aws

Panduan Pengguna

AWS Secrets Manager

AWS Secrets Manager: Panduan Pengguna

Table of Contents

Apa itu AWS Secrets Manager?

AWS Secrets Manager membantu Anda mengelola, mengambil, dan memutar kredensi database, kredensi aplikasi, token OAuth, kunci API, dan rahasia lainnya sepanjang siklus hidupnya. Banyak AWS layanan menyimpan dan menggunakan rahasia di Secrets Manager.

Secrets Manager membantu Anda meningkatkan postur keamanan Anda, karena Anda tidak lagi memerlukan kredensi hard-code dalam kode sumber aplikasi. Menyimpan kredensi di Secrets Manager membantu menghindari kemungkinan kompromi oleh siapa saja yang dapat memeriksa aplikasi atau komponen Anda. Anda mengganti kredensi hard-code dengan panggilan runtime ke layanan Secrets Manager untuk mengambil kredenal secara dinamis saat Anda membutuhkannya.

Dengan Secrets Manager, Anda dapat mengonfigurasi jadwal rotasi otomatis untuk rahasia Anda. Ini memungkinkan Anda untuk mengganti rahasia jangka panjang dengan rahasia jangka pendek, secara signifikan mengurangi risiko kompromi. Karena kredensil tidak lagi disimpan dengan aplikasi, kredensil berputar tidak lagi memerlukan pembaruan aplikasi Anda dan menerapkan perubahan ke klien aplikasi.

Untuk jenis rahasia lain yang mungkin Anda miliki di organisasi Anda:

- AWS kredensi — Kami merekomendasikan. AWS Identity and Access Management
- Kunci enkripsi — Kami merekomendasikan AWS Key Management Service.
- Kunci SSH - Kami merekomendasikan Amazon EC2 Instance Connect.
- Kunci pribadi dan sertifikat — Kami merekomendasikan AWS Certificate Manager.

Memulai Secrets Manager

Jika Anda baru mengenal Secrets Manager, mulailah dengan Konsep atau salah satu tutorial berikut:

- the section called “Ganti rahasia hardcode ”
- the section called “Ganti kredensyal DB hardcode ”
- the section called “Rotasi pengguna secara bergantian”
- the section called “Rotasi pengguna tunggal”

Tugas lain yang dapat Anda lakukan dengan rahasia:

- Buat dan kelola rahasia
- Kontrol akses ke rahasia Anda
- Ambil rahasia
- Putar rahasia
- Memantau rahasia
- Rahasia audit untuk kepatuhan
- Buat rahasia di AWS CloudFormation

Kepatuhan dengan standar

AWS Secrets Manager telah menjalani audit untuk berbagai standar dan dapat menjadi bagian dari solusi Anda ketika Anda perlu mendapatkan sertifikasi kepatuhan. Untuk informasi selengkapnya, lihat Validasi Kepatuhan.

Harga

Saat Anda menggunakan Secrets Manager, Anda hanya membayar untuk apa yang Anda gunakan, tanpa biaya minimum atau pengaturan. Tidak ada biaya untuk rahasia yang ditandai untuk dihapus. Untuk daftar harga lengkap saat ini, lihat AWS Secrets Manager Harga.

Anda dapat menggunakan Secrets Manager Kunci yang dikelola AWS `aws/secretsmanager` yang dibuat untuk mengenkripsi rahasia Anda secara gratis. Jika Anda membuat kunci KMS Anda sendiri untuk mengenkripsi rahasia Anda, AWS menagih Anda dengan tarif saat ini AWS KMS . Untuk informasi selengkapnya, silakan lihat HargaAWS Key Management Service.

Ketika Anda mengaktifkan rotasi otomatis (kecuali rotasi terkelola), Secrets Manager menggunakan AWS Lambda fungsi untuk memutar rahasia, dan Anda dikenakan biaya untuk fungsi rotasi pada tingkat Lambda saat ini. Untuk informasi selengkapnya, silakan lihat HargaAWS Lambda.

Jika Anda mengaktifkan AWS CloudTrail di akun, Anda bisa mendapatkan log panggilan API yang dikirimkan Secrets Manager. Secrets Manager mencatat semua peristiwa sebagai acara manajemen. AWS CloudTrail menyimpan salinan pertama dari semua acara manajemen secara gratis. Namun, Anda dapat dikenakan biaya untuk Amazon S3 untuk penyimpanan log dan untuk Amazon SNS jika Anda mengaktifkan notifikasi. Juga, jika Anda mengatur jalur tambahan, salinan tambahan dari acara manajemen dapat menimbulkan biaya. Untuk informasi selengkapnya, lihat hargaAWS CloudTrail.

AWS Layanan yang Menggunakan AWS Secrets Manager Rahasia

- AWS App Runner— Lihat Merujuk variabel lingkungan dan Mengelola variabel lingkungan di PanduanAWS App Runner Pengembang.
- AWS App2Container - Lihat Kelola rahasia untuk AWS App2Container di Panduan Penggunaan App2Container.AWS
- AWS AppConfig— Lihat Membuat profil konfigurasi bentuk bebas di PanduanAWS AppConfig Pengguna.
- Amazon AppFlow — LihatRahasia yang dikelola oleh layanan lain.
- AWS AppSync— Lihat Tutorial: Aurora Tanpa Server di Panduan Pengembang.AWS AppSync
- Amazon Athena — Lihat Menggunakan Kueri Federasi Amazon Athena di Panduan Pengguna Amazon Athena.
- Amazon Aurora — Lihat Lihat. Rahasia yang dikelola oleh layanan lain
- AWS CodeBuild— Lihat Registri pribadi dengan AWS Secrets Manager sampel untuk CodeBuild di PanduanAWS CodeBuild Pengguna.
- AWS DataSync – Lihat Rahasia yang dikelola oleh layanan lain.
- Amazon DataZone — Lihat Membuat sumber data untuk database Amazon Redshift menggunakan AWS Glue koneksi baru di DataZone Panduan Pengguna Amazon.
- AWS Direct Connect – Lihat Rahasia yang dikelola oleh layanan lain.
- AWS Directory Service— Lihat menggabungkan instans EC2 Linux dengan mulus ke direktori AWS Microsoft AD Terkelola Anda, menggabungkan instans EC2 Linux dengan mulus ke direktori AD Connector Anda, dan bergabung dengan instans Linux EC2 dengan mulus ke direktori Simple AD Anda di Panduan Pengguna.AWS Direct Connect
- Amazon DocumentDB (dengan kompatibilitas MongoDB) - Lihat dan the section called “Buat rahasia database” Mengelola Pengguna Amazon DocumentDB di Panduan Pengembang Amazon DocumentDB.
- AWS Elastic Beanstalk— Lihat konfigurasi Docker di PanduanAWS Elastic Beanstalk Pengembang.
- Amazon Elastic Container Registry — Lihat Membuat aturan cache tarik melalui di Panduan Pengguna Amazon ECR.
- Amazon Elastic Container Service - Lihat Tutorial: Menentukan data sensitif menggunakan rahasia Secrets Manager, Mengambil rahasia secara terprogram melalui aplikasi Anda, Mengambil rahasia melalui variabel lingkungan, Mengambil rahasia untuk konfigurasi logging, Tutorial: Menggunakan

sistem file FSx for Windows File Server dengan Amazon ECS, volume fsX for Windows File Server , dan otentikasi registri pribadi untuk tugas-tugas di Amazon Elastic Panduan Pengembang Layanan Kontainer.

- Amazon Elastic Container Service Service Connect — LihatRahasia yang dikelola oleh layanan lain.
- Amazon ElastiCache — Lihat Kata sandi yang berputar secara otomatis untuk pengguna di Panduan ElastiCache Pengguna Amazon.
- AWS Elemental Live— Lihat Cara pengiriman dari AWS Elemental Live ke MediaConnect bekerja saat runtime di Panduan Pengguna Elemental Live.
- AWS Elemental MediaConnect— Lihat Enkripsi kunci statis AWS Elemental MediaConnectdi PanduanAWS Elemental MediaConnect Pengguna.
- AWS Elemental MediaConvert— Lihat Menggunakan Kantar untuk watermarking audio dalam AWS Elemental MediaConvert output di Panduan Pengguna.AWS Elemental MediaConvert
- AWS Elemental MediaLive— Lihat Menyiapkan MediaLive sebagai entitas tepercaya di Panduan MediaLive Pengguna.
- AWS Elemental MediaPackage— Lihat akses Secrets Manager untuk otorisasi CDN di AWS Elemental MediaPackage Panduan Pengguna.
- AWS Elemental MediaTailor— Lihat Mengonfigurasi otentikasi token AWS Secrets Manager akses di AWS Elemental MediaTailor Panduan Pengguna.
- Amazon EMR berjalan di Amazon EC2 - Lihat Simpan data konfigurasi sensitif di Secrets Manager dan Tambahkan Repositori berbasis Git ke Amazon EMR di Panduan Manajemen EMR Amazon.
- EMR Tanpa Server - Lihat Secrets Manager untuk perlindungan data dengan EMR Tanpa Server di Panduan Pengguna Tanpa Server Amazon EMR.
- Amazon EventBridge — LihatRahasia yang dikelola oleh layanan lain.
- Amazon FSx - Lihat Berbagi file dan Memigrasi konfigurasi berbagi file ke Amazon FSx di Panduan Pengguna Amazon FSx for Windows File Server.
- AWS Glue DataBrew – Lihat Rahasia yang dikelola oleh layanan lain.
- AWS Glue Studio - Lihat Tutorial: Menggunakan Konektor AWS Glue untuk Elasticsearch di Panduan AWS Glue Pengembang.
- AWS IoT SiteWise— Lihat Mengonfigurasi otentikasi sumber data di AWS IoT SiteWise Panduan Pengguna.
- Amazon Kendra — Lihat Menggunakan sumber data database di Panduan Pengguna Amazon Kendra.

