisiones de vídeo altamente seguras, escalables y confiables a una amplia variedad de dispositivos de reproducción y redes de entrega de contenido (CDNs). Para obtener más información, consulte Acceso a Secrets Manager para autorización de CDN en la Guía del usuario de AWS Elemental MediaPackage .

¿Cómo se AWS Elemental MediaTailor usa AWS Secrets Manager

AWS Elemental MediaTailor es un servicio escalable de inserción de anuncios y ensamblaje de canales que se ejecuta en Nube de AWS.

MediaTailor admite la autenticación mediante token de acceso de Secrets Manager a sus ubicaciones de origen. Con la autenticación mediante token de acceso a Secrets Manager, MediaTailor utiliza un secreto de Secrets Manager para autenticar las solicitudes que llegan a tu origen. Para obtener más información, consulte Configurar la autenticación con token de AWS Secrets Manager acceso en la Guía del AWS Elemental MediaTailor usuario.

La forma en la que Amazon EMR utiliza Secrets Manager

Amazon EMR es una plataforma que simplifica la ejecución de marcos de big data, como Apache Hadoop y Apache Spark, AWS para procesar y analizar grandes cantidades de datos. Cuando utiliza estos marcos de trabajo y proyectos de código abierto relacionados, como Apache Hive y Apache Pig, puede procesar datos para cargas de trabajo de análisis e inteligencia empresarial. También

puede utilizar Amazon EMR para transformar y mover grandes cantidades de datos dentro y fuera de otros almacenes de datos y bases de AWS datos, como Amazon S3 y Amazon DynamoDB.

Cómo EC2 utiliza Secrets Manager Amazon EMR que se ejecuta en Amazon

Al crear un clúster en Amazon EMR, puede proporcionar datos de configuración de la aplicación al clúster mediante un secreto en Secrets Manager. Para obtener más información, consulte Almacenamiento de datos de configuración confidenciales en Secrets Manager en la Guía de administración de Amazon EMR.

Además, cuando crea un cuaderno de EMR, puede almacenar sus credenciales de registro privado basadas en Git con Secrets Manager. Para obtener más información consulte Add a Git-based Repository to Amazon EMR (Agregar un repositorio basado en Git a Amazon EMR) en la Guía de administración de Amazon EMR.

La forma en la que EMR sin servidor utiliza Secrets Manager

EMR sin servidor le ofrece un entorno de tiempo de ejecución sin servidor para simplificar el funcionamiento de las aplicaciones de análisis, de modo que no tenga que configurar, optimizar, proteger ni operar clústeres.

Puede almacenar sus datos AWS Secrets Manager y, a continuación, utilizar el ID secreto en sus configuraciones EMR Serverless. De esta forma, no pasa los datos de configuración confidenciales en texto plano y los expone a fuentes externas. APIs

Para obtener más información, consulteSecrets Manager para la protección de datos con EMR sin servidoren laGuía del usuario de Amazon EMR sin servidor.

Cómo EventBridge usa Amazon AWS Secrets Manager

Amazon EventBridge es un servicio de bus de eventos sin servidor que puede utilizar para conectar sus aplicaciones con datos de diversas fuentes.

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.

Cómo FSx usa Amazon AWS Secrets Manager los secretos

Amazon FSx for Windows File Server proporciona servidores de archivos Microsoft Windows totalmente gestionados, respaldados por un sistema de archivos Windows totalmente nativo. Al crear o administrar archivos compartidos, puede transferir las credenciales de un archivo AWS Secrets Manager secreto. Para obtener más información, consulte Recursos compartidos de archivos y Migración de configuraciones de recursos compartidos de archivos a Amazon FSx en la Guía del usuario de Amazon FSx para Windows File Server.

¿Cómo se usa AWS Glue DataBrewAWS Secrets Manager

AWS Glue DataBrew es una herramienta visual de preparación de datos que se puede utilizar para limpiar y normalizar los datos sin necesidad de escribir código. En DataBrew, un conjunto de pasos de transformación de datos se denomina receta. AWS Glue DataBrew proporciona los `DETERMINISTIC_DECRYPT`pasos y la `CRYPTOGRAPHIC_HASH`receta 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. `DETERMINISTIC_ENCRYPT` Si usa el secreto DataBrew predeterminado para almacenar la clave de cifrado, DataBrew crea un secreto administrado con el prefijo`databrew`. El coste de almacenar el secreto está incluido en el coste de su uso DataBrew. Si crea un secreto nuevo para almacenar la clave de cifrado, DataBrew crea un secreto con el prefijo`AwsGlueDataBrew`. Se le cobrará ese secreto.

Cómo usa AWS Glue Studio AWS Secrets Manager

AWS Glue Studio es una interfaz gráfica que facilita la creación, la ejecución y la supervisión de los trabajos de extracción, transformación y carga (ETL) AWS Glue. Puede usar Amazon OpenSearch Service como almacén de datos para sus trabajos de extracción, transformación y carga (ETL) configurando el Elasticsearch Spark Connector en. AWS Glue Studio Para conectarte al OpenSearch clúster, puedes usar un secreto en Secrets Manager. Para obtener más información, consulta el Tutorial: Uso del AWS Glue Connector para Elasticsearch en la Guía para AWS Glue desarrolladores.

¿Cómo se usa AWS IoT SiteWiseAWS Secrets Manager

AWS IoT SiteWise es un servicio gestionado que le permite recopilar, modelar, analizar y visualizar datos de equipos industriales a escala. Puede usar la AWS IoT SiteWise consola para crear

una puerta de enlace. A continuación, agregue orígenes de datos, servidores locales o equipos industriales conectados a puertas de enlace. Si el origen requiere autenticación, utilice un secreto para autenticarse. Para más información, consulte Configuring data source authentication (Configuración de la autenticación de orígenes de datos) en la Guía del usuario de AWS IoT SiteWise .

Cómo usa Amazon Kendra AWS Secrets Manager

Amazon Kendra es un servicio de búsqueda inteligente y de alta precisión que permite a los usuarios buscar datos no estructurados y estructurados mediante el procesamiento de lenguaje natural y los algoritmos de búsqueda avanzados.

Puede indexar documentos almacenados en una base de datos especificando un secreto que contenga credenciales para la base de datos. Para obtener más información, consulte Uso de un origen de datos de base de datos en la Guía del usuario de Amazon Kendra.

Cómo utiliza Amazon Kinesis Video Streams AWS Secrets Manager

Puede utilizar Amazon Kinesis Video Streams para conectarse a las cámaras IP de las instalaciones del cliente, grabar y almacenar de manera local el video de las cámaras y transmitir videos a la nube para su almacenamiento, reproducción y procesamiento analítico a largo plazo. Para grabar y cargar contenido multimedia desde cámaras IP, implemente el Kinesis Video Streams Edge Agent en AWS IoT Greengrass. Las credenciales necesarias para acceder a los archivos multimedia que se transmiten a la cámara se almacenan en un secreto de Secrets Manager. Para obtener más información, consulte Deploy the Amazon Kinesis Video Streams Edge Agent to AWS IoT Greengrass en la Guía para desarrolladores de Amazon Kinesis Video Streams.

¿Cómo se usa AWS Launch WizardAWS Secrets Manager

AWS Launch Wizard for Active Directory es un servicio que aplica las prácticas recomendadas de las Nube de AWS aplicaciones para guiarlo a la hora de configurar una nueva infraestructura de Active Directory o de agregar controladores de dominio a una infraestructura existente, ya sea local Nube de AWS o local.

AWS Launch Wizard requiere que se agreguen credenciales de administrador de dominio a Secrets Manager para unir los controladores de dominio a Active Directory. Para obtener más información,

consulte Configurar AWS Launch Wizard para Active Directory en la Guía del AWS Launch Wizard usuario.

Cómo Amazon Lookout for Metrics usa AWS Secrets Manager

Amazon Lookout for Metrics es un servicio que busca anomalías en los datos, determina sus causas raíz y le permite tomar acción rápidamente. Puede utilizar Amazon Redshift o Amazon RDS como origen de datos para un detector Lookout for Metrics. Para configurar el origen de datos, se utiliza un secreto que contiene la contraseña de la base de datos. Para obtener más información, consulte Using Amazon RDS with Lookout for Metrics (Uso de Amazon RDS con Lookout for Metrics) y Using Amazon Redshift with Lookout for Metrics (Uso de Amazon Redshift con Lookout for Metrics) en la Guía para desarrolladores de Amazon Lookout for Metrics.

Cómo utiliza Amazon Managed Grafana AWS Secrets Manager

Amazon Managed Grafana es un servicio de visualización de datos seguro y completamente administrado que puede utilizar para consultar, correlacionar y visualizar al instante métricas operativas, registros y seguimientos de varios orígenes. Cuando utiliza Amazon Redshift como fuente de datos, puede proporcionar las credenciales de Amazon Redshift mediante un secreto. AWS Secrets Manager Para obtener más información, consulte Configuración de Amazon Redshift en la Guía del usuario de Amazon Managed Grafana.