- Amazon Kinesis Video Streams — Lihat Menyebarkan Agen Edge Streams Video Kinesis Amazon ke dalam Panduan AWS IoT GreengrassPengembang Amazon Kinesis Video Streams.
- AWS Launch Wizard— Lihat Mengatur AWS Launch Wizard untuk Active Directory di PanduanAWS Launch Wizard Pengguna.
- Amazon Lookout for Metrics — Lihat Menggunakan Amazon RDS dengan Lookout for Metrics dan Menggunakan Amazon Redshift dengan Lookout for Metrics di Amazon Lookout for Metrics.
- Grafana Terkelola Amazon — Lihat Mengonfigurasi Amazon Redshift di Panduan Pengguna Grafana Terkelola Amazon.
- AWS Managed Services— Lihat AWS Secrets Manager (penyediaan layanan mandiri AMS) di Panduan Pengguna AMS Advanced.
- Amazon Managed Streaming for Apache Kafka - Lihat Otentikasi nama pengguna dan kata sandi dengan AWS Secrets Manager di Panduan Pengembang Amazon Managed Streaming for Apache Kafka.
- Alur Kerja Terkelola Amazon untuk Apache Airflow — Lihat Mengonfigurasi koneksi Apache Airflow menggunakan rahasia Secrets Manager dan Menggunakan kunci rahasia untuk variabel Apache Airflow di Amazon Managed Workflows AWS Secrets Manager for Apache Airflow User Guide.
- AWS Marketplace – Lihat Rahasia yang dikelola oleh layanan lain.
- AWS Migration Hub— Lihat Migrasi NetWeaver aplikasi SAP ke AWSdan Rehost aplikasi di Amazon EC2 di Panduan Pengguna Orchestrator.AWS Migration Hub
- AWS OpsWorks for Chef Automate – Lihat Rahasia yang dikelola oleh layanan lain.
- AWS Panorama— Lihat Mengelola aliran kamera AWS Panoramadi PanduanAWS Panorama Pengembang.
- AWS ParallelCluster— Lihat Mengintegrasikan Direktori Aktif di PanduanAWS ParallelCluster Pengguna.
- Amazon Q - Lihat Konsep - Otentikasi di Panduan Pengembang Amazon Q.
- Amazon QuickSight — Lihat Menggunakan AWS Secrets Manager rahasia sebagai pengganti kredensi basis data QuickSight di Amazon di QuickSight Panduan Pengguna Amazon.
- Amazon RDS — LihatRahasia yang dikelola oleh layanan lain.
- Amazon Redshift — LihatRahasia yang dikelola oleh layanan lain,the section called “Buat rahasia database”, Menyimpan kredenal database di AWS Secrets Manager, Menggunakan Amazon Redshift Data API, dan Query database menggunakan editor kueri di Panduan Manajemen Amazon Redshift.
- Editor kueri Amazon Redshift v2 — Lihat. Rahasia yang dikelola oleh layanan lain

- Amazon SageMaker — Lihat Mengaitkan Repositori Git dengan Instans SageMaker Notebook Amazon, Mengimpor data dari Databricks (JDBC), dan Impor data dari Snowflake di Panduan Pengembang Amazon. SageMaker
- AWS Schema Conversion Tool— Lihat Menggunakan AWS Secrets Manager di antarmuka AWS SCT pengguna di PanduanAWS Schema Conversion Tool Pengguna.
- AWS Toolkit for JetBrains— Lihat Mengakses cluster Amazon Redshift diAWS Toolkit for JetBrains Panduan Pengguna.
- AWS Transfer Family— Lihat otentikasi dasar untuk konektor AS2, Bekerja dengan penyedia identitas khusus, dan Menghasilkan dan mengelola kunci PGP di Panduan Pengguna.AWS Transfer Family
- AWS Wickr — Lihat Mulai bot retensi data di Panduan Administrasi AWS Wickr.

Akses AWS Secrets Manager

Anda dapat bekerja dengan Secrets Manager dengan salah satu cara berikut:

- Konsol Secrets Manager
- Alat baris perintah
- AWS SDK
- API Kueri HTTPS
- AWS Secrets Manager titik akhir

Konsol Secrets Manager

Anda dapat mengelola rahasia Anda menggunakan konsol Secrets Manager berbasis browser dan melakukan hampir semua tugas yang terkait dengan rahasia Anda dengan menggunakan konsol.

Alat baris perintah

Alat baris AWS perintah memungkinkan Anda mengeluarkan perintah di baris perintah sistem Anda untuk melakukan Secrets Manager dan AWS tugas lainnya. Ini mungkin lebih cepat dan nyaman dibandingkan jika menggunakan konsol. Alat baris perintah dapat berguna jika Anda ingin membangun skrip untuk melakukan AWS tugas.

Saat Anda memasukkan perintah di shell perintah, ada risiko riwayat perintah diakses atau utilitas memiliki akses ke parameter perintah Anda. Lihat the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

Alat baris perintah secara otomatis menggunakan titik akhir default untuk layanan di AWS Wilayah. Anda dapat menentukan titik akhir yang berbeda untuk permintaan API Anda. Lihat the section called “Titik akhir Secrets Manager”.

AWS menyediakan dua set alat baris perintah:

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

AWS SDK

AWS SDK terdiri dari pustaka dan kode sampel untuk berbagai bahasa dan platform pemrograman. SDK mencakup tugas-tugas seperti menandatangani permintaan secara kriptografis, mengelola kesalahan, dan mencoba ulang permintaan secara otomatis. Untuk mengunduh dan menginstal salah satu SDK, lihat Alat untuk Amazon Web Services.

AWS SDK secara otomatis menggunakan titik akhir default untuk layanan di Wilayah. AWS Anda dapat menentukan titik akhir yang berbeda untuk permintaan API Anda. Lihat the section called “Titik akhir Secrets Manager”.

Untuk dokumentasi SDK, lihat:

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

API Kueri HTTPS

HTTPS Query API memberi Anda akses terprogram ke Secrets Manager dan AWS. HTTPS Query API memungkinkan Anda untuk mengeluarkan permintaan HTTPS langsung ke layanan.

Meskipun Anda dapat melakukan panggilan langsung ke Secrets Manager HTTPS Query API, sebaiknya gunakan salah satu SDK sebagai gantinya. SDK melakukan banyak tugas berguna yang harus Anda lakukan secara manual. Misalnya, SDK secara otomatis menandatangani permintaan Anda dan mengonversi respons menjadi struktur yang secara sintaksis sesuai dengan bahasa Anda.

Untuk melakukan panggilan HTTPS ke Secrets Manager, Anda terhubung ke???.

AWS Secrets Manager titik akhir

Untuk terhubung secara terprogram ke Secrets Manager, Anda menggunakan endpoint, URL titik masuk untuk layanan. Secrets Manager endpoint adalah endpoint dual-stack, yang berarti mereka mendukung IPv4 dan IPv6.

Secrets Manager menawarkan titik akhir yang mendukung Federal Information Processing Standard (FIPS) 140-2 di beberapa Wilayah.

Secrets Manager mendukung TLS 1.2 dan 1.3. Secrets Manager mendukung PQTLS di semua wilayah kecuali Wilayah China.

> **Note**
>
> Python AWS SDK dan AWS CLI upaya untuk memanggil IPv6 dan kemudian IPv4 secara berurutan, jadi jika Anda tidak mengaktifkan IPv6, diperlukan beberapa waktu sebelum waktu panggilan habis dan mencoba lagi dengan IPv4. Untuk mengatasi masalah ini, Anda dapat menonaktifkan IPv6 sepenuhnya atau bermigrasi ke IPv6.

Berikut ini adalah endpoint layanan untuk Secrets Manager. Perhatikan bahwa penamaan berbeda dari konvensi penamaan dual-stack yang khas.

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
AS Timur (Ohio)	us–east-2	secretsmanager.us-east-2.amazonaws.com secretsmanager-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
AS Timur (Virginia Utara)	us-east-1	secretsmanager.us-east-1.amazonaws.com secretsmanager-fips.us-east-1.amazonaws.com	HTTPS HTTPS	

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
AS Barat (California Utara)	us-west-1	secretsmanager.us-west-1.amazonaws.com secretsmanager-fips.us-west-1.amazonaws.com	HTTPS HTTPS	
US West (Oregon)	us-west-2	secretsmanager.us-west-2.amazonaws.com secretsmanager-fips.us-west-2.amazonaws.com	HTTPS HTTPS	
Africa (Cape Town)	af-south-1	secretsmanager.af-south-1.amazonaws.com	HTTPS	
Asia Pasifik (Hong Kong)	ap-east-1	secretsmanager.ap-east-1.amazonaws.com	HTTPS	
Asia Pasifik (Hyderabad)	ap-south-2	secretsmanager.ap-south-2.amazonaws.com	HTTPS	
Asia Pasifik (Jakarta)	ap-southeast-3	secretsmanager.ap-southeast-3.amazonaws.com	HTTPS	
Asia Pasifik (Melbourne)	ap-southeast-4	secretsmanager.ap-southeast-4.amazonaws.com	HTTPS	
Asia Pasifik (Mumbai)	ap-south-1	secretsmanager.ap-south-1.amazonaws.com	HTTPS	

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Osaka)	ap-northeast-3	secretsmanager.ap-northeast-3.amazonaws.com	HTTPS
Asia Pasifik (Seoul)	ap-northeast-2	secretsmanager.ap-northeast-2.amazonaws.com	HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	secretsmanager.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	secretsmanager.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	secretsmanager.ap-northeast-1.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	secretsmanager.ca-central-1.amazonaws.com secretsmanager-fips.ca-central-1.amazonaws.com	HTTPS HTTPS
Kanada Barat (Calgary)	ca-west-1	secretsmanager.ca-west-1.amazonaws.com secretsmanager-fips.ca-west-1.amazonaws.com	HTTPS HTTPS
Eropa (Frankfurt)	eu-central-1	secretsmanager.eu-central-1.amazonaws.com	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
Eropa (Irlandia)	eu-west-1	secretsmanager.eu-west-1.amazonaws.com	HTTPS	
Europe (London)	eu-west-2	secretsmanager.eu-west-2.amazonaws.com	HTTPS	
Eropa (Milan)	eu-south-1	secretsmanager.eu-south-1.amazonaws.com	HTTPS	
Eropa (Paris)	eu-west-3	secretsmanager.eu-west-3.amazonaws.com	HTTPS	
Eropa (Spanyol)	eu-south-2	secretsmanager.eu-south-2.amazonaws.com	HTTPS	
Eropa (Stockholm)	eu-north-1	secretsmanager.eu-north-1.amazonaws.com	HTTPS	
Eropa (Zürich)	eu-central-2	secretsmanager.eu-central-2.amazonaws.com	HTTPS	
Israel (Tel Aviv)	il-central-1	secretsmanager.il-central-1.amazonaws.com	HTTPS	
Timur Tengah (Bahrain)	me-south-1	secretsmanager.me-south-1.amazonaws.com	HTTPS	
Timur Tengah (UAE)	me-central-1	secretsmanager.me-central-1.amazonaws.com	HTTPS	

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
Amerika Selatan (Sao Paulo)	sa-east-1	secretsmanager.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (AS-Timur)	us-gov-east-1	secretsmanager.us-gov-east-1.amazonaws.com secretsmanager-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS	
AWS GovCloud (AS-Barat)	us-gov-west-1	secretsmanager.us-gov-west-1.amazonaws.com secretsmanager-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS	

Konsep AWS Secrets Manager

Konsep-konsep berikut ini penting untuk memahami cara kerja Secrets Manager.

- Rahasia
- Versi
- Rotasi
- Strategi rotasi

Rahasia

Dalam Secrets Manager, rahasia terdiri dari informasi rahasia, nilai rahasia, ditambah metadata tentang rahasia. Nilai rahasia dapat berupa string atau biner. Untuk menyimpan beberapa nilai string dalam satu rahasia, kami sarankan Anda menggunakan string teks JSON dengan pasangan kunci/nilai, misalnya:

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

Metadata rahasia meliputi:

- Nama Sumber Daya Amazon (ARN) dengan format berikut:

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

  Secrets Manager mencakup enam karakter acak di akhir nama rahasia untuk membantu memastikan bahwa ARN rahasia itu unik. Jika rahasia asli dihapus, dan kemudian rahasia baru dibuat dengan nama yang sama, kedua rahasia memiliki ARN yang berbeda karena karakter ini. Pengguna dengan akses ke rahasia lama tidak secara otomatis mendapatkan akses ke rahasia baru karena ARN berbeda.
- Nama rahasia, deskripsi, kebijakan sumber daya, dan tag.

- ARN untuk kunci enkripsi, yang digunakan Secrets Manager untuk mengenkripsi dan mendekripsi nilai rahasia. AWS KMS key Secrets Manager menyimpan teks rahasia dalam bentuk terenkripsi dan mengenkripsi rahasia dalam perjalanan. Lihat the section called “Enkripsi rahasia dan dekripsi”.
- Informasi tentang cara memutar rahasia, jika Anda mengatur rotasi. Lihat the section called “Rotasi”.

Secrets Manager menggunakan kebijakan izin IAM untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses atau memodifikasi rahasia. Lihat Kontrol autentikasi dan akses untuk AWS Secrets Manager.

Sebuah rahasia memiliki versi yang menyimpan salinan dari nilai rahasia terenkripsi. Ketika Anda mengubah nilai rahasia, atau rahasia diputar, Secrets Manager membuat versi baru. Lihat the section called “Versi”.

Anda dapat menggunakan rahasia di beberapa Wilayah AWS dengan mereplikasi itu. Ketika Anda mereplikasi rahasia, Anda membuat salinan rahasia asli atau primer yang disebut rahasia replika. Rahasia replika tetap terkait dengan rahasia utama. Lihat the section called “Replikasi rahasia ke Wilayah lain”.

Lihat Buat dan kelola rahasia.

Versi

Sebuah rahasia memiliki versi yang menyimpan salinan dari nilai rahasia terenkripsi. Ketika Anda mengubah nilai rahasia, atau rahasia diputar, Secrets Manager membuat versi baru.

Secrets Manager tidak menyimpan riwayat rahasia linier dengan versi. Sebagai gantinya, ia melacak tiga versi tertentu dengan memberi label:

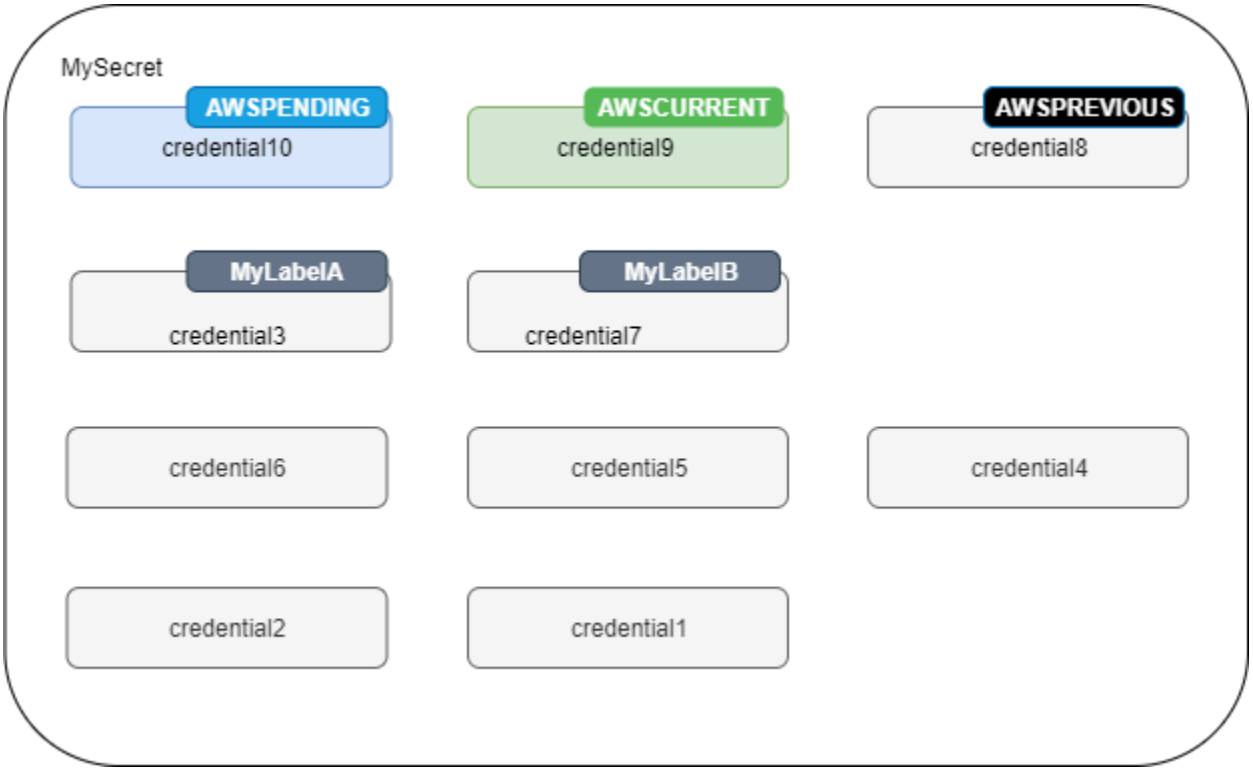
- Versi saat ini - `AWSCURRENT`
- Versi sebelumnya - `AWSPREVIOUS`
- Versi yang tertunda (selama rotasi) - `AWSPENDING`

Rahasia selalu memiliki versi berlabel`AWSCURRENT`, dan Secrets Manager mengembalikan versi tersebut secara default saat Anda mengambil nilai rahasia.

Anda juga dapat memberi label versi dengan label Anda sendiri `update-secret-version-stage`dengan memanggilAWS CLI. Anda dapat melampirkan hingga 20 label ke versi secara rahasia. Dua versi rahasia tidak dapat memiliki label pementasan yang sama. Versi dapat memiliki beberapa label.

Secrets Manager tidak pernah menghapus versi berlabel, tetapi versi yang tidak berlabel dianggap usang. Secrets Manager menghapus versi usang ketika ada lebih dari 100. Secrets Manager tidak menghapus versi yang dibuat kurang dari 24 jam yang lalu.

Gambar berikut menunjukkan rahasia yang memiliki versi AWS berlabel dan versi berlabel pelanggan. Versi tanpa label dianggap usang dan akan dihapus oleh Secrets Manager di beberapa titik di masa mendatang.



Rotasi

Rotasi adalah proses memperbarui rahasia secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensialnya. Di Secrets Manager, Anda dapat mengatur rotasi otomatis untuk rahasia Anda. Ketika Secrets Manager memutar rahasia, ia memperbarui kredensialnya baik dalam rahasia maupun database atau layanan. Lihat Putar rahasia.

> **Tip**
>
> Untuk beberapaRahasia yang dikelola oleh layanan lain, Anda menggunakan rotasi terkelola. Untuk menggunakanRotasi terkelola, Anda pertama kali membuat rahasia melalui layanan pengelolaan.

Strategi rotasi

Secrets Manager menawarkan dua strategi rotasi:

- Strategi rotasi: pengguna tunggal
- Strategi rotasi: pengguna bergantian

Strategi rotasi: pengguna tunggal

Strategi ini memperbarui kredensil untuk satu pengguna dalam satu rahasia. Untuk instans Amazon RDS Db2, karena pengguna tidak dapat mengubah kata sandi mereka sendiri, Anda harus memberikan kredensi admin dalam rahasia terpisah. Ini adalah strategi rotasi paling sederhana, dan cocok untuk sebagian besar kasus penggunaan. Secara khusus, kami menyarankan Anda menggunakan strategi ini untuk kredensil untuk satu kali (ad hoc) atau pengguna interaktif.

Ketika rahasia berputar, koneksi database terbuka tidak terputus. Sementara rotasi sedang terjadi, ada periode waktu singkat antara ketika kata sandi dalam database berubah dan ketika rahasia diperbarui. Selama waktu ini, ada risiko rendah database menolak panggilan yang menggunakan kredenal yang diputar. Anda dapat mengurangi risiko ini dengan strategi coba lagi yang tepat. Setelah rotasi, koneksi baru menggunakan kredensil baru.

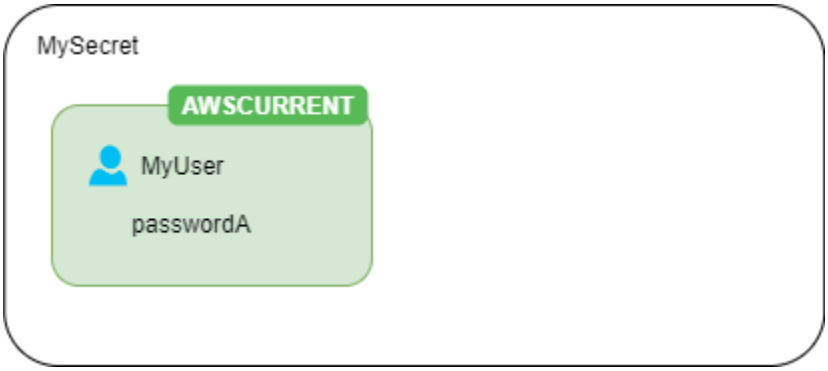
Strategi rotasi: pengguna bergantian

Strategi ini memperbarui kredensil untuk dua pengguna dalam satu rahasia. Anda membuat pengguna pertama, dan selama rotasi pertama, fungsi rotasi mengkloningnya untuk membuat pengguna kedua. Setiap kali rahasia berputar, fungsi rotasi mengganti kata sandi pengguna mana yang diperbarui. Karena sebagian besar pengguna tidak memiliki izin untuk mengkloning diri mereka sendiri, Anda harus memberikan kredensialnya untuk rahasia lain. `superuser` Sebaiknya gunakan strategi rotasi pengguna tunggal ketika pengguna kloning di database Anda tidak memiliki izin yang sama dengan pengguna asli, dan untuk kredensi untuk pengguna satu kali (ad hoc) atau interaktif.

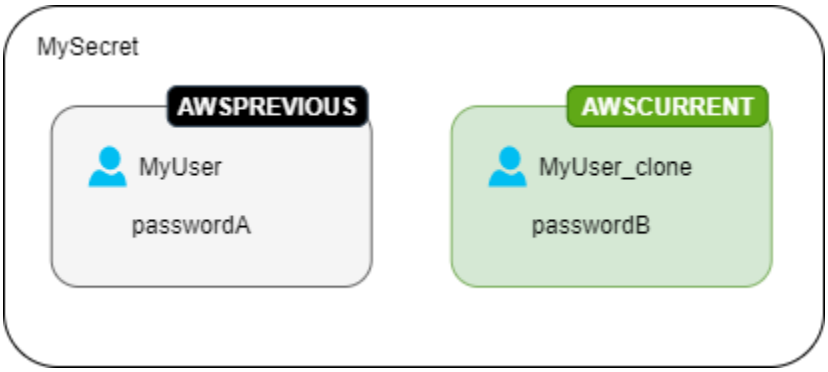
Strategi ini sesuai untuk database dengan model izin di mana satu peran memiliki tabel database dan peran kedua memiliki izin untuk mengakses tabel database. Ini juga sesuai untuk aplikasi yang membutuhkan ketersediaan tinggi. Jika aplikasi mengambil rahasia selama rotasi, aplikasi masih mendapatkan set kredensil yang valid. Setelah rotasi, keduanya `user` dan `user_clone` kredensialnya valid. Bahkan ada lebih sedikit kemungkinan aplikasi mendapatkan penolakan selama jenis rotasi ini daripada rotasi pengguna tunggal. Jika database di-host di server farm di mana perubahan kata sandi membutuhkan waktu untuk menyebar ke semua server, ada risiko database menolak panggilan yang menggunakan kredensi baru. Anda dapat mengurangi risiko ini dengan strategi coba lagi yang tepat.

Secrets Manager membuat pengguna kloning dengan izin yang sama dengan pengguna asli. Jika Anda mengubah izin pengguna asli setelah klon dibuat, Anda juga harus mengubah izin pengguna kloning.