¿Cómo se usa AWS Managed ServicesAWS Secrets Manager

AWS Managed Services es un servicio empresarial que proporciona una administración continua de su AWS infraestructura. El modo de aprovisionamiento autoservicio (SSP) de AMS proporciona acceso completo a las capacidades nativas de Servicio de AWS y la API en las cuentas administradas por AMS. Para obtener información sobre cómo solicitar acceso a Secrets Manager en AMS, consulte AWS Secrets Manager (aprovisionamiento autoservicio de AMS) en la Guía del usuario avanzado de AMS.

Cómo Amazon Managed Streaming for Apache Kafka usa AWS Secrets Manager

Amazon Managed Streaming for Apache Kafka (Amazon MSK) es un servicio totalmente administrado que permite crear y ejecutar aplicaciones que utilizan Apache Kafka para procesar datos de streaming. Puede controlar el acceso a los clústeres de Amazon MSK utilizando nombres

de usuario y contraseñas que se almacenan y protegen mediante AWS Secrets Manager. Para obtener más información, consulte Autenticación de usuario y contraseña con AWS Secrets Manager en la Guía para desarrolladores de Amazon Managed Streaming for Apache Kafka.

Cómo utiliza Amazon Managed Workflows for Apache Airflow AWS Secrets Manager

Amazon Managed Workflows for Apache Airflow es un servicio de organización gestionado para Apache Airflow que facilita la configuración y el funcionamiento de las canalizaciones de end-to-end datos en la nube a escala.

Puede configurar una conexión de Apache Airflow mediante un secreto de Secrets Manager. Para obtener más información, 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

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 coste de almacenar el secreto está incluido en los gastos AWS Marketplace. Para actualizar el secreto, debes usar Secrets Manager AWS Marketplace en lugar de Secrets Manager. Para obtener más información, consulte la configuración de Inicio Rápido en la AWS Marketplace Guía del Vendedor.

¿Cómo se AWS Migration Hub usa AWS Secrets Manager

AWS Migration Hub proporciona una ubicación única para realizar un seguimiento de las tareas de migración en varias AWS herramientas y soluciones de socios.

AWS Migration Hub Orchestrator simplifica y automatiza la migración de servidores y aplicaciones empresariales a. AWS Migration Hub Orchestrator utiliza un secreto para la información de conexión al servidor de origen. Para obtener más información, consulte lo siguiente en la Guía del usuario del Orquestador de AWS Migration Hub :

- Migre las aplicaciones de SAP a NetWeaver AWS
- Rehospeda aplicaciones en Amazon EC2

Migration Hub Strategy Recommendations ofrece recomendaciones de estrategias de migración y modernización para rutas de transformación viables para sus aplicaciones. Strategy Recommendations puede analizar las bases de datos de SQL Server utilizando un secreto para la información de conexión. Para obtener más información, consulte Análisis de bases de datos de Strategy Recommendations.

Cómo AWS Panorama usa Secrets Manager

AWS Panorama es un servicio que lleva la visión artificial a la red de cámaras local. Se utiliza AWS Panorama para registrar un dispositivo, actualizar su software e implementar aplicaciones en él. Cuando registras una transmisión de vídeo como fuente de datos para tu aplicación, si la transmisión está protegida con contraseña, AWS Panorama guarda sus credenciales en un secreto de Secrets Manager. Para obtener más información, consulte Administración de transmisiones de cámara en AWS Panorama en la Guía para desarrolladores de AWS Panorama .

Cómo utiliza AWS Parallel Computing Service AWS Secrets Manager

AWS El servicio de computación paralela (AWS PCS) es un servicio gestionado que facilita la ejecución y el escalado de cargas de trabajo de computación de alto rendimiento (HPC) y aprendizaje automático distribuido. AWS

Para conectarse al programador de tareas del clúster, AWS PCS crea un secreto administrado con el prefijo `pcs` para almacenar la clave del programador. El costo de almacenar el secreto está incluido en el precio del PCS. AWS AWS PCS elimina automáticamente el secreto cuando usted elimina su clúster de AWS PCS. Para obtener más información, consulte Cómo trabajar con los secretos de los clústeres en AWS PCS en la Guía del usuario de AWS PCS.

> **Important**
>
> No modifique ni elimine los secretos del clúster de AWS PCS.

¿Cómo se AWS ParallelCluster usa AWS Secrets Manager

AWS ParallelCluster es una herramienta de administración de clústeres de código abierto que puede utilizar para implementar y administrar clústeres de computación de alto rendimiento (HPC) en. Nube

de AWS Puede crear un entorno de varios usuarios que incluya uno AWS ParallelCluster que esté integrado con un Microsoft AD (Active Directory) AWS administrado. AWS ParallelCluster Utiliza un secreto de Secrets Manager para validar los inicios de sesión en Active Directory. Para obtener más información, consulte Integrating Active Directory en la Guía del usuario de AWS ParallelCluster .

Cómo Amazon Q utiliza Secrets Manager

Para autenticar Amazon Q para acceder a su origen de datos, debe proporcionar sus credenciales de acceso al origen de datos a Amazon Q mediante un secreto de Secrets Manager. Si utiliza la consola, puede optar por crear un nuevo secreto o usar uno existente. Para obtener más información, consulte Conceptos: autenticación en la Guía para desarrolladores de Amazon Q.

¿Cómo se AWS OpsWorks for Chef Automate usa AWS Secrets Manager

AWS OpsWorks es un servicio de administración de la configuración que le ayuda a configurar y operar aplicaciones en una empresa en la nube mediante OpsWorks Puppet Enterprise o AWS OpsWorks for Chef Automate.

Al crear un nuevo servidor en AWS OpsWorks CM, OpsWorks CM almacena la información del servidor en un secreto gestionado por Secrets Manager con el prefijo`opsworks-cm`. El coste del secreto está incluido en el precio. AWS OpsWorks Para obtener más información, consulte Integración con AWS Secrets Manager en la Guía del usuario de AWS OpsWorks .

Cómo QuickSight usa Amazon AWS Secrets Manager

Amazon QuickSight es un servicio de inteligencia empresarial (BI) a escala de nube que puede utilizar para análisis, visualización de datos e informes. Puedes usar diversas fuentes de datos en Amazon QuickSight. Si guardas las credenciales de la base de datos en Secrets Manager Secrets, Amazon QuickSight puede usar esos secretos para conectarse a las bases de datos. Para obtener más información, 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.

Cómo utiliza Amazon RDS AWS Secrets Manager

Amazon Relational Database Service (Amazon RDS) es un servicio web que facilita la configuración, la operación y la escala de una base de datos relacional en Nube de AWS.

Para administrar las credenciales de usuario maestras de Amazon Relational Database Service (Amazon RDS), incluyendo Aurora, Amazon RDS puede 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.

Para otras credenciales de Amazon RDS, consulte Crear secretos.

Cuando utiliza Amazon RDS Query Editor para conectarse a una base de datos, puede almacenar las credenciales de la base de datos en Secrets Manager. Para obtener más información, consulte Uso del editor de consultas en la Guía del usuario de Amazon RDS.

Cómo utiliza Amazon Redshift AWS Secrets Manager

Amazon Redshift es un servicio de almacenamiento de datos administrado a escala de petabytes en la nube .

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 Crear secretos.

Al llamar a la API de datos de Amazon Redshift, puede transferir las credenciales del clúster mediante un secreto en Secrets Manager. Para obtener más información, consulte Uso de la API de datos de Amazon Redshift.

Cuando utiliza el Amazon Redshift Query Editor para conectarse a una base de datos, Amazon Redshift puede almacenar sus credenciales en un secreto de Secrets Manager con el prefijo `redshiftqueryeditor`. Se le cobrará ese secreto. Para obtener más información, consulte Consulta de una base de datos mediante el editor de consultas en la Guía de administración de Amazon Redshift.

Para Query Editor v2, consulte the section called “Editor de consultas V2 de Amazon Redshift”.

Amazon Redshift Query Editor v2

Amazon Redshift Query Editor v2 es una aplicación de cliente SQL basada en la Web que puede utilizar para crear y ejecutar consultas en su almacenamiento de datos de Amazon Redshift. Cuando se utiliza el Amazon Redshift Query Editor 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 Query Editor v2 en la Guía de administración de Amazon Redshift.

Para ver el editor de consultas anterior, consulte the section called “Amazon Redshift”.

Cómo usa Amazon SageMaker AI AWS Secrets Manager

SageMaker La IA es un servicio de aprendizaje automático totalmente gestionado. Con la SageMaker IA, los científicos de datos y los desarrolladores pueden crear y entrenar modelos de aprendizaje automático de forma rápida y sencilla y, a continuación, implementarlos directamente en un entorno hospedado listo para la producción. Incluye una instancia de bloc de notas de creación de Jupyter integrada para obtener acceso de manera sencilla a sus orígenes de datos y poder realizar estudios y análisis sin tener que administrar servidores.

Puede asociar repositorios de Git a sus instancias de cuaderno Jupyter para guardar sus cuadernos en un entorno de control de origen que persiste incluso si se detiene o se elimina su instancia de cuaderno. Puede administrar sus credenciales de repositorio privado con Secrets Manager. Para obtener más información, consulte Asociar repositorios de Git con instancias de Amazon SageMaker Notebook en la Guía para desarrolladores de Amazon SageMaker AI.