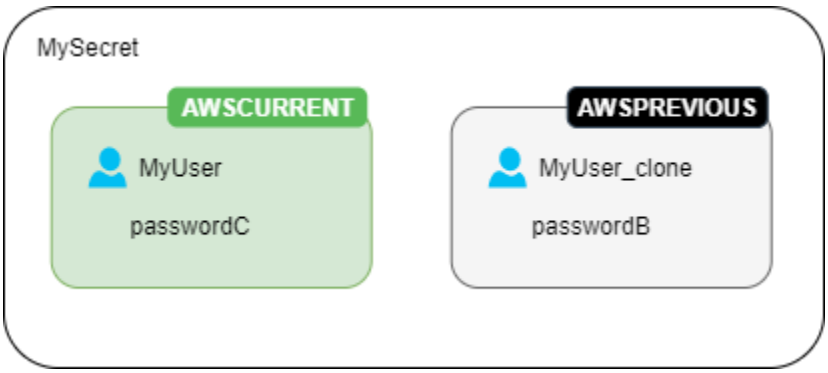
Misalnya, jika Anda membuat rahasia dengan kredensi pengguna database, rahasia berisi satu versi dengan kredensialnya.



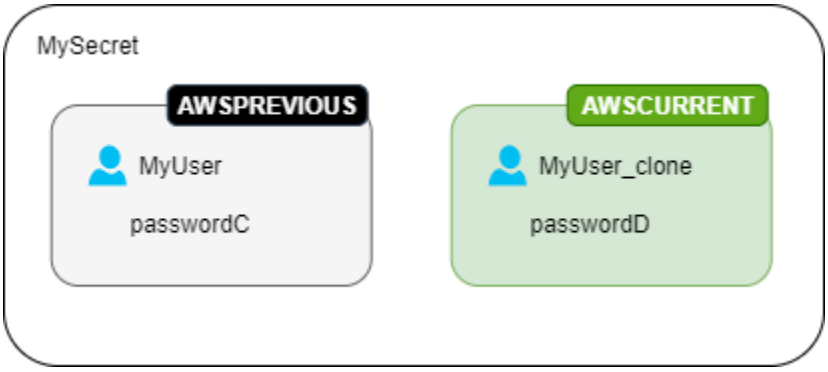
Rotasi pertama - Fungsi rotasi membuat tiruan pengguna Anda dengan kata sandi yang dihasilkan, dan kredensi tersebut menjadi versi rahasia saat ini.



Rotasi kedua - Fungsi rotasi memperbarui kata sandi untuk pengguna asli.



Rotasi ketiga - Fungsi rotasi memperbarui kata sandi untuk pengguna kloning.



Tutorial AWS Secrets Manager

Topik

Temukan rahasia yang tidak dilindungi dalam kode Anda dengan Amazon Reviewer CodeGuru

Amazon CodeGuru Reviewer adalah layanan yang menggunakan analisis program dan pembelajaran mesin untuk mendeteksi potensi cacat yang sulit ditemukan oleh pengembang dan menawarkan saran untuk meningkatkan kode Java dan Python Anda. CodeGuru Reviewer terintegrasi dengan Secrets Manager untuk menemukan rahasia yang tidak dilindungi dalam kode Anda. Untuk jenis rahasia yang dapat ditemukan, lihat Jenis rahasia yang terdeteksi oleh CodeGuru Reviewer di Panduan Pengguna Amazon CodeGuru Reviewer.

Setelah Anda menemukan rahasia hardcode, ambil tindakan untuk menggantinya:

- the section called “Ganti kredensyal DB hardcode ”
- the section called “Ganti rahasia hardcode ”

Pindahkan rahasia hardcode ke AWS Secrets Manager

Jika Anda memiliki rahasia plaintext dalam kode Anda, kami sarankan Anda memutarnya dan menyimpannya di Secrets Manager. Memindahkan rahasia ke Secrets Manager memecahkan masalah rahasia yang terlihat oleh siapa saja yang melihat kode, karena ke depan, kode Anda mengambil rahasia langsung dari Secrets Manager. Memutar rahasia mencabut rahasia hardcode saat ini sehingga tidak lagi valid.

Untuk rahasia kredensyal basis data, lihatPindahkan kredensyal basis data hardcode ke AWS Secrets Manager.

Sebelum Anda mulai, Anda perlu menentukan siapa yang membutuhkan akses ke rahasia. Sebaiknya gunakan dua peran IAM untuk mengelola izin rahasia Anda:

- Peran yang mengelola rahasia dalam organisasi Anda. Untuk informasi selengkapnya, lihat the section called “Izin administrator Secrets Manager”. Anda akan membuat dan memutar rahasia menggunakan peran ini.
- Peran yang dapat menggunakan rahasia saat runtime, misalnya dalam tutorial ini yang Anda gunakan `RoleToRetrieveSecretAtRuntime`. Kode Anda mengasumsikan peran ini untuk mengambil rahasia. Dalam tutorial ini, Anda memberikan peran hanya izin untuk mengambil satu nilai rahasia, dan Anda memberikan izin dengan menggunakan kebijakan sumber daya rahasia. Untuk alternatif lain, lihatthe section called “Langkah selanjutnya”.

Langkah:

- Langkah 1: Buat Rahasianya
- Langkah 2: Perbarui kode Anda
- Langkah 3: Perbarui rahasianya
- Langkah selanjutnya

Langkah 1: Buat Rahasianya

Langkah pertama adalah menyalin rahasia hardcode yang ada ke Secrets Manager. Jika rahasianya terkait dengan AWS sumber daya, simpan di Wilayah yang sama dengan sumber daya. Jika tidak, simpan di Wilayah yang memiliki latensi terendah untuk kasus penggunaan Anda.

Untuk membuat rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih jenis rahasia, lakukan hal berikut:
    a. Untuk tipe Rahasia, pilih Jenis rahasia lainnya.
    b. Masukkan rahasia Anda sebagai pasangan kunci/nilai atau di Plaintext. Beberapa contoh:

Pasangan kunci/nilai kunci API:

ClientID: ***my_client_id***

ClientSecret : ***bPxRfiWJALRXUTNFEMI/K7MDENG/EXAMPLEKEY***

Pasangan kunci/nilai kredensyal:

Username: *saanvis*

Password: *CONTOH-KATA SANDI*

Teks biasa token OAuth:

AKIAI44QH8DHBEXAMPLE

Sertifikat digital plaintext:

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

Plaintext kunci pribadi:

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

c. Untuk kunci Enkripsi, pilih aws/secretsmanager untuk menggunakan for Kunci yang dikelola AWS Secrets Manager. Tidak ada biaya untuk menggunakan kunci ini. Anda juga dapat menggunakan kunci yang dikelola pelanggan Anda sendiri, misalnya untuk mengakses rahasia dari yang lain Akun AWS. Untuk informasi tentang biaya penggunaan kunci yang dikelola pelanggan, lihatHarga.

d. Pilih Selanjutnya.

4. Pada halaman Pilih jenis rahasia, lakukan hal berikut:

a. Masukkan nama Rahasia deskriptif dan Deskripsi.

b. Di Izin sumber daya, pilih Edit izin. Tempel kebijakan berikut, yang memungkinkan *RoleToRetrieveSecretAtRuntime* untuk mengambil rahasia, lalu pilih Simpan.

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

c. Di bagian bawah halaman, pilih Selanjutnya.

5. Pada halaman Konfigurasi rotasi, matikan rotasi. Pilih Selanjutnya.
6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Langkah 2: Perbarui kode Anda

Kode Anda harus mengambil peran IAM *RoleToRetrieveSecretAtRuntime* untuk dapat mengambil rahasia. Untuk informasi selengkapnya, lihat Beralih ke peran IAM (AWSAPI).

Selanjutnya, Anda memperbarui kode Anda untuk mengambil rahasia dari Secrets Manager menggunakan kode contoh yang disediakan oleh Secrets Manager.

Untuk menemukan kode sampel

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pada halaman Rahasia, pilih rahasia Anda.
3. Gulir ke bawah ke kode Contoh. Pilih bahasa pemrograman Anda, lalu salin cuplikan kode.

Dalam aplikasi Anda, hapus rahasia hardcode dan tempel cuplikan kode. Bergantung pada bahasa kode Anda, Anda mungkin perlu menambahkan panggilan ke fungsi atau metode dalam cuplikan.

Uji apakah aplikasi Anda berfungsi seperti yang diharapkan dengan rahasia menggantikan rahasia hardcode.

Langkah 3: Perbarui rahasianya

Langkah terakhir adalah mencabut dan memperbarui rahasia hardcode. Lihat sumber rahasia untuk menemukan instruksi untuk mencabut dan memperbarui rahasia. Misalnya, Anda mungkin perlu menonaktifkan rahasia saat ini dan menghasilkan rahasia baru.

Untuk memperbarui rahasia dengan nilai baru

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Rahasia, lalu pilih rahasianya.
3. Pada halaman Detail rahasia, gulir ke bawah dan pilih Ambil nilai rahasia, lalu pilih Edit.
4. Perbarui rahasianya lalu pilih Simpan.

Selanjutnya, uji apakah aplikasi Anda berfungsi seperti yang diharapkan dengan rahasia baru.

Langkah selanjutnya

Setelah Anda menghapus rahasia hardcode dari kode Anda, beberapa ide untuk dipertimbangkan selanjutnya:

- Untuk menemukan rahasia hardcode di aplikasi Java dan Python Anda, kami merekomendasikan Amazon Reviewer. CodeGuru
- Anda dapat meningkatkan kinerja dan mengurangi biaya dengan menyimpan rahasia. Untuk informasi selengkapnya, lihat Ambil rahasia.
- Untuk rahasia yang Anda akses dari beberapa Wilayah, pertimbangkan untuk mereplikasi rahasia Anda untuk meningkatkan latensi. Untuk informasi selengkapnya, lihat the section called “Replikasi rahasia ke Wilayah lain”.
- Dalam tutorial ini, Anda *RoleToRetrieveSecretAtRuntime*hanya memberikan izin untuk mengambil nilai rahasia. Untuk memberikan peran lebih banyak izin, misalnya untuk mendapatkan metadata tentang rahasia atau untuk melihat daftar rahasia, lihat. the section called “Contoh kebijakan izin”
- Dalam tutorial ini, Anda diberikan izin untuk *RoleToRetrieveSecretAtRuntime*menggunakan kebijakan sumber daya rahasia. Untuk cara lain untuk memberikan izin, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

Pindahkan kredensyal basis data hardcode ke AWS Secrets Manager

Jika Anda memiliki kredensyal database plaintext dalam kode Anda, kami sarankan Anda memindahkan kredensialnya ke Secrets Manager dan kemudian segera memutarnya. Memindahkan kredensyal ke Secrets Manager memecahkan masalah kredensyal yang terlihat oleh siapa saja yang melihat kode, karena ke depan, kode Anda mengambil kredensyal langsung dari Secrets Manager. Memutar rahasia memperbarui kata sandi dan kemudian mencabut kata sandi hardcode saat ini sehingga tidak lagi valid.

Untuk database Amazon RDS, Amazon Redshift, dan Amazon DocumentDB, gunakan langkah-langkah di halaman ini untuk memindahkan kredensyal hardcode ke Secrets Manager. Untuk jenis kredensyal dan rahasia lainnya, lihat. the section called “Ganti rahasia hardcode ”

Sebelum Anda mulai, Anda perlu menentukan siapa yang membutuhkan akses ke rahasia. Sebaiknya gunakan dua peran IAM untuk mengelola izin rahasia Anda:

- Peran yang mengelola rahasia dalam organisasi Anda. Untuk informasi selengkapnya, lihat the section called “Izin administrator Secrets Manager”. Anda akan membuat dan memutar rahasia menggunakan peran ini.
- Peran yang dapat menggunakan kredensyal saat runtime, `RoleToRetrieveSecretAtRuntime`dalam tutorial ini. Kode Anda mengasumsikan peran ini untuk mengambil rahasia.

Langkah:

- Langkah 1: Buat Rahasianya
- Langkah 2: Perbarui kode Anda
- Langkah 3: Putar rahasianya
- Langkah selanjutnya

Langkah 1: Buat Rahasianya

Langkah pertama adalah menyalin kredensi hardcode yang ada menjadi rahasia di Secrets Manager. Untuk latensi terendah, simpan rahasia di Wilayah yang sama dengan database.

Untuk membuat rahasia

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih jenis rahasia, lakukan hal berikut:

   a. Untuk tipe Rahasia, pilih jenis kredensyal database yang akan disimpan:

      - Basis data Amazon RDS
      - Basis data Amazon DocumentDB
      - Gudang data Amazon Redshift.
      - Untuk jenis rahasia lainnya, lihat Mengganti rahasia hardcode.

   b. Untuk Credentials, masukkan kredensyal hardcode yang ada untuk database.

   c. Untuk kunci Enkripsi, pilih aws/secretsmanager untuk menggunakan for Kunci yang dikelola AWS Secrets Manager. Tidak ada biaya untuk menggunakan kunci ini. Anda juga dapat menggunakan kunci yang dikelola pelanggan Anda sendiri, misalnya untuk mengakses rahasia dari yang lain Akun AWS. Untuk informasi tentang biaya penggunaan kunci yang dikelola pelanggan, lihatHarga.

   d. Untuk Database, pilih database Anda.

   e. Pilih Berikutnya.

4. Pada halaman Konfigurasi rahasia, lakukan hal berikut:

   a. Masukkan nama Rahasia deskriptif dan Deskripsi.

   b. Di Izin sumber daya, pilih Edit izin. Tempel kebijakan berikut, yang memungkinkan *RoleToRetrieveSecretAtRuntime*untuk mengambil rahasia, lalu pilih Simpan.

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

```
}
```

c. Di bagian bawah halaman, pilih Selanjutnya.

5. Pada halaman Konfigurasi rotasi, matikan rotasi untuk saat ini. Anda akan menyalakannya nanti. Pilih Berikutnya.
6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Langkah 2: Perbarui kode Anda

Kode Anda harus mengambil peran IAM *RoleToRetrieveSecretAtRuntime*untuk dapat mengambil rahasia. Untuk informasi selengkapnya, lihat Beralih ke peran IAM (AWS API).

Selanjutnya, Anda memperbarui kode Anda untuk mengambil rahasia dari Secrets Manager menggunakan kode contoh yang disediakan oleh Secrets Manager.

Untuk menemukan kode sampel

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pada halaman Rahasia, pilih rahasia Anda.
3. Gulir ke bawah ke kode Contoh. Pilih bahasa Anda, lalu salin cuplikan kode.

Dalam aplikasi Anda, hapus kredensi hardcode dan tempel cuplikan kode. Bergantung pada bahasa kode Anda, Anda mungkin perlu menambahkan panggilan ke fungsi atau metode dalam cuplikan.

Uji apakah aplikasi Anda berfungsi seperti yang diharapkan dengan rahasia menggantikan kredensi hardcode.

Langkah 3: Putar rahasianya

Langkah terakhir adalah mencabut kredensi hardcode dengan memutar rahasianya. Rotasi adalah proses memperbarui rahasia secara berkala. Ketika Anda memutar rahasia, Anda memperbarui kredensyal di kedua rahasia dan database. Secrets Manager dapat secara otomatis memutar rahasia untuk Anda pada jadwal yang Anda tetapkan.

Bagian dari pengaturan rotasi adalah memastikan bahwa fungsi rotasi Lambda dapat mengakses Secrets Manager dan database Anda. Saat Anda mengaktifkan rotasi otomatis, Secrets Manager membuat fungsi rotasi Lambda di VPC yang sama dengan database Anda sehingga memiliki akses

jaringan ke database. Fungsi rotasi Lambda juga harus dapat melakukan panggilan ke Secrets Manager untuk memperbarui rahasia. Kami menyarankan Anda membuat titik akhir Secrets Manager di VPC sehingga panggilan dari Lambda ke Secrets Manager tidak meninggalkan infrastruktur. AWS Untuk petunjuk, lihat Titik akhir VPC.

Untuk mengaktifkan rotasi

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pada halaman Rahasia, pilih rahasia Anda.
3. Pada halaman Detail rahasia, di bagian konfigurasi Rotasi, pilih Edit rotasi.
4. Dalam kotak dialog Edit konfigurasi rotasi, lakukan hal berikut:
    a. Nyalakan Rotasi otomatis.
    b. Di bawah jadwal Rotasi, masukkan jadwal Anda di zona waktu UTC.
    c. Pilih Putar segera ketika rahasia disimpan untuk memutar rahasia Anda ketika Anda menyimpan perubahan Anda.
    d. Di bawah fungsi Rotasi, pilih Buat fungsi Lambda baru dan masukkan nama untuk fungsi baru Anda. Secrets Manager menambahkan SecretsManager "" ke awal nama fungsi Anda.
    e. Untuk strategi Rotasi, pilih Single user.
    f. Pilih Simpan.

Untuk memeriksa bahwa rahasianya diputar

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Rahasia, lalu pilih rahasianya.
3. Pada halaman Detail rahasia, gulir ke bawah dan pilih Ambil nilai rahasia.

    Jika nilai rahasia berubah, maka rotasi berhasil. Jika nilai rahasia tidak berubah, Anda perlu Memecahkan masalah rotasi melihat CloudWatch Log untuk fungsi rotasi.

Uji apakah aplikasi Anda berfungsi seperti yang diharapkan dengan rahasia yang diputar.

Langkah selanjutnya

Setelah Anda menghapus rahasia hardcode dari kode Anda, beberapa ide untuk dipertimbangkan selanjutnya:

- Anda dapat meningkatkan kinerja dan mengurangi biaya dengan menyimpan rahasia. Untuk informasi selengkapnya, lihat Ambil rahasia.
- Anda dapat memilih jadwal rotasi yang berbeda. Untuk informasi selengkapnya, lihat the section called “Ekspresi jadwal”.
- Untuk menemukan rahasia hardcode di aplikasi Java dan Python Anda, kami merekomendasikan Amazon Reviewer. CodeGuru

Siapkan rotasi pengguna bergantian untuk AWS Secrets Manager

Dalam tutorial ini, Anda belajar cara mengatur rotasi pengguna bergantian untuk rahasia yang berisi kredensi database. Rotasi pengguna bergantian adalah strategi rotasi di mana Secrets Manager mengkloning pengguna dan kemudian mengganti kredensi pengguna mana yang diperbarui. Strategi ini adalah pilihan yang baik jika Anda membutuhkan ketersediaan tinggi untuk rahasia Anda, karena salah satu pengguna bolak-balik memiliki kredensi saat ini ke database sementara yang lain sedang diperbarui. Untuk informasi selengkapnya, lihat the section called “Pengguna bergantian”.

Untuk mengatur rotasi pengguna bergantian, Anda memerlukan dua rahasia:

- Satu rahasia dengan kredenasia yang ingin Anda rotasikan.
- Rahasia kedua yang memiliki kredensi admin.

  Pengguna ini memiliki izin untuk mengkloning pengguna pertama dan mengubah kata sandi pengguna pertama. Dalam tutorial ini, Anda memiliki Amazon RDS membuat rahasia ini untuk pengguna admin. Amazon RDS juga mengelola rotasi kata sandi admin. Untuk informasi selengkapnya, lihat the section called “Rotasi terkelola”.

Bagian pertama dari tutorial ini adalah menyiapkan lingkungan yang realistis. Untuk menunjukkan cara kerja rotasi, tutorial ini menggunakan contoh database Amazon RDS MySQL. Untuk keamanan, database berada dalam VPC yang membatasi akses internet masuk. Untuk terhubung ke database dari komputer lokal Anda melalui internet, Anda menggunakan bastion host, server di VPC yang dapat terhubung ke database, tetapi itu juga memungkinkan koneksi SSH dari internet. Host bastion dalam tutorial ini adalah instans Amazon EC2, dan grup keamanan untuk instance mencegah jenis koneksi lainnya.

Setelah Anda menyelesaikan tutorial, kami sarankan Anda membersihkan sumber daya dari tutorial. Jangan menggunakannya dalam pengaturan produksi.

Rotasi Secrets Manager menggunakan AWS Lambda fungsi untuk memperbarui rahasia dan database. Untuk informasi tentang biaya penggunaan fungsi Lambda, lihat. Harga

Tutorial:

- Izin
- Prasyarat
- Langkah 1: Buat pengguna basis data Amazon RDS
- Langkah 2: Buat rahasia untuk kredenasia pengguna
- Langkah 3: Uji rahasia yang diputar
- Langkah 4: Bersihkan Sumber Daya
- Langkah selanjutnya

Izin

Untuk prasyarat tutorial, Anda memerlukan izin administratif untuk Anda. Akun AWS Dalam pengaturan produksi, ini adalah praktik terbaik untuk menggunakan peran yang berbeda untuk setiap langkah. Misalnya, peran dengan izin admin database akan membuat database Amazon RDS, dan peran dengan izin admin jaringan akan mengatur VPC dan grup keamanan. Untuk langkah-langkah tutorial, kami sarankan Anda terus menggunakan identitas yang sama.

Untuk informasi tentang cara menyiapkan izin di lingkungan produksi, lihatKontrol autentikasi dan akses.

Prasyarat

Dalam tutorial ini, Anda akan melakukan langkah-langkah berikut:

- Prasyarat A: Amazon VPC
- Prasyarat B: Instans Amazon EC2
- Prereq C: Basis data Amazon RDS dan rahasia Secrets Manager untuk kredensi admin
- Prereq D: Izinkan komputer lokal Anda terhubung ke instans EC2

Prasyarat A: Amazon VPC

Di langkah ini, Anda membuat VPC tempat Anda dapat meluncurkan basis data Amazon RDS dan instans Amazon EC2. Pada langkah selanjutnya, Anda akan menggunakan komputer Anda untuk

terhubung melalui internet ke benteng dan kemudian ke database, jadi Anda perlu mengizinkan lalu lintas keluar dari VPC. Untuk melakukan ini, Amazon VPC melampirkan gateway internet ke VPC dan menambahkan rute di tabel rute sehingga lalu lintas yang ditujukan untuk di luar VPC dikirim ke gateway internet.

Di dalam VPC, Anda membuat titik akhir Secrets Manager dan endpoint Amazon RDS. Saat Anda mengatur rotasi otomatis di langkah selanjutnya, Secrets Manager membuat fungsi rotasi Lambda di dalam VPC sehingga dapat mengakses database. Fungsi rotasi Lambda juga memanggil Secrets Manager untuk memperbarui rahasia, dan memanggil Amazon RDS untuk mendapatkan informasi koneksi database. Dengan membuat titik akhir dalam VPC, Anda memastikan bahwa panggilan dari fungsi Lambda ke Secrets Manager dan Amazon RDS tidak meninggalkan infrastruktur. AWS Sebaliknya, mereka diarahkan ke titik akhir dalam VPC.