Para importar datos de Databricks, Data Wrangler almacena la URL de JDBC en Secrets Manager. Para obtener más información, consulte Importación de datos desde Databricks (JDBC).

Para importar datos de Snowflake, Data Wrangler almacena las credenciales en un secreto de Secrets Manager. Para obtener más información, consulte Importación de datos desde Snowflake.

¿Cómo se usa AWS Schema Conversion ToolAWS Secrets Manager

Puede usar AWS Schema Conversion Tool (AWS SCT) para convertir el esquema de base de datos existente de un motor de base de datos a otro. Puede convertir esquemas relacionales OLTP o esquemas de data warehouse. El esquema convertido es adecuado para una base de datos MySQL de Amazon Relational Database Service (Amazon RDS), MariaDB, Oracle, SQL Server o PostgreSQL, un clúster de base de datos de Amazon Aurora o un clúster de Amazon Redshift. El esquema convertido también puede utilizarse con una base de datos en una instancia de Amazon Elastic Compute Cloud o almacenarse como datos en un bucket de S3.

Al convertir un esquema de base de datos, AWS SCT puede usar las credenciales de base de datos que almacene AWS Secrets Manager. Para obtener más información, consulte Utilización AWS Secrets Manager en la interfaz AWS SCT de usuario de la Guía del AWS Schema Conversion Tool usuario.

Cómo utiliza Amazon Timestream para InfluxDB AWS Secrets Manager

Timestream for InfluxDB es un motor de base de datos de series temporales gestionado que facilita la ejecución de bases de datos de InfluxDB para aplicaciones de series temporales en tiempo real mediante código abierto. AWS APIs Con Timestream para InfluxDB, puede configurar, manejar y escalar cargas de trabajo de serie temporal que pueden responder consultas con un tiempo de respuesta de consultas de milisegundos de un solo dígito.

Al crear una base de datos de Timestream para InfluxDB, Timestream crea automáticamente un secreto para almacenar las credenciales de administrador. Para obtener más información, consulte How Amazon Timestream for InfluxDB uses secrets en la Guía para desarrolladores de Timestream.

¿Cómo se usa AWS Toolkit for JetBrainsAWS Secrets Manager

AWS Toolkit for JetBrains Es un complemento de código abierto para los entornos de desarrollo integrados (IDEs) de JetBrains. Este kit de herramientas facilita a los desarrolladores el desarrollo, la depuración y la implementación de aplicaciones sin servidor que utilicen AWS. Al conectarse a un clúster de Amazon Redshift mediante el kit de herramientas, puede autenticarse mediante un secreto de Secrets Manager. Para obtener más información, consulte Accessing Amazon Redshift clusters (Acceso a clústeres de Amazon Redshift) en la Guía del usuario de AWS Toolkit for JetBrains .

¿Cómo AWS Transfer Family usa AWS Secrets Manager los secretos

AWS Transfer Family es un servicio de transferencia segura que permite transferir archivos hacia y desde los servicios de AWS almacenamiento.

Transfer Family ahora admite el uso de la autenticación básica para los servidores que utilizan el protocolo Applicability Statement 2 (AS2). Puede crear un nuevo secreto de Secrets Manager o elegir un secreto existente para sus credenciales. Para obtener más información, consulte Autenticación básica para AS2 conectores en la Guía del AWS Transfer Family usuario.

Para autenticar a los usuarios de Transfer Family, puedes utilizarla AWS Secrets Manager como proveedor de identidad. Para obtener más información, consulte Cómo trabajar con proveedores de identidad personalizados en la Guía del AWS Transfer Family usuario y en el artículo del blog sobre cómo habilitar la autenticación mediante contraseña para su AWS Transfer Family uso AWS Secrets Manager.

Puede utilizar el descifrado de Pretty Good Privacy (PGP) con los archivos que Transfer Family procesa mediante flujos de trabajo. Para utilizar el descifrado en un paso del flujo de trabajo, debe proporcionar una clave PGP que administre en Secrets Manager. Para obtener más información, consulte Generate and manage PGP keys (Generar y adminsitrar claves PGP) en la Guía del usuario de AWS Transfer Family .

¿Cómo AWS Wickr usa AWS Secrets Manager los secretos

AWS Wickr es un servicio end-to-end cifrado que ayuda a las organizaciones y agencias gubernamentales a comunicarse de forma segura a través one-to-one de la mensajería grupal, las llamadas de voz y vídeo, el uso compartido de archivos, el uso compartido de pantallas y mucho más. Puede automatizar los flujos de trabajo con los bots de retención de datos de Wickr. Si el bot va a tener acceso a Servicios de AWS, debes crear un secreto de Secrets Manager para almacenar las credenciales del bot. Para obtener más información, consulte Iniciar el bot de retención de datos en la Guía de AWS Wickr administración.

Crea AWS Secrets Manager secretos en AWS CloudFormation

Puede crear secretos en una CloudFormation pila utilizando el `AWS::SecretsManager::Secret` recurso de una CloudFormation plantilla, como se muestra enCreación 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 Aprenda los conceptos básicos de las plantillas en la Guía del AWS CloudFormation usuario. También puede utilizar la AWS Cloud Development Kit (AWS CDK). Para obtener más información, consulte Biblioteca de construcción AWS Secrets Manager.

Crea un AWS Secrets Manager secreto 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 CloudFormation recurso:

- AWS::SecretsManager::Secret

Para obtener información sobre cómo crear recursos con AWS CloudFormation, consulte Aprenda los conceptos básicos de las plantillas en la Guía del AWS CloudFormation usuario.

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
```

Cree un AWS Secrets Manager secreto 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 de `AWS::Redshift::Cluster` y `AWS::RedshiftServerless::Namespace`.

Cree un AWS Secrets Manager secreto 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 CloudFormation recursos para Secrets Manager:

- `AWS::SecretsManager::Secret`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecretsManager::RotationSchedule`

Para obtener información sobre cómo crear recursos con AWS CloudFormation, consulte Aprenda los conceptos básicos de las plantillas en la Guía del AWS CloudFormation usuario.

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: !Select
        - '0'
        - !GetAZs
          Ref: AWS::Region
```

```
      VpcId: !Ref TestVPC
  TestSubnet02:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.128.0/19
      AvailabilityZone: !Select
        - '1'
        - !GetAZs
          Ref: AWS::Region
      VpcId: !Ref TestVPC
  SecretsManagerVPCEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      SubnetIds:
        - !Ref TestSubnet01
        - !Ref TestSubnet02
      SecurityGroupIds:
        - !GetAtt TestVPC.DefaultSecurityGroup
      VpcEndpointType: Interface
      ServiceName: !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: !Sub '{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::username}}'
      MasterUserPassword: !Sub '{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::password}}'
      VpcSecurityGroupIds:
        - !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: !GetAtt TestVPC.DefaultSecurityGroup
        VpcSubnetIds: !Join
          - ','
          - - !Ref TestSubnet01
            - !Ref TestSubnet02
      RotationRules:
        Duration: 2h
        ScheduleExpression: cron(0 8 1 * ? *)
```

Cómo usa Secrets Manager AWS CloudFormation

Cuando utilizas la consola para activar la rotación, Secrets Manager la utiliza AWS CloudFormation para crear recursos para la rotación. Si crea una nueva función de rotación durante ese proceso, AWS CloudFormation crea una función `AWS::Serverless::Function`basada en la adecuadaPlantillas de función de rotación. A continuación, AWS CloudFormation establece el `RotationSchedule`, que establece la función de rotación y las reglas de rotación del secreto. Para

ver la AWS CloudFormation pila, selecciona 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 .

Crea AWS Secrets Manager secretos 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`.

Una práctica recomendada para usar secretos en las aplicaciones de CDK es primero crear el secreto con la consola o la CLI y, a continuación, importarlo a la aplicación de CDK.

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.

Supervise AWS Secrets Manager los secretos

AWS proporciona herramientas de supervisión para ver los secretos de Secrets Manager, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario. 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

AWS Secrets Manager Registra eventos con AWS CloudTrail

AWS CloudTrail registra todas las llamadas a la API de Secrets Manager como eventos, incluidas las llamadas desde la consola de Secrets Manager, así como varios otros eventos para la rotación y la eliminación de versiones secretas. Para obtener una lista de las entradas de registro de los registros en Secrets Manager, consulte CloudTrail entradas.