Untuk membuat VPC

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Pilih Buat VPC.
3. Pada halaman Buat VPC, pilih VPC dan lainnya.
4. Di bawah Generasi otomatis tag nama, di bawah Generasi otomatis, masukkan. **`SecretsManagerTutorial`**
5. Untuk opsi DNS, pilih keduanya **`Enable DNS hostnames`** dan**`Enable DNS resolution`**.
6. Pilih Buat VPC.

Untuk membuat titik akhir Secrets Manager di dalam VPC

1. Di konsol Amazon VPC, di bawah Endpoints, pilih Create Endpoint.
2. Di bawah Pengaturan titik akhir, untuk Nama, masukkan**`SecretsManagerTutorialEndpoint`**.
3. Di bawah Layanan, masukkan **`secretsmanager`** untuk memfilter daftar, lalu pilih titik akhir Secrets Manager di bagian AndaWilayah AWS. Misalnya, di US East (N. Virginia), pilih`com.amazonaws.us-east-1.secretsmanager`.
4. Untuk VPC, pilih. **`vpc**** (SecretsManagerTutorial)`**
5. Untuk Subnet, pilih semua Availability Zone, dan kemudian untuk masing-masing Subnet, pilih Subnet ID untuk disertakan.
6. Untuk jenis alamat IP, pilih**`IPv4`**.

7. Untuk grup Keamanan, pilih grup keamanan default.
8. Untuk Kebijakan, pilih**`Full access`**.
9. Pilih Buat Titik Akhir.

Untuk membuat titik akhir Amazon RDS dalam VPC

1. Di konsol Amazon VPC, di bawah Endpoints, pilih Create Endpoint.
2. Di bawah Pengaturan titik akhir, untuk Nama, masukkan**`RDSTutorialEndpoint`**.
3. Di bawah Layanan, masukkan **`rds`** untuk memfilter daftar, lalu pilih titik akhir Amazon RDS di Anda. Wilayah AWS Misalnya, di US East (N. Virginia), pilih`com.amazonaws.us-east-1.rds`.
4. Untuk VPC, pilih. **`vpc**** (SecretsManagerTutorial)`**
5. Untuk Subnet, pilih semua Availability Zone, dan kemudian untuk masing-masing Subnet, pilih Subnet ID untuk disertakan.
6. Untuk jenis alamat IP, pilih**`IPv4`**.
7. Untuk grup Keamanan, pilih grup keamanan default.
8. Untuk Kebijakan, pilih**`Full access`**.
9. Pilih Buat Titik Akhir.

Prasyarat B: Instans Amazon EC2

Basis data Amazon RDS yang Anda buat di langkah selanjutnya akan berada di VPC, jadi untuk mengaksesnya, Anda memerlukan host benteng. Host bastion juga ada di VPC, tetapi pada langkah selanjutnya, Anda mengonfigurasi grup keamanan untuk memungkinkan komputer lokal Anda terhubung ke host bastion dengan SSH.

Untuk membuat instans EC2 untuk host bastion

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pilih Instans dan kemudian pilih Launch Instances.
3. Di bawah Nama dan tag, untuk Nama, masukkan**`SecretsManagerTutorialInstance`**.
4. Di bawah Application dan OS Images, pertahankan default**`Amazon Linux 2 AMI (HMV) Kernel 5.10`**.
5. Di bawah tipe Instance, pertahankan default**`t2.micro`**.
6. Di bawah Key pair, pilih Create key pair.

Dalam kotak dialog Create key pair, untuk nama Key pair, masukkan**SecretsManagerTutorialKeyPair**, lalu pilih Create key pair.

Key pair secara otomatis diunduh.

7. Di bawah Pengaturan jaringan, pilih Edit, lalu lakukan hal berikut:
   a. Untuk VPC, pilih. **vpc-**** SecretsManagerTutorial**
   b. Untuk Tetapkan IP Publik secara Otomatis, pilih. **Enable**
   c. Untuk Firewall, pilih Pilih grup keamanan yang ada.
   d. Untuk grup keamanan umum, pilih**default**.
8. Pilih Luncurkan instans.

Prereq C: Basis data Amazon RDS dan rahasia Secrets Manager untuk kredensi admin

Pada langkah ini, Anda membuat basis data Amazon RDS MySQL dan mengonfigurasinya sehingga Amazon RDS membuat rahasia untuk berisi kredenasia admin. Kemudian Amazon RDS secara otomatis mengelola rotasi rahasia admin untuk Anda. Untuk informasi selengkapnya, lihat Rotasi terkelola.

Sebagai bagian dari membuat basis data Anda, Anda menentukan host bastion yang Anda buat di langkah sebelumnya. Kemudian Amazon RDS menyiapkan grup keamanan sehingga database dan instans dapat saling mengakses. Anda menambahkan aturan ke grup keamanan yang dilampirkan ke instance untuk memungkinkan komputer lokal Anda terhubung dengannya juga.

Untuk membuat database Amazon RDS dengan rahasia Secrets Manager yang berisi kredensi admin

1. Di konsol Amazon RDS, pilih Buat basis data.
2. Di bagian Opsi mesin, untuk jenis mesin, pilih**MySQL**.
3. Di bagian Template, pilih**Free tier**.
4. Di bagian Pengaturan, lakukan hal berikut:
   a. Untuk Pengidentifikasi instans DB, memasukkan **SecretsManagerTutorial**.
   b. Di bawah Pengaturan kredensi, pilih Kelola kredenal master di. AWS Secrets Manager
5. Di bagian Konektivitas, untuk sumber daya Komputer, pilih Connect ke sumber daya komputer EC2, dan kemudian untuk Instans EC2, pilih. **SecretsManagerTutorialInstance**

6. Pilih Buat basis data.

Prereq D: Izinkan komputer lokal Anda terhubung ke instans EC2

Pada langkah ini, Anda mengonfigurasi instans EC2 yang Anda buat di Prereq B untuk memungkinkan komputer lokal Anda terhubung dengannya. Untuk melakukan ini, Anda mengedit grup keamanan yang ditambahkan Amazon RDS di Prereq C untuk menyertakan aturan yang memungkinkan alamat IP komputer Anda terhubung dengan SSH. Aturan ini memungkinkan komputer lokal Anda (diidentifikasi oleh alamat IP Anda saat ini) untuk terhubung ke host bastion dengan menggunakan SSH melalui internet.

Untuk memungkinkan komputer lokal Anda terhubung ke instans EC2

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada instans EC2 SecretsManagerTutorialInstance, pada tab Keamanan, di bawah Grup keamanan, pilih**`sg-*** (ec2-rds-X)`**.
3. Di bawah Aturan input, pilih Edit aturan masuk.
4. Pilih Tambah aturan, dan kemudian untuk aturan, lakukan hal berikut:
   a. Untuk Jenis, pilih **`SSH`**.
   b. Untuk tipe Sumber, pilih**`My IP`**.

Langkah 1: Buat pengguna basis data Amazon RDS

Pertama, Anda memerlukan pengguna yang kredensialnya akan disimpan dalam rahasia. Untuk membuat pengguna, masuk ke database Amazon RDS dengan kredensi admin. Untuk kesederhanaan, dalam tutorial, Anda membuat pengguna dengan izin penuh ke database. Dalam pengaturan produksi, ini tidak khas, dan kami menyarankan Anda mengikuti prinsip hak istimewa paling sedikit.

Untuk terhubung ke database, Anda menggunakan alat klien MySQL. Dalam tutorial ini, Anda menggunakan MySQL Workbench, aplikasi berbasis GUI. Untuk menginstal MySQL Workbench, lihat Download MySQL Workbench.

Untuk terhubung ke database, buat konfigurasi koneksi di MySQL Workbench. Untuk konfigurasi, Anda memerlukan beberapa informasi dari Amazon EC2 dan Amazon RDS.

Untuk membuat koneksi database di MySQL Workbench

1. Di MySQL Workbench, di sebelah MySQL Connections, pilih tombol (+).
2. Di kotak dialog Setup New Connection, lakukan hal berikut:
   a. Untuk Nama Koneksi, masukkan**`SecretsManagerTutorial`**.
   b. Untuk Metode Koneksi, pilih**`Standard TCP/IP over SSH`**.
   c. Pada tab Parameter, lakukan hal berikut:
      i. Untuk SSH Hostname, masukkan alamat IP publik instans Amazon EC2.

         Anda dapat menemukan alamat IP di konsol Amazon EC2 dengan memilih instans. SecretsManagerTutorialInstance Salin alamat IP di bawah DNS IPv4 Publik.
      ii. Untuk Nama Pengguna SSH, masukkan**`ec2-user`**.
      iii. Untuk SSH Keyfile, pilih file key pair SecretsManagerTutorialKeyPair.pem yang Anda download di prasyarat sebelumnya.
      iv. Untuk MySQL Hostname, masukkan alamat endpoint Amazon RDS.

         Anda dapat menemukan alamat titik akhir di konsol Amazon RDS dengan memilih instans basis data secretsmanagertutorialdb. Salin alamat di bawah Endpoint.
      v. Untuk Nama Pengguna, masukkan**`admin`**.
   d. Pilih OKE.

Untuk mengambil kata sandi admin

1. Di konsol Amazon RDS, arahkan ke basis data Anda.
2. Pada tab Konfigurasi, di bawah Master Credentials ARN, pilih Manage in Secrets Manager.

   Konsol Secrets Manager terbuka.
3. Di halaman detail rahasia, pilih Ambil nilai rahasia.
4. Kata sandi muncul di bagian Nilai rahasia.

Untuk membuat pengguna basis data

1. Di MySQL Workbench, pilih koneksi. SecretsManagerTutorial
2. Masukkan kata sandi admin yang Anda ambil dari rahasia.

3. Di MySQL Workbench, di jendela Query, masukkan perintah berikut (termasuk kata sandi yang kuat) dan kemudian pilih Execute.

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

   Di jendela Output, Anda melihat perintah berhasil.

Langkah 2: Buat rahasia untuk kredenasia pengguna

Selanjutnya, Anda membuat rahasia untuk menyimpan kredenasia pengguna yang baru saja Anda buat. Ini adalah rahasia yang akan Anda putar. Anda mengaktifkan rotasi otomatis, dan untuk menunjukkan strategi pengguna bergantian, Anda memilih rahasia superuser terpisah yang memiliki izin untuk mengubah kata sandi pengguna pertama.

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Di halaman Pilih tipe rahasia, lakukan hal berikut:
   a. Untuk jenis Rahasia, pilih Credentials for Amazon RDS database.
   b. Untuk Kredensial, masukkan nama pengguna **`appuser`** dan kata sandi yang Anda masukkan untuk pengguna database yang Anda buat menggunakan MySQL Workbench.
   c. Untuk Database, pilih secretsmanagertutorialdb.
   d. Pilih Selanjutnya.
4. Pada halaman Konfigurasi rahasia, untuk nama Rahasia, masukkan **`SecretsManagerTutorialAppuser`** dan kemudian pilih Berikutnya.
5. Pada halaman Konfigurasikan rotasi, lakukan hal berikut:
   a. Nyalakan Rotasi otomatis.
   b. Untuk jadwal Rotasi, atur jadwal Hari: **2** Hari dengan Durasi:**2h**. Tetap Putar segera dipilih.
   c. Untuk fungsi Rotasi, pilih Buat fungsi rotasi, dan kemudian untuk nama fungsi, masukkan**`tutorial-alternating-users-rotation`**.
   d. Untuk strategi Rotasi, pilih Alternating users, dan kemudian di bawah Admin credential secret, pilih rahasia bernama rds! kluster... yang memiliki Deskripsi yang menyertakan nama database yang Anda buat dalam tutorial ini**`secretsmanagertutorial`**,

misalnya`Secret associated with primary RDS DB instance: arn:aws:rds:`*`Region`*`:`*`AccountId`*`:db:secretsmanagertutorial`.

e. Pilih Selanjutnya.