Puede usar la CloudTrail consola para ver los eventos registrados en los últimos 90 días. 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 Crear un registro para su AWS cuenta. 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:
    a. Para ver todos los eventos de Secrets Manager, para Atributos de búsqueda, elija Origen del evento. A continuación, para Enter event source (Escribir origen del evento), elija **`secretsmanager.amazonaws.com`**.
    b. Para ver todos los eventos de un secreto, en Atributos de búsqueda, elija Nombre del recurso. A continuación, en Introducir un nombre de recurso, introduzca el nombre del secreto.
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
```

AWS CloudTrail entradas para Secrets Manager

AWS Secrets Manager escribe entradas en su AWS CloudTrail registro 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 Obtener 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 Administrar secretos.

DeleteResourcePolicy

Generado por la DeleteResourcePolicyoperación. Para obtener información acerca de los permisos, consulte the section called “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 the section called “Autenticación y control de acceso”.

GetSecretValue

Generado por las BatchGetSecretValueoperaciones GetSecretValuey. Para obtener información sobre cómo recuperar secretos, consulte Obtener 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 the section called “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 Replicación multirregional.

ReplicateSecretToRegions

Generado por la ReplicateSecretToRegionsoperación. Para obtener información acerca de un secreto, consulte Replicación multirregional.

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 Replicación multirregional.

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 “Versiones de un secreto”.

ValidateResourcePolicy

Generado por la ValidateResourcePolicyoperación. Para obtener información acerca de los permisos, consulte the section called “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 Replicación multirregional.

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 Replicación multirregional.

ReplicationSucceeded

Generado por el servicio de Secrets Manager cuando un secreto se replica correctamente. Para obtener información acerca de un secreto, consulte Replicación multirregional.

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”.

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”.

Monitoriza AWS Secrets Manager con Amazon CloudWatch

Con Amazon CloudWatch, puedes monitorear AWS los servicios y crear alarmas que te avisen cuando cambien las métricas. CloudWatch guarda estas estadísticas durante 15 meses, para que puedas acceder a la información histórica y obtener una mejor perspectiva del rendimiento de tu aplicación o servicio web. Pues AWS Secrets Manager, puedes controlar la cantidad de secretos de tu cuenta, incluidos los secretos marcados para su eliminación, y las llamadas a la API a Secrets Manager, incluidas las llamadas realizadas a través de la consola. Para obtener información sobre cómo supervisar las métricas, consulte Uso de CloudWatch métricas en la Guía del CloudWatch usuario.

Buscar métricas de Secrets Manager

1. En la CloudWatch consola, en Métricas, selecciona Todas las métricas.
2. En el cuadro de búsqueda de Métricas, escriba `secret`.
3. Haga lo siguiente:
   - Para controlar la cantidad de datos secretos de tu cuenta, selecciona AWS/ySecretsManager, a continuación, selecciona SecretCount. Esta métrica se publica cada hora.

- Para supervisar las llamadas de API a Secrets Manager, incluidas las llamadas realizadas a través de la consola, selecciona Uso > Por AWS recurso y, a continuación, selecciona las llamadas a la API que deseas supervisar. Para obtener una lista de Secrets Manager APIs, consulte las operaciones de Secrets Manager.

4. Haga lo siguiente:
   - Para crear un gráfico de la métrica, consulta Cómo graficar métricas en la Guía del CloudWatch usuario de Amazon.
   - Para detectar anomalías, consulte Uso de la detección de CloudWatch anomalías en la Guía del usuario de Amazon CloudWatch .
   - Para obtener estadísticas de una métrica, consulta Obtener estadísticas de una métrica en la Guía del CloudWatch usuario de Amazon.

CloudWatch alarmas

Puede crear una CloudWatch alarma que envíe un mensaje de Amazon SNS cuando el valor de una métrica cambie y haga que la alarma cambie de estado. Puede configurar una alarma en la métrica `ResourceCount` de Secrets Manager, que es el número de secretos de su cuenta. También puede configurar alarmas que vigilen una métrica durante el periodo especificado y realicen 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 encuentren en un estado determinado; el estado debe haber cambiado y 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 en la Guía del CloudWatch usuario.

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.

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

> **Note**
>
> Dado que algunos eventos de Secrets Manager devuelven el ARN del secreto con mayúsculas diferentes, en los patrones de eventos que coinciden con más de una acción, para especificar un secreto a través de ARN, es posible que tenga que incluir tanto las claves `arn` como `aRN`. Para obtener más información, consulte AWS re:Post.

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"]
    }
}
```

Supervise cuándo se accede a los AWS Secrets Manager secretos cuya eliminación está programada

Puedes usar una combinación de AWS CloudTrail Amazon CloudWatch Logs y Amazon Simple Notification Service (Amazon SNS) para crear una alarma que te notifique cualquier intento de acceso a un secreto 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 recibir una notificación cuando se solicita una `GetSecretValue` operación que dé lugar a un mensaje de error específico en sus archivos de CloudTrail registro. Se pueden realizar otras operaciones de API en el secreto sin activar la alarma. Esta CloudWatch alarma detecta un uso que podría indicar que una persona o aplicación utiliza credenciales desactualizadas.

Antes de iniciar estos procedimientos, debes activar la cuenta Región de AWS y CloudTrail en la que pretendes supervisar las solicitudes de AWS Secrets Manager API. 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 la entrega de archivos de CloudTrail registro a CloudWatch Logs

Debe configurar la entrega de sus archivos de CloudTrail registro a CloudWatch Logs. Esto se hace para que CloudWatch Logs pueda supervisarlos en busca de solicitudes de la API Secrets Manager para recuperar un secreto pendiente de ser eliminado.

Para configurar la entrega de archivos de CloudTrail registro a CloudWatch Logs

1. Abra la CloudTrail consola en https://console.aws.amazon.com/cloudtrail/.
2. En la barra de navegación superior, selecciona la opción Región de AWS para monitorizar los secretos.
3. En el panel de navegación izquierdo, selecciona Rutas y, a continuación, elige el nombre de la ruta que deseas configurar CloudWatch.
4. En la página de configuración de senderos, desplácese hacia abajo hasta la sección CloudWatch Registros 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. Para el rol de IAM, puede usar el rol predeterminado denominado CloudTrail_ CloudWatchLogs _Role. Este rol tiene una política de roles predeterminada con los permisos necesarios para entregar CloudTrail eventos al grupo de registros.
7. Elija Continue (Continuar) para guardar la configuración.
8. En la AWS CloudTrail página para enviar CloudTrail los eventos asociados a la actividad de la API de tu cuenta a tu grupo de CloudWatch registros, selecciona Permitir.

Paso 2: Crea la CloudWatch alarma

Para recibir una notificación cuando una operación de la `GetSecretValue` API Secrets Manager solicite acceder a un secreto pendiente de eliminación, debe crear una CloudWatch alarma y configurar la notificación.

Para crear una CloudWatch alarma

1. Inicie sesión en la CloudWatch consola en https://console.aws.amazon.com/cloudwatch/.
2. En la barra de navegación superior, elige la AWS región en la que quieres supervisar los secretos.
3. En el panel de navegación izquierdo, elija Logs (Registros).
4. En la lista de grupos de registros, active la casilla de verificación situada junto al grupo de registros que creó en el procedimiento anterior, como 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. Seleccione Crear alarma.

Paso 3: Pruebe la CloudWatch alarma

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.

Supervise AWS Secrets Manager los secretos para garantizar el cumplimiento mediante AWS Config

Puede AWS Config utilizarla para evaluar sus secretos y comprobar si cumplen con sus normas. Los requisitos internos de seguridad y cumplimiento de los secretos se definen mediante AWS Config reglas. A continuación, AWS Config podrá identificar los secretos que no se ajusten a sus 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 configurarlo AWS Config para que le notifique los cambios. Para obtener más información, consulte el tema Notificaciones que se AWS Config envían a un sitio de Amazon SNS.

Si tiene secretos en varios sitios Cuentas de AWS y Regiones de AWS en su organización, puede agregar esos datos de configuración y conformidad. Para obtener más información, consulte Acumulación de datos de varias cuentas y regiones.

Evaluar la conformidad de los secretos

- Siga las instrucciones de la sección Evaluación de los recursos con AWS Config reglas y elija una de las siguientes reglas:
  - `secretsmanager-secret-unused`: verifica si se accedió a los secretos dentro de la cantidad de días especificada.
  - `secretsmanager-using-cmk`— Comprueba si los secretos se cifran con la clave gestionada por el cliente Clave administrada de AWS `aws/secretsmanager` o con una clave que hayas creado AWS KMS.
  - `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.

Monitoreo de los costos de Secrets Manager

Puedes usar Amazon CloudWatch para controlar los AWS Secrets Manager cargos estimados. Para obtener más información, consulta Cómo crear una alarma de facturación para controlar AWS los cargos estimados en la Guía del CloudWatch usuario.

Otra opción para monitorear sus costos es la detección de anomalías en los AWS costos. Para obtener más información, consulte Detección de gastos inusuales mediante la detección de anomalías en los costos de AWS en la Guía del usuario de gestión de costos de AWS .

Para obtener información sobre cómo supervisar el uso de Secrets Manager, consulte the section called “Supervise con CloudWatch” y the section called “Inicia sesión con AWS CloudTrail ”.

Para obtener información sobre AWS Secrets Manager los precios, consultethe section called “Precios”.

Detecta amenazas con Amazon GuardDuty

Amazon GuardDuty es un servicio de detección de amenazas que le ayuda a proteger sus cuentas, contenedores, cargas de trabajo y los datos de su AWS entorno. Mediante el uso de modelos de aprendizaje automático (ML) y funciones de detección de anomalías y amenazas, supervisa GuardDuty continuamente las diferentes fuentes de registro para identificar y priorizar los posibles riesgos de seguridad y las actividades maliciosas en su entorno. Por ejemplo, GuardDuty detectará posibles amenazas, como el acceso inusual o sospechoso a secretos y la exfiltración de credenciales en caso de que detecte credenciales que se crearon exclusivamente para una EC2 instancia de Amazon a través de una función de lanzamiento de instancias, pero que se utilizan desde otra cuenta interna. AWS Para obtener más información, consulta la Guía del GuardDuty usuario de Amazon.

Otro ejemplo de caso de uso para la detección es el comportamiento anómalo. Por ejemplo, si AWS Secrets Manager normalmente recibe`create-secret`, `get-secret-value``describe-secret`, y `list-secrets` llamadas de una entidad que utiliza el SDK de Java y, a continuación, otra entidad empieza a llamar `batch-get-secret-value` y a `get-secret-value` utilizar las llamadas AWS CLI desde fuera de la VPN, GuardDuty puede indicar que la segunda entidad está APIs invocando de forma anómala. Para obtener más información, consulte el tipo CredentialAccess de búsqueda de GuardDuty IAM:/. IAMUser AnomalousBehavior

Validación de conformidad para AWS Secrets Manager

Su responsabilidad de cumplimiento al utilizar Secrets Manager viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Security and Compliance Quick Start Guides (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan las consideraciones en materia de arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS Recursos de cumplimiento Recursos de de trabajo y guías puede 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 “Supervisión de secretos para la conformidad”.
- AWS Security Hubproporciona una visión completa del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la 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.
- Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Estándares de conformidad

AWS Secrets Manager se ha sometido a una auditoría para cumplir con los siguientes estándares y puede ser parte de su solución cuando necesite obtener una certificación de conformidad.

- HIPAA: AWS ha ampliado su programa de cumplimiento de la Ley de Portabilidad y Responsabilidad de los Seguros de Salud (HIPAA) para incluirlo AWS Secrets Manager como un servicio elegible para la HIPAA. Si ha firmado un acuerdo de asociación comercial (BAA) con usted AWS, puede usar Secrets Manager para ayudarlo a crear sus aplicaciones que cumplan con la HIPAA. AWS ofrece un documento técnico centrado en la HIPAA para los clientes que estén interesados en obtener más información sobre cómo pueden aprovechar AWS el procesamiento y el almacenamiento de la información de salud. Para obtener más información, consulte Conformidad con HIPAA.
- Organización participante en la PCI: AWS Secrets Manager cuenta con un certificado de conformidad con la versión 3.2 de la norma de seguridad de datos (DSS) de la industria de tarjetas de pago (PCI), de nivel 1. Los clientes que utilizan AWS productos y servicios para almacenar, procesar o transmitir datos de titulares de tarjetas pueden utilizarlos para gestionar su propia certificación de conformidad con la AWS Secrets Manager PCI DSS. Para obtener más información sobre PCI DSS, incluida la forma de solicitar una copia del PCI AWS Compliance Package, consulte PCI DSS Level 1.
- ISO: AWS Secrets Manager ha obtenido satisfactoriamente las certificaciones de conformidad de las normas 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.
- AICPA SOC: 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. El objetivo de estos informes es ayudarlo a usted y a sus auditores a comprender los AWS controles que se establecen para respaldar las operaciones y el cumplimiento. Para obtener más información, consulte Conformidad con SOC.
- FedRAMP: El Programa Federal de Gestión de Riesgos y Autorizaciones (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 del Este y el Oeste GovCloud y para consumir datos gubernamentales o regulados. Para obtener más información, consulte Conformidad con FedRAMP.
- Departamento de Defensa: la Guía de requisitos de seguridad de la computación en la nube (SRG) del Departamento de Defensa (DoD) proporciona un proceso estandarizado de evaluación y

autorización para que los proveedores de servicios en la nube (CSPs) obtengan una autorización provisional del DoD, de modo que puedan atender a los clientes del DoD. Para obtener más información, consulte Recursos de DoD SRG

- IRAP: El Programa de Asesores Registrados de Seguridad de la Información (IRAP) 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) del gobierno australiano producido por el Centro Australiano de Ciberseguridad (ACSC). Para obtener más información, consulte Recursos de IRAP.
- OSPAR — Amazon Web Services (AWS) obtuvo la certificación del informe de auditoría del proveedor de servicios subcontratado (OSPAR). AWS La conformidad con las Directrices de la Asociación de Bancos de Singapur (ABS) sobre los objetivos y procedimientos de control para los proveedores de servicios subcontratados (Directrices ABS) demuestra a AWS los clientes el compromiso de cumplir con las altas expectativas de los proveedores de servicios en la nube establecidas por la industria de servicios financieros en Singapur. Para obtener más información, consulte Recursos OSPAR.

Seguridad en AWS Secrets Manager

La seguridad AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

Usted y yo AWS compartimos 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 AWS los servicios en la AWS nube. AWS también le 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 cumplimiento aplicables AWS Secrets Manager, consulte los AWS servicios incluidos en el ámbito de aplicación por programa de cumplimiento.
- Seguridad en la nube: su AWS servicio determina su responsabilidad. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Para obtener más recursos, consulte Security Pillar — AWS Well-Architected Framework.

Temas

Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos AWS Secrets Manager

Cuando usa AWS Command Line Interface (AWS CLI) para invocar AWS operaciones, ingresa esos comandos en una consola de comandos. Por ejemplo, puede usar la línea de comandos de Windows o Windows PowerShell, o los shell Bash o Z, 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. El siguiente ejemplo muestra cómo escribir el texto secreto en un archivo de texto y, a continuación, pasar el archivo al AWS Secrets Manager comando y destruirlo 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 de la esquina superior izquierda y, a continuación, elija 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 de la línea de comandos de Windows, puede descargar la SysInternals SDeleteherramienta y, a continuación, 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.
```

Autenticación y control de acceso para 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 Iniciar 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.

Temas

- Referencia de permisos para AWS Secrets Manager
- Permisos de Secrets Manager
- Permisos para acceder a secretos
- Permisos para las funciones de rotación de Lambda
- Permisos para claves de cifrado
- Permisos de replicación
- Políticas basadas en identidad
- Políticas basadas en recursos
- Controlar el acceso a los secretos mediante el control de acceso basado en atributos (ABAC)
- AWS política gestionada para AWS Secrets Manager
- Determina quién tiene permisos para acceder a tus AWS Secrets Manager secretos
- Accede a AWS Secrets Manager los secretos desde una cuenta diferente
- Acceso a los secretos desde un entorno en las instalaciones

Referencia de permisos para AWS Secrets Manager

La referencia de permisos para Secrets Manager está disponible en Acciones, recursos y claves de condición para AWS Secrets Manager en la Referencia de autorizaciones de servicio.

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. Si bien esto permite a los usuarios crear y administrar sus secretos, el permiso necesario para habilitar la rotación (IAMFullAccess) concede permisos importantes 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 “Políticas basadas en identidad”
- the section called “Políticas basadas en recursos”

Permisos para las funciones de rotación de Lambda

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

Permisos de replicación

Mediante las políticas de permisos de IAM, puede controlar qué usuarios o servicios pueden replicar sus secretos a otras regiones. Consulte the section called “Impedir la replicación”.

Políticas basadas en 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 EC2 instancia de Amazon puede necesitar acceso a una base de datos. Puedes crear un rol de IAM adjunto al perfil de la EC2 instancia y, a continuación, usar una política de permisos para conceder al rol acceso al secreto que contiene las credenciales de la base de datos. Para obtener más información, consulta Cómo usar un rol de IAM para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon. 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.

Ejemplos:

- Ejemplo: permiso para recuperar valores secretos
- Ejemplo: Permiso para leer y describir secretos individuales

- Ejemplo: permiso para recuperar un grupo de valores secretos en un lote
- Ejemplo: comodines
- Ejemplo: permiso para crear secretos
- Ejemplo: denegar una AWS KMS clave específica para cifrar los secretos

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 “Políticas basadas en recursos” y the section called “Políticas basadas en identidad”.

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 “Ejemplo: permiso para recuperar un grupo de valores secretos en un lote”.

Example Leer un secreto cifrado mediante una clave administrada por el cliente

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. \

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

Ejemplo: Permiso para leer y describir secretos individuales

Example Leer y describir un secreto

Puede conceder acceso a un secreto adjuntando la siguiente política una identidad.

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

Ejemplo: permiso para recuperar un grupo de valores secretos en un lote

Example Leer un grupo de secretos en un lote

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 SecretARN2SecretARN3*, e 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. `BatchGetSecretValue` .