6. Pada halaman Review, pilih Store.

   Secrets Manager kembali ke halaman detail rahasia. Di bagian atas halaman, Anda dapat melihat status konfigurasi rotasi. Secrets Manager menggunakan CloudFormation untuk membuat sumber daya seperti fungsi rotasi Lambda dan peran eksekusi yang menjalankan fungsi Lambda. Setelah CloudFormation selesai, spanduk berubah menjadi Rahasia yang dijadwalkan untuk rotasi. Rotasi pertama selesai.

Langkah 3: Uji rahasia yang diputar

Sekarang rahasianya diputar, Anda dapat memeriksa apakah rahasia tersebut berisi kredensil baru yang valid. Kata sandi dalam rahasia telah berubah dari kredensi asli.

Untuk mengambil kata sandi baru dari rahasia

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Rahasia, lalu pilih rahasianya**`SecretsManagerTutorialAppuser`**.
3. Pada halaman Detail rahasia, gulir ke bawah dan pilih Ambil nilai rahasia.
4. Dalam tabel kunci/Nilai, salin nilai Rahasia untuk. **`password`**

Untuk menguji kredensialnya

1. Di MySQL Workbench, klik kanan koneksi dan kemudian pilih Edit SecretsManagerTutorialKoneksi.
2. Dalam kotak dialog Kelola Koneksi Server, untuk Nama Pengguna**`appuser`**, masukkan, lalu pilih Tutup.
3. Kembali di MySQL Workbench, pilih koneksi. SecretsManagerTutorial
4. Dalam Buka Koneksi SSH kotak dialog, untuk Kata Sandi, tempel kata sandi yang Anda ambil dari rahasia, lalu pilih OK.

   Jika kredensialnya valid, maka MySQL Workbench terbuka ke halaman desain untuk database.

Ini menunjukkan bahwa rotasi rahasia berhasil. Kredensi dalam rahasia telah diperbarui dan itu adalah kata sandi yang valid untuk terhubung ke database.

Langkah 4: Bersihkan Sumber Daya

Jika Anda ingin mencoba strategi rotasi lain, rotasi pengguna tunggal, lewati pembersihan sumber daya dan bukathe section called “Rotasi pengguna tunggal”.

Jika tidak, untuk menghindari potensi biaya, dan untuk menghapus instans EC2 yang memiliki akses ke internet, hapus sumber daya berikut yang Anda buat dalam tutorial ini dan prasyaratnya:

- Instans basis data Amazon RDS. Untuk petunjuknya, lihat Menghapus instans DB di Panduan Pengguna Amazon RDS.
- Instans Amazon EC2. Untuk petunjuknya, lihat Mengakhiri instance di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Rahasia Secrets Manager`SecretsManagerTutorialAppuser`. Untuk petunjuk, lihat the section called “Hapus rahasia”.
- Titik akhir Secrets Manager. Untuk petunjuk, lihat Menghapus titik akhir VPC di Panduan. AWS PrivateLink
- Titik akhir VPC. Untuk petunjuk, lihat Menghapus VPC Anda di Panduan. AWS PrivateLink

Langkah selanjutnya

- Pelajari cara mengambil rahasia di aplikasi Anda.
- Pelajari tentang jadwal rotasi lainnya.

Siapkan rotasi pengguna tunggal untuk AWS Secrets Manager

Dalam tutorial ini, Anda belajar cara mengatur rotasi pengguna tunggal untuk rahasia yang berisi kredensi database. Rotasi pengguna tunggal adalah strategi rotasi di mana Secrets Manager memperbarui kredensi pengguna baik dalam rahasia maupun database. Untuk informasi selengkapnya, lihat the section called “Pengguna tunggal”.

Setelah Anda menyelesaikan tutorial, kami sarankan Anda membersihkan sumber daya dari tutorial. Jangan menggunakannya dalam pengaturan produksi.

Rotasi Secrets Manager menggunakan AWS Lambda fungsi untuk memperbarui rahasia dan database. Untuk informasi tentang biaya penggunaan fungsi Lambda, lihat. Harga

Daftar Isi

Izin

Untuk prasyarat tutorial, Anda memerlukan izin administratif untuk Anda. Akun AWS Dalam pengaturan produksi, ini adalah praktik terbaik untuk menggunakan peran yang berbeda untuk setiap langkah. Misalnya, peran dengan izin admin database akan membuat database Amazon RDS, dan peran dengan izin admin jaringan akan mengatur VPC dan grup keamanan. Untuk langkah-langkah tutorial, kami sarankan Anda terus menggunakan identitas yang sama.

Untuk informasi tentang cara mengatur izin di lingkungan produksi, lihatKontrol autentikasi dan akses.

Prasyarat

Prasyarat untuk tutorial ini adalah. the section called “Rotasi pengguna secara bergantian” Jangan membersihkan sumber daya di akhir tutorial pertama. Setelah tutorial itu, Anda memiliki lingkungan yang realistis dengan database Amazon RDS dan rahasia Secrets Manager yang berisi kredensi admin untuk database. Anda juga memiliki rahasia kedua yang berisi kredensil untuk pengguna database, tetapi Anda tidak menggunakan rahasia itu dalam tutorial ini.

Anda juga memiliki koneksi yang dikonfigurasi di MySQL Workbench untuk terhubung ke database dengan kredensi admin.

Langkah 1: Membuat basis data Amazon RDS

Pertama, Anda memerlukan pengguna yang kredensialnya akan disimpan dalam rahasia. Untuk membuat pengguna, masuk ke database Amazon RDS dengan kredensi admin yang disimpan dalam

rahasia. Untuk kesederhanaan, dalam tutorial, Anda membuat pengguna dengan izin penuh ke database. Dalam pengaturan produksi, ini tidak khas, dan kami menyarankan Anda mengikuti prinsip hak istimewa paling sedikit.

Untuk mengambil kata sandi admin

1. Di konsol Amazon RDS, navigasikan ke basis data.
2. Pada tab Konfigurasi, di bawah Master Credentials ARN, pilih Manage in Secrets Manager.

   Konsol Secrets Manager terbuka.
3. Di halaman detail rahasia, pilih Ambil nilai rahasia.
4. Kata sandi muncul di bagian Nilai rahasia.

Untuk membuat pengguna basis data

1. Di MySQL Workbench, klik kanan koneksi dan kemudian pilih Edit SecretsManagerTutorialKoneksi.
2. Dalam kotak dialog Kelola Koneksi Server, untuk Nama Pengguna**admin**, masukkan, lalu pilih Tutup.
3. Kembali di MySQL Workbench, pilih koneksi. SecretsManagerTutorial
4. Masukkan kata sandi admin yang Anda ambil dari rahasia.
5. Di MySQL Workbench, di jendela Query, masukkan perintah berikut (termasuk kata sandi yang kuat) dan kemudian pilih Execute.

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

   Di jendela Output, Anda melihat perintah berhasil.

Langkah 2: Membuat rahasia untuk kredensi basis data

Selanjutnya, Anda membuat rahasia untuk menyimpan kredensil pengguna yang baru saja Anda buat, dan Anda mengaktifkan rotasi otomatis, termasuk rotasi langsung. Secrets Manager memutar rahasia, yang berarti kata sandi dihasilkan secara terprogram - tidak ada manusia yang melihat kata sandi baru ini. Memulai rotasi segera juga dapat membantu Anda menentukan apakah rotasi diatur dengan benar.

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih tipe rahasia, lakukan hal berikut:
   a. Untuk jenis Rahasia, pilih Credentials for Amazon RDS database.
   b. Untuk Kredensial, masukkan nama pengguna **`dbuser`** dan kata sandi yang Anda masukkan untuk pengguna database yang Anda buat menggunakan MySQL Workbench.
   c. Untuk Database, pilih secretsmanagertutorialdb.
   d. Pilih Selanjutnya.
4. Pada halaman Konfigurasi rahasia, untuk nama Rahasia, masukkan **`SecretsManagerTutorialDbuser`** dan kemudian pilih Berikutnya.
5. Pada halaman Konfigurasi rotasi, lakukan hal berikut:
   a. Nyalakan Rotasi otomatis.
   b. Untuk jadwal Rotasi, atur jadwal Hari: **2** Hari dengan Durasi:**2h**. Tetap Putar segera dipilih.
   c. Untuk fungsi Rotasi, pilih Buat fungsi rotasi, dan kemudian untuk nama fungsi, masukkan**`tutorial-single-user-rotation`**.
   d. Untuk strategi Rotasi, pilih Single user.
   e. Pilih Selanjutnya.
6. Pada halaman Review, pilih Store.

   Secrets Manager kembali ke halaman detail rahasia. Di bagian atas halaman, Anda dapat melihat status konfigurasi rotasi. Secrets Manager menggunakan CloudFormation untuk membuat sumber daya seperti fungsi rotasi Lambda dan peran eksekusi yang menjalankan fungsi Lambda. Setelah CloudFormation selesai, spanduk berubah menjadi Rahasia yang dijadwalkan untuk rotasi. Rotasi pertama selesai.

Langkah 3: Uji Kata sandi yang diputar

Setelah rotasi rahasia pertama, yang mungkin memakan waktu beberapa detik, Anda dapat memeriksa bahwa rahasia tersebut masih berisi kredensil yang valid. Kata sandi dalam rahasia telah berubah dari kredensi asli.

Untuk mengambil kata sandi baru dari rahasia

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.

2. Pilih Rahasia, lalu pilih rahasianya**`SecretsManagerTutorialDbuser`**.
3. Pada halaman Detail rahasia, gulir ke bawah dan pilih Ambil nilai rahasia.
4. Dalam tabel kunci/Nilai, salin nilai Rahasia untuk. **`password`**

Untuk menguji kredensialnya

1. Di MySQL Workbench, klik kanan koneksi dan kemudian pilih Edit SecretsManagerTutorialKoneksi.
2. Dalam kotak dialog Kelola Koneksi Server, untuk Nama Penggguna**`dbuser`**, masukkan, lalu pilih Tutup.
3. Kembali di MySQL Workbench, pilih koneksi. SecretsManagerTutorial
4. Di kotak dialog Open SSH Connection, untuk Kata Sandi, tempel kata sandi yang Anda ambil dari rahasia, lalu pilih OK.

   Jika kredensialnya valid, maka MySQL Workbench terbuka ke halaman desain untuk database.

Langkah 4: Bersihkan Sumber Daya

Untuk menghindari potensi biaya, hapus rahasia yang Anda buat dalam tutorial ini. Untuk petunjuk, lihat the section called “Hapus rahasia”.

Untuk membersihkan sumber daya yang dibuat dalam tutorial sebelumnya, lihatthe section called “Langkah 4: Bersihkan Sumber Daya”.

Langkah selanjutnya

- Pelajari cara mengambil rahasia di aplikasi Anda. Lihat Ambil rahasia.
- Pelajari tentang jadwal rotasi lainnya. Lihat the section called “Ekspresi jadwal”.