```
{
"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

La siguiente política permite recuperar todos los secretos cuyo nombre comience por "*TestEnv/*».

```
{
"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

Las siguientes políticas conceden `DescribeSecret` y permisos comenzando con `List`: `ListSecrets` y `ListSecretVersionIds`.

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

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

Example Coincidir el nombre del secreto

La siguiente política concede permisos de Secrets Manager para un secreto por su nombre. Para utilizar esta política, visite the section called “Políticas basadas en 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

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

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

Ejemplo: denegar una AWS KMS clave específica para cifrar los secretos

> **Important**
>
> Para denegar una clave administrada por el cliente, le recomendamos que restrinja el acceso mediante una política de claves o una concesión de claves. Para obtener más información, consulte Autenticación y control de acceso para AWS KMS en la Guía del desarrollador de AWS Key Management Service .

Example Denegar la clave AWS gestionada **`aws/secretsmanager`**

La siguiente política deniega el uso de la Clave administrada de AWS `aws/secretsmanager` para crear o actualizar secretos. Esta política exige que los secretos se cifren mediante una clave administrada por el cliente. La política incluye dos instrucciones:

1. La primera declaración,`Sid: "RequireCustomerManagedKeysOnSecrets"`, deniega las solicitudes de creación o actualización de secretos mediante el Clave administrada de AWS `aws/secretsmanager`.
2. La segunda sentencia,`Sid: "RequireKmsKeyIdParameterOnCreate"`, deniega las solicitudes de creación de secretos que no incluyan una clave de KMS, ya que Secrets Manager utilizaría de forma predeterminada la Clave administrada de AWS `aws/secretsmanager`.

```
{
"Version": "2012-10-17",
"Statement": [
  {
      "Sid": "RequireCustomerManagedKeysOnSecrets",
      "Effect": "Deny",
      "Action": [
          "secretsmanager:CreateSecret",
          "secretsmanager:UpdateSecret"
      ],
      "Resource": "*",
      "Condition": {
          "StringLikeIfExists": {
              "secretsmanager:KmsKeyArn": "<key_ARN_of_the_AWS_managed_key>"
          }
      }
  },
  {
      "Sid": "RequireKmsKeyIdParameterOnCreate",
      "Effect": "Deny",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "*",
      "Condition": {
          "Null": {
              "secretsmanager:KmsKeyArn": "true"
          }
      }
  }
]
}
```

Políticas basadas en recursos

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.

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 parámetro `BlockPublicPolicy` ayudan a proteger sus recursos al impedir que se conceda acceso público a través de las políticas de recursos que se adjuntan directamente a sus secretos. Además de usar estas características, analice detenidamente las siguientes políticas para confirmar que no otorgan 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 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.

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

Ejemplos

Ejemplos:

- Ejemplo: permiso para recuperar valores secretos
- Ejemplo: permisos y VPCs
- Ejemplo: Entidad principal de servicio

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 “Políticas basadas en recursos” y the section called “Políticas basadas en identidad”.

Este ejemplo es útil cuando desea conceder acceso a un secreto único a varios usuarios o roles. Para conceder permiso para recuperar un grupo de secretos en una llamada a la API por lotes,

consulte the section called “Ejemplo: permiso para recuperar un grupo de valores secretos en un lote”.

Example Leer un secreto

Puede conceder acceso a un secreto adjuntando la siguiente política al secreto.

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

Ejemplo: permisos y VPCs

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 Limitar solicitudes con condiciones del punto de conexión de VPC y the section called “Puntos de conexión de VPC (AWS PrivateLink)”.

Asegúrese de que las solicitudes de acceso al secreto desde otros AWS servicios también provengan de la VPC; de lo contrario, esta política les negará el acceso.

Example Requerir que las solicitudes lleguen a través de un punto de conexión de VPC

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*.

```
{
"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

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.

```
{
"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: Entidad principal de servicio

Si la política de recursos adjunta a su secreto incluye un director de AWS servicio, le recomendamos que utilice las claves de condición SourceAccount globales aws: SourceArn y aws:. 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.

Los directores de servicio no suelen utilizarse como principales en una política asociada a un secreto, pero algunos AWS servicios sí lo exigen. 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

```
{
```

```
"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"
    }
  }

}
]
}
```

Controlar el acceso a los secretos mediante el control de acceso basado en atributos (ABAC)

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos basados en atributos o características del usuario, los datos o el entorno, como el departamento, la unidad de negocio u otros factores que podrían afectar al resultado de la autorización. En AWS, estos atributos se denominan 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. Las reglas del ABAC se evalúan de forma dinámica durante el tiempo de ejecución, lo que significa que el acceso de los usuarios a las aplicaciones y los datos y el tipo de operaciones permitidas cambian automáticamente en función de los factores contextuales de la política. Por ejemplo, si un usuario cambia de departamento, el acceso se ajusta automáticamente sin necesidad de actualizar los permisos ni solicitar nuevos roles. Para obtener más información, consulte: ¿Para qué sirve ABAC? AWS , Defina los permisos para acceder a los secretos en función de las etiquetas. y amplíe sus necesidades de autorización para Secrets Manager mediante ABAC con IAM Identity Center.

Ejemplo: Permitir que una identidad acceda a secretos que tienen etiquetas específicas

La siguiente política permite el `DescribeSecret` acceso a los secretos mediante una etiqueta con la clave *`ServerName`* y el valor. *`ServerABC`* Si vincula esta política a una identidad, esta tendrá permiso para guardar cualquier secreto de la cuenta que tenga esa etiqueta.

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

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

La siguiente política permite que las identidades de la cuenta `GetSecretValue` accedan a todos los secretos de la cuenta en los que la etiqueta *`AccessProject`* de la identidad tenga el mismo valor que la etiqueta *`AccessProject`* del secreto.

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

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

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 de 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 Amazon EC2 VPCs. 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 funciones de AWS Lambda ejecución. Puede asignar la política de IAMFullacceso gestionado. 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 usa Secrets Manager AWS CloudFormation”.
- `ec2`— Permite a los directores describir Amazon EC2 VPCs. 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 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 las entidades principales describir los espacios de nombres de Amazon Redshift sin servidor. Esto es necesario para que las entidades principales que utilicen la consola puedan seleccionar espacios de nombres de Amazon Redshift sin servidor.
- `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 AWS CloudFormation conjuntos de cambios. 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 usa Secrets Manager 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 usa Secrets Manager 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	Versión
SecretsManagerReadWrite: actualización de una política actual	Esta política se actualizó para permitir que se describa el acceso a Amazon Redshift sin servidor de modo que los usuarios de la consola puedan seleccionar un espacio de nombres de Amazon Redshift sin servidor cuando crean un secreto de Amazon Redshift.	12 de marzo de 2024	v5
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	12 de septiembre de 2023	v4

Cambio	Descripción	Fecha	Versión
	puedan seleccionar un clúster elástico al crear un secreto de Amazon DocumentDB.		
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	v3

Cambio	Descripción	Fecha	Versión
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	v2
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	v1

Determina quién tiene permisos para acceder a tus AWS Secrets Manager secretos

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 una `allow` en la política de recursos para el secreto y la política asociada a la identidad; de lo contrario, AWS denegará 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, lo que se denomina denegación implícita.

Ver la política basada en recursos de un secreto

- Realice 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).
  - Usa el AWS CLI para llamar `get-resource-policy`o el AWS SDK para llamar `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.

Accede a AWS Secrets Manager los secretos desde 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 puedes usar el Clave administrada de AWS (`aws/secretsmanager`) para

acceder a varias 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 *ApplicationRole* acceder *Account2* a la entrada secreta. *Account1* Para utilizar esta política, visite the section called “Políticas basadas en recursos”.

```
{
  "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 de claves permite que *ApplicationRole* en *Account2* use la clave de KMS en *Account1* para descifrar el secreto en *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 *Account2* acceda al secreto de *Account1* y descifre el valor secreto mediante la clave de cifrado que también está en *Account1*. Para utilizar esta política, visite the section called “Políticas basadas en 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"
    }
  ]
}
```

Acceso a los secretos desde un entorno en las instalaciones

Puedes usar AWS Identity and Access Management Roles Anywhere para obtener credenciales de seguridad temporales en IAM para cargas de trabajo como servidores, contenedores y aplicaciones que se ejecutan fuera de ellas. AWS Sus cargas de trabajo pueden usar las mismas políticas de IAM y las mismas funciones de IAM que usa con AWS las aplicaciones para acceder a los recursos. AWS Con IAM Roles Anywhere, puede utilizar Secrets Manager para almacenar y gestionar las credenciales a las que pueden acceder los recursos y los dispositivos locales, AWS como los servidores de aplicaciones. Para obtener más información, consulte la Guía del usuario de IAM Roles Anywhere.

Protección de datos en AWS Secrets Manager

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en AWS Secrets Manager. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Es responsable de mantener el control sobre su contenido que se encuentra alojado en esta infraestructura. Este contenido incluye la configuración de seguridad y las tareas de administración para el Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de datos, consulte Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y GDPR en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure cuentas de usuario individuales con AWS Identity and Access Management (IAM). De esta manera, cada usuario recibe únicamente los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes maneras:

- Utilice autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. 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 de los usuarios con. AWS CloudTrail Consulte the section called “Inicia sesión con AWS CloudTrail ”.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Consulte the section called “Puntos de conexión de Secrets Manager”.
- Si usa el AWS CLI para acceder a Secrets Manager,the section called “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos AWS Secrets Manager”.

Cifrado en reposo

Secrets Manager utiliza el cifrado mediante AWS Key Management Service (AWS KMS) para proteger la confidencialidad de los datos en reposo. AWS KMS proporciona un servicio de almacenamiento y cifrado de claves utilizado por muchos AWS servicios. 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 finales seguros y privados permiten AWS proteger la integridad de las solicitudes de API a Secrets Manager. AWS requiere que las llamadas a la API estén firmadas por la persona que llama mediante certificados X.509 y/o una clave de acceso secreta de Secrets Manager. Este requisito se indica en Proceso de firma de Signature Versión 4 (Sigv4).

Si utilizas el AWS Command Line Interface (AWS CLI) o alguno de ellos AWS SDKs para realizar llamadas AWS, configuras la clave de acceso que vas a utilizar. A continuación, esas herramientas utilizan automáticamente la clave de acceso para firmar las solicitudes por usted. Consulte the section called “Mitigue los riesgos de utilizarlos AWS CLI para almacenar sus secretos 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

Dispone de dos opciones de conectividad entre su red privada y AWS Secrets Manager:

- Una conexión AWS Site-to-Site VPN. Para obtener más información, consulta ¿Qué es una VPN AWS Site-to-Site?
- Una conexión AWS Direct Connect. Para obtener más información, consulte ¿Qué es AWS Direct Connect?

Tráfico entre AWS recursos de la misma región

Si quieres proteger el tráfico entre los clientes de Secrets Manager y API AWS, configura y accede de forma privada AWS PrivateLinka los puntos finales 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 para generar una nueva clave de datos AWS KMS a partir de la clave 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 es necesario descifrar el secreto, Secrets Manager solicita descifrar AWS KMS 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 las claves y claves de datos de AWS KMS para proteger cada valor del secreto. Siempre que cambia el valor secreto en un secreto, Secrets Manager solicita una nueva clave de datos de 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.

Temas

- Elegir una clave AWS KMS
- ¿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

Elegir una clave AWS KMS

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 las versiones `AWSCURRENT`, `AWSPENDING` y `AWSPREVIOUS` con la nueva clave. Para evitar que descubra el secreto, Secrets Manager mantiene todas las versiones existentes cifradas con la clave anterior. Esto significa que puede descifrar las versiones `AWSCURRENT`, `AWSPENDING` y `AWSPREVIOUS` con la clave anterior o con la nueva clave. Si no tiene permiso `kms:Decrypt` para usar la clave anterior, al cambiar la clave de cifrado, Secrets Manager no podrá descifrar las versiones secretas para volver a cifrarlas. En este caso, las versiones existentes no se vuelven a cifrar.

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

Puede denegar el permiso Clave administrada de AWS `aws/secretsmanager` y exigir que los secretos estén cifrados con una clave gestionada por el cliente. Para obtener más información, consulte the section called “Ejemplo: denegar una AWS KMS clave específica para cifrar los secretos”.

Para encontrar la clave KMS asociada a un secreto, consulta el secreto en la consola o llama 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

¿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, ninguno de los Secrets Manager APIs devuelve el secreto cifrado o la clave de datos cifrados.

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 registros 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 política para un. 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 clave-valor que contienen datos arbitrarios no secretos. 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 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"
  ```

Al establecer o cambiar la clave KMS de un secreto, Secrets Manager envía GenerateDataKey y 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 rastrear 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 obtiene o cambia 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 servicio gestionado, AWS Secrets Manager está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

El acceso a Secrets Manager a través de la red se realiza mediante AWS publicación APIs mediante TLS. APIs Se puede llamar a Secrets Manager desde cualquier ubicación de la 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 usar las políticas de recursos de Secrets Manager para controlar el acceso a los secretos desde puntos finales de nube privada virtual (VPC) específicos o específicos. VPCs De hecho, esto aísla el acceso de red a un secreto determinado únicamente de la VPC específica de la red. AWS Para obtener más información, consulte the section called “Puntos de conexión de VPC (AWS PrivateLink)”.

Uso de un punto final AWS Secrets Manager de VPC

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 finales de la interfaz funcionan con una tecnología que le permite acceder de forma privada a Secrets Manager APIs sin una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una AWS Direct Connect conexión. AWS PrivateLink Las instancias de su VPC no necesitan direcciones IP públicas para comunicarse con Secrets Manager. APIs 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 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 VPCs”.

Puede usar AWS CloudTrail los registros para auditar el uso de los secretos a través del punto final de la VPC.

Para crear un punto de conexión de VPC de Secrets Manager

1. Consulte Creación de un punto de conexión de interfaz en la Guía del usuario de Amazon VPC. Usa el nombre del servicio:`com.amazonaws.region.secretsmanager`.
2. Para controlar el acceso al punto de conexión, consulte Controlar el acceso a puntos de conexión de VPC con políticas de punto de conexión.
3. Para utilizar IPv6 un direccionamiento de doble pila, consulteIPv4 y IPv6 acceso.

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.

Control de acceso a API mediante políticas de IAM

Si utilizas políticas de IAM para controlar el acceso en Servicios de AWS función de las direcciones IP, es posible que tengas que actualizar tus políticas para incluir los rangos de IPv6 direcciones. En esta guía, se explican las diferencias entre IPv4 IPv6 y se describe cómo actualizar las políticas de IAM para que sean compatibles con ambos protocolos. La implementación de estos cambios le ayuda a mantener un acceso seguro a sus AWS recursos y, al mismo tiempo, le brinda soporte IPv6.

¿Qué es IPv6?

IPv6 es el estándar IP de próxima generación que se pretende reemplazar eventualmente IPv4. La versión anterior, IPv4, utilizaba un esquema de direccionamiento de 32 bits para admitir 4.300 millones de dispositivos. IPv6 en su lugar, utiliza un direccionamiento de 128 bits para admitir aproximadamente 340 billones de billones de billones de billones de dispositivos (o 2 a la 128ª potencia).

Para obtener más información, consulte la IPv6página web de la VPC.

Estos son ejemplos de IPv6 direcciones:

```
2001:cdba:0000:0000:0000:0000:3257:9652 # This is a full, unabbreviated IPv6 address.
2001:cdba:0:0:0:0:3257:9652             # The same address with leading zeros in each
 group omitted
2001:cdba::3257:965                     # A compressed version of the same address.
```

Políticas de IAM de doble pila (IPv4 and IPv6)

Puede utilizar las políticas de IAM para controlar el acceso a Secrets Manager APIs e impedir que las direcciones IP fuera del rango configurado accedan a Secrets Manager APIs.

El administrador de secretos. El punto de conexión de doble pila {region} .amazonaws.com para Secrets Manager admite tanto como. APIs IPv6 IPv4

Si necesitas admitir ambas opciones IPv6, actualiza tus políticas IPv4 de filtrado de direcciones IP para gestionar las direcciones. IPv6 De lo contrario, es posible que no puedas conectarte a Secrets Manager a través de IPv6.

¿Quién debe realizar este cambio?

Este cambio le afecta si utiliza el direccionamiento dual con políticas que lo contienen`aws:sourceIp`. El direccionamiento doble significa que la red admite IPv4 tanto como IPv6.

Si utiliza el direccionamiento dual, actualice las políticas de IAM que actualmente utilizan direcciones de IPv4 formato para incluir las direcciones de IPv6 formato.

¿Quién no debería realizar este cambio?

Este cambio no le afecta si solo usa IPv4 redes.

Añadir IPv6 a una política de IAM

Las políticas de IAM utilizan la clave de `aws:SourceIp` condición para controlar el acceso desde direcciones IP específicas. Si su red utiliza el direccionamiento dual (IPv4 y IPv6), actualice las políticas de IAM para incluir los rangos de IPv6 direcciones.

Como `Condition` elemento de sus políticas, utilice los `NotIpAddress` operadores `IpAddress` y para las condiciones de direcciones IP. No utilices operadores de cadenas, ya que no pueden gestionar los distintos formatos de IPv6 dirección válidos.

En estos ejemplos se utilizaaws:SourceIp. Para VPCs, utilice aws:VpcSourceIp en su lugar.

La siguiente es la política de denegación del acceso a la IP de origen AWS según la política de referencia de la IP de origen de la Guía del usuario de IAM. NotIpAddressEn el Condition elemento para, se enumeran dos rangos de IPv4 direcciones 192.0.2.0/24 y 203.0.113.0/24 a cuáles se les denegará el acceso a la API.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
                    "192.0.2.0/24",
                    "203.0.113.0/24"
                ]
            },
            "Bool": {
                "aws:ViaAWSService": "false"
            }
        }
    }
}
```

Para actualizar esta política, cambie el Condition elemento para incluir los rangos de IPv6 direcciones 2001:DB8:1234:5678::/64 y2001:cdba:3257:8593::/64.

> **Note**
>
> No elimines las IPv4 direcciones existentes. Son necesarias para la compatibilidad con versiones anteriores.

```
"Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24", <<DO NOT REMOVE existing IPv4 address>>
                        "203.0.113.0/24", <<DO NOT REMOVE existing IPv4 address>>
```

```
                    "2001:DB8:1234:5678::/64", <<New IPv6 IP address>>
                    "2001:cdba:3257:8593::/64" <<New IPv6 IP address>>
                ]
            },
            "Bool": {
                "aws:ViaAWSService": "false"
            }
        }
```

Para actualizar esta política para una VPC, utilice `aws:VpcSourceIp` en lugar de: `aws:SourceIp`

```
"Condition": {
            "NotIpAddress": {
                "aws:VpcSourceIp": [
                    "10.0.2.0/24", <<DO NOT REMOVE existing IPv4 address>>
                    "10.0.113.0/24", <<DO NOT REMOVE existing IPv4 address>>
                    "fc00:DB8:1234:5678::/64", <<New IPv6 IP address>>
                    "fc00:cdba:3257:8593::/64" <<New IPv6 IP address>>
                ]
            },
            "Bool": {
                "aws:ViaAWSService": "false"
            }
        }
```

Verificar el soporte de su cliente IPv6

Si usa el administrador de secretos. Punto final {region} .amazonaws.com, comprueba que puedes conectarte a él. En los siguientes pasos, se describe cómo realizar la verificación.

Este ejemplo utiliza la versión 8.6.0 de Linux y curl, y utiliza el AWS Secrets Manager servicio que ha IPv6 habilitado los puntos de conexión ubicados en el punto de conexión de amazonaws.com.

> **Note**
>
> El administrador de secretos. {region} .amazonaws.com difiere de la típica convención de nomenclatura de doble pila. Para obtener una lista completa de los puntos finales de Secrets Manager, consulteAWS Secrets Manager puntos finales.
> Región de AWS Cámbielo a la misma región en la que se encuentra su servicio. En este ejemplo, utilizamos el punto de conexión `us-east-1` del Este de EE. UU. (Norte de Virginia)

1. Determine si el punto final se resuelve con una IPv6 dirección mediante el siguiente `dig` comando.

```
$ dig +short AAAA secretsmanager.us-east-1.amazonaws.com

> 2600:1f18:e2f:4e05:1a8a:948e:7c08:c1c3
```

2. Determine si la red del cliente puede establecer una IPv6 conexión mediante el siguiente `curl` comando. Un código de respuesta 404 indica que la conexión se realizó correctamente, mientras que un código de respuesta 0 significa que la conexión falló.

```
$ curl --ipv6 -o /dev/null --silent -w "\nremote ip: %{remote_ip}\nresponse code:
 %{response_code}\n" https://secretsmanager.us-east-1.amazonaws.com

> remote ip: 2600:1f18:e2f:4e05:1a8a:948e:7c08:c1c3
> response code: 404
```

Si se identificó una IP remota y el código de respuesta no`0`, significa que se estableció correctamente una conexión de red con el punto final mediante IPv6. La IP remota debe ser una IPv6 dirección porque el sistema operativo debe seleccionar el protocolo que sea válido para el cliente.

Si la IP remota está en blanco o el código de respuesta está en blanco`0`, la red del cliente o la ruta de red al punto final es IPv4 únicamente «-». Puede verificar esta configuración con el siguiente comando de `curl`.

```
$ curl -o /dev/null --silent -w "\nremote ip: %{remote_ip}\nresponse code:
 %{response_code}\n" https://secretsmanager.us-east-1.amazonaws.com

> remote ip: 3.123.154.250
> response code: 404
```

Resiliencia en AWS Secrets Manager

AWS crea la infraestructura global en torno a las zonas Regiones de AWS de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que se conectan con redes de baja latencia, alto rendimiento y alta redundancia. 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 la resiliencia y la recuperación ante desastres, consulte el pilar de confiabilidad: el marco de AWS buena arquitectura.

Para obtener más información sobre las zonas de disponibilidad Regiones de AWS y las zonas de disponibilidad, consulte Infraestructura global 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 contra posibles ataques futuros, AWS participa con la comunidad criptográfica en el desarrollo de algoritmos cuánticos resistentes o poscuánticos. 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. Añada el cliente AWS Common Runtime 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. Añada el AWS SDK for Java 2.x a su 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 entrada de blog AWS SDK for Java 2.x publicada.
- Presentamos s2n-tls, una nueva implementación y uso de TLS de código abierto s2n-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 poscuántico para Secrets Manager está disponible en todos los países excepto en Regiones de AWS China.

Solución de problemas 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”.

Temas

Mensajes de acceso denegado

Cuando realizas una llamada a la API, por ejemplo, GetSecretValue o CreateSecret a Secrets Manager, debes tener permisos de IAM para realizar esa llamada. Cuando utiliza la consola, esta realiza las mismas llamadas a la API en su nombre, por lo que también debe tener permisos de IAM. Un administrador puede conceder permisos asociando una política de IAM a su usuario de IAM o a un grupo del que sea miembro. Si las declaraciones de política que otorgan esos permisos incluyen alguna condición, como time-of-day restricciones de direcciones IP, también debes cumplir esos requisitos al enviar 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 the section called “Autenticación y control de acceso”.

Si firmas las solicitudes de la API de forma manual, sin AWS SDKsutilizarlas, comprueba que has firmado correctamente la solicitud.

“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 información, consulta la documentación del kit de herramientas del SDK que elijas o Cómo usar credenciales de seguridad temporales para solicitar acceso a AWS los recursos 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 the section called “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 realices en Secrets Manager (u otros AWS servicios) tarda en ser visible desde todos los puntos de conexión 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 algunos otros AWS servicios se ven afectados por la posible coherencia, 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
- Referencia sobre EC2 la coherencia eventual de Amazon en la EC2 API de Amazon

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 AWS CLI operación de nuestro AWS SDK no puede encontrar mi secreto en 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, los dos secretos son diferentes ARNs debido a estos caracteres. Los usuarios con acceso al secreto anterior no acceden automáticamente al secreto nuevo porque ARNs 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 lo administra un AWS servicio y debes usarlo 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 campo AWS CLI, llama a describe-secret y, a continuación, revisa el campo. `OwningService`

La importación del módulo Python falla cuando se usa **`Transform: AWS::SecretsManager-2024-09-16`**

Si está utilizando el módulo Transform: `AWS::SecretsManager-2024-09-16` y encuentra errores de importación del módulo Python cuando se ejecuta la función Lambda de rotación, es probable que el problema se deba a un valor incompatible`Runtime`. Con esta versión de transformación, AWS CloudFormation administra automáticamente la versión en tiempo de ejecución, el código y los archivos de objetos compartidos. No es necesario que los gestione usted mismo.

AWS Secrets Manager cuotas

Secrets Manager lee APIs tiene cuotas de TPS altas y los planos de control, APIs que se denominan con menos frecuencia, tienen cuotas de 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 aplicación Cuenta de AWS utiliza un secreto propiedad de otra cuenta, se denomina solicitud de cuentas cruzadas. 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 DeleteResourcePolicy solicitudes GetResourcePolicy PutResourcePolicy, y de ValidateResourcePolicy API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para DeleteResourcePolicy, GetResourcePolicy PutResourcePolicy, y las solicitudes de ValidateResourcePolicy API combinadas.
Tasa combinada de solicitudes PutSecretValue RemoveRegionsFromReplication, ReplicateSecretToRegion, StopReplicationToReplica UpdateSec	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de PutSecret

Nombre	Valor predeterminado	Ajusta	Descripción
ret, y a UpdateSecretVersionStage la API			Value RemoveRegionsFromReplication ReplicateSecretToRegion StopReplicationToReplica UpdateSecret,,, y UpdateSecretVersionStage API combinadas.
Tasa combinada de solicitudes de RestoreSecret API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de RestoreSecret API.
Tasa combinada de solicitudes a la CancelRotateSecret API RotateSecret y a la API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de CancelRotateSecret API RotateSecret y las solicitudes de API combinadas.
Tasa combinada de solicitudes a la UntagResource API TagResource y a la API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de UntagResource API TagResource y las solicitudes de API combinadas.
Tasa de solicitudes a BatchGetSecretValue la API	Cada región admitida: 100 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de BatchGetSecretValue API.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes a CreateSecret la API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de CreateSecret API.
Tasa de solicitudes a DeleteSecret la API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de DeleteSecret API.
Tasa de solicitudes a DescribeSecret la API	Cada región compatible: 40 000 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de DescribeSecret API.
Tasa de solicitudes a GetRandom Password la API	Cada región admitida: 50 por segundo	No	El número máximo de todas las demás solicitudes de la API de Secrets Manager que se pueden hacer por segundo en esta cuenta.
Tasa de solicitudes a GetSecretValue la API	Cada región admitida: 10 000 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de GetSecretValue API.
Tasa de solicitudes a ListSecretVersionIds la API	Cada región admitida: 50 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de ListSecretVersionIds API.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes a ListSecrets la API	Cada región admitida: 100 por segundo	No	El número máximo de transacciones por segundo para las solicitudes de ListSecrets API.
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.
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	El número máximo de secretos en cada AWS región de esta AWS cuenta.
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

Es posible que su AWS cliente vea que las llamadas a Secrets Manager fallan debido a problemas inesperados por parte 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 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
- Se produce un error al volver a intentarlo y se produce un retraso exponencial. AWS

Historial de documentos

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
Soporte agregado IPv6 y de doble pila	Secrets Manager ahora admite puntos finales de doble pila. Consulte IPv4 y IPv6 acceda a ella para obtener más información.	20 de diciembre de 2024
Cambio de Secrets Manager a una política AWS gestionada	La política de `SecretsManagerReadWrite` administrada incluye permisos `redshift-serverless` . 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 versión pública inicial 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.