Kontrol autentikasi dan akses untuk AWS Secrets Manager

Secrets Manager menggunakan AWS Identity and Access Management(IAM) untuk mengamankan akses ke rahasia. IAM menyediakan otentikasi dan kontrol akses. Otentikasi memverifikasi identitas permintaan individu. Secrets Manager menggunakan proses masuk dengan kata sandi, kunci akses, dan token otentikasi multi-faktor (MFA) untuk memverifikasi identitas pengguna. Lihat Masuk ke AWS. Kontrol akses memastikan bahwa hanya individu yang disetujui yang dapat melakukan operasi pada AWS sumber daya seperti rahasia. Secrets Manager menggunakan kebijakan untuk menentukan siapa yang memiliki akses ke sumber daya mana, dan tindakan apa yang dapat diambil identitas terhadap sumber daya tersebut. Lihat Kebijakan dan izin di IAM.

Anda dapat menggunakan AWS Identity and Access Management Roles Anywhere untuk mendapatkan kredensil keamanan sementara di IAM untuk beban kerja seperti server, kontainer, dan aplikasi yang berjalan di luar. AWS Beban kerja Anda dapat menggunakan kebijakan IAM dan peran IAM yang sama yang Anda gunakan dengan AWS aplikasi untuk mengakses sumber daya. AWS Dengan IAM Roles Anywhere, Anda dapat menggunakan Secrets Manager untuk menyimpan dan mengelola kredensil yang dapat diakses oleh sumber daya di AWS serta perangkat lokal seperti server aplikasi. Untuk informasi selengkapnya, lihat Panduan Pengguna IAM Roles Anywhere.

Izin administrator Secrets Manager

Untuk memberikan izin administrator Secrets Manager, ikuti petunjuk di Menambahkan dan menghapus izin identitas IAM, dan lampirkan kebijakan berikut:

- SecretsManagerReadWrite
- IAMFullAccess

Kami menyarankan Anda untuk tidak memberikan izin administrator kepada pengguna akhir. Meskipun ini memungkinkan pengguna Anda untuk membuat dan mengelola rahasia mereka, izin yang diperlukan untuk mengaktifkan rotation (IAMFullAccess) memberikan izin signifikan yang tidak sesuai untuk pengguna akhir.

Izin untuk mengakses rahasia

Dengan menggunakan kebijakan izin IAM, Anda mengontrol pengguna atau layanan mana yang memiliki akses ke rahasia Anda. Kebijakan izin menjelaskan siapa yang dapat melakukan tindakan apa pada sumber daya mana. Anda dapat:

- the section called “Melampirkan kebijakan izin ke identitas”
- the section called “Lampirkan kebijakan izin ke rahasia”

Izin untuk fungsi rotasi Lambda

Secrets Manager menggunakan AWS Lambda fungsi untuk memutar rahasia. Fungsi Lambda harus memiliki akses ke rahasia serta database atau layanan yang rahasia berisi kredensialnya. Lihat Izin untuk rotasi.

Izin untuk kunci enkripsi

Secrets Manager menggunakan AWS Key Management Service (AWS KMS) kunci untuk mengenkripsi rahasia. Kunci yang dikelola AWS`aws/secretsmanager`Secara otomatis memiliki izin yang benar. Jika Anda menggunakan kunci KMS yang berbeda, Secrets Manager memerlukan izin untuk kunci tersebut. Lihat the section called “Izin untuk kunci KMS”.

Melampirkan kebijakan izin ke identitas

Anda dapat melampirkan kebijakan izin ke identitas IAM: pengguna, grup pengguna, dan peran. Dalam kebijakan berbasis identitas, Anda menentukan rahasia mana yang dapat diakses identitas dan tindakan yang dapat dilakukan identitas pada rahasia. Untuk informasi selengkapnya, lihat Menambahkan dan menghapus izin identitas IAM.

Anda dapat memberikan izin untuk peran yang mewakili aplikasi atau pengguna di layanan lain. Misalnya, aplikasi yang berjalan pada instans Amazon EC2 mungkin memerlukan akses ke database. Anda dapat membuat peran IAM yang dilampirkan ke profil instans EC2 dan kemudian menggunakan kebijakan izin untuk memberikan akses peran ke rahasia yang berisi kredensi untuk database. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2. Layanan lain yang dapat Anda lampirkan peran untuk menyertakan Amazon Redshift, AWS Lambda, dan Amazon ECS.

Anda juga dapat memberikan izin kepada pengguna yang diautentikasi oleh sistem identitas selain IAM. Misalnya, Anda dapat mengaitkan peran IAM ke pengguna aplikasi seluler yang masuk dengan Amazon Cognito. Peran tersebut memberikan kredensi sementara aplikasi dengan izin dalam kebijakan izin peran. Kemudian Anda dapat menggunakan kebijakan izin untuk memberikan akses peran ke rahasia. Untuk informasi selengkapnya, lihat Penyedia identitas dan federasi.

Anda dapat menggunakan kebijakan berbasis identitas untuk:

- Berikan akses identitas ke beberapa rahasia.
- Kontrol siapa yang dapat membuat rahasia baru, dan siapa yang dapat mengakses rahasia yang belum dibuat.
- Berikan akses grup IAM ke rahasia.

Untuk informasi selengkapnya, lihat the section called “Contoh kebijakan izin”.

Lampirkan kebijakan izin ke rahasia AWS Secrets Manager

Dalam kebijakan berbasis sumber daya, Anda menentukan siapa yang dapat mengakses rahasia dan tindakan yang dapat mereka lakukan pada rahasia tersebut. Anda dapat menggunakan kebijakan berbasis sumber daya untuk:

- Berikan akses ke satu rahasia ke beberapa pengguna dan peran.
- Berikan akses ke pengguna atau peran di AWS akun lain.

Lihat the section called “Contoh kebijakan izin”.

Saat Anda melampirkan kebijakan berbasis sumber daya ke rahasia di konsol, Secrets Manager menggunakan mesin penalaran otomatis Zelkova dan API `ValidateResourcePolicy` untuk mencegah Anda memberikan berbagai kepala sekolah IAM akses ke rahasia Anda. Atau, Anda dapat memanggil `PutResourcePolicy` API dengan `BlockPublicPolicy` parameter dari CLI atau SDK.

> **Important**
>
> Validasi kebijakan sumber daya dan `BlockPublicPolicy` parameter membantu melindungi sumber daya Anda dengan mencegah akses publik diberikan melalui kebijakan sumber daya yang secara langsung melekat pada rahasia Anda. Selain menggunakan fitur-fitur ini, periksa

dengan cermat kebijakan berikut untuk mengonfirmasi bahwa mereka tidak memberikan akses publik:

- Kebijakan berbasis identitas yang dilampirkan pada AWS prinsipal terkait (misalnya, peran IAM)
- Kebijakan berbasis sumber daya yang dilampirkan pada AWS sumber daya terkait (misalnya, AWS Key Management Service () kunci)AWS KMS

Untuk meninjau izin ke rahasia Anda, lihatTentukan siapa yang memiliki izin untuk rahasia Anda.

Untuk melihat, mengubah, atau menghapus kebijakan sumber daya untuk rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dari daftar rahasia, pilih rahasia Anda.
3. Pada halaman detail rahasia, pada tab Ikhtisar, di bagian Izin sumber daya, pilih Edit izin.
4. Di bidang kode, lakukan salah satu hal berikut, lalu pilih Simpan:
   - Untuk melampirkan atau mengubah kebijakan sumber daya, masukkan kebijakan.
   - Untuk menghapus kebijakan, kosongkan bidang kode.

AWS CLI

Example Mengambil kebijakan sumber daya

`get-resource-policy`Contoh berikut mengambil kebijakan berbasis sumber daya yang dilampirkan pada rahasia.

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

Example Menghapus kebijakan sumber daya

`delete-resource-policy`Contoh berikut menghapus kebijakan berbasis sumber daya yang dilampirkan pada rahasia.

```
aws secretsmanager delete-resource-policy \
```

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

Example Menambahkan kebijakan sumber daya

`put-resource-policy`Contoh berikut menambahkan kebijakan izin ke rahasia, memeriksa terlebih dahulu bahwa kebijakan tersebut tidak menyediakan akses luas ke rahasia tersebut. Kebijakan dibaca dari file. Untuk informasi selengkapnya, lihat Memuat AWS CLI parameter dari file di Panduan AWS CLI Pengguna.

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

Isi dari `mypolicy.json`:

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

AWS SDK

Untuk mengambil kebijakan yang dilampirkan pada rahasia, gunakan `GetResourcePolicy`.

Untuk menghapus kebijakan yang dilampirkan pada rahasia, gunakan `DeleteResourcePolicy`.

Untuk melampirkan kebijakan ke rahasia, gunakan `PutResourcePolicy`. Jika sudah ada kebijakan yang dilampirkan, perintah menggantinya dengan kebijakan baru. Kebijakan harus diformat sebagai teks terstruktur JSON. Lihat Struktur dokumen kebijakan JSON. Gunakan the section called “Contoh kebijakan izin” untuk mulai menulis kebijakan Anda.

Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

AWS kebijakan terkelola untuk AWS Secrets Manager

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pemutakhiran akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: SecretsManagerReadWrite

Kebijakan ini menyediakan akses baca/tulis AWS Secrets Manager, termasuk izin untuk menjelaskan sumber daya Amazon RDS, Amazon Redshift, dan Amazon DocumentDB, serta izin yang digunakan untuk mengenkripsi dan mendekripsi rahasia. AWS KMS Kebijakan ini juga memberikan izin untuk membuat set AWS CloudFormation perubahan, mendapatkan templat rotasi dari bucket Amazon S3 yang dikelola oleh AWS, mencantumkan AWS Lambda fungsi, dan menjelaskan VPC Amazon EC2. Izin ini diperlukan oleh konsol untuk mengatur rotasi dengan fungsi rotasi yang ada.

Untuk membuat fungsi rotasi baru, Anda juga harus memiliki izin untuk membuat AWS CloudFormation tumpukan dan peran AWS Lambda eksekusi. Anda dapat menetapkan kebijakan FullAccess terkelola IAM. Lihat Izin untuk rotasi.

Detail izin

Kebijakan ini mencakup izin berikut.

- `secretsmanager`— Memungkinkan kepala sekolah untuk melakukan semua tindakan Secrets Manager.
- `cloudformation`— Memungkinkan kepala sekolah untuk membuat tumpukan. AWS CloudFormation Ini diperlukan agar prinsipal yang menggunakan konsol untuk mengaktifkan rotasi dapat membuat fungsi rotasi Lambda melalui tumpukan. AWS CloudFormation Untuk informasi selengkapnya, lihat the section called “Bagaimana Secrets Manager menggunakan AWS CloudFormation”.
- `ec2`- Memungkinkan kepala sekolah untuk menggambarkan VPC Amazon EC2. Ini diperlukan agar prinsipal yang menggunakan konsol dapat membuat fungsi rotasi di VPC yang sama dengan database kredenal yang mereka simpan secara rahasia.
- `kms`— Memungkinkan kepala sekolah untuk menggunakan AWS KMS kunci untuk operasi kriptografi. Ini diperlukan agar Secrets Manager dapat mengenkripsi dan mendekripsi rahasia. Untuk informasi selengkapnya, lihat the section called “Enkripsi rahasia dan dekripsi”.
- `lambda`- Memungkinkan kepala sekolah untuk mencantumkan fungsi rotasi Lambda. Ini diperlukan agar prinsipal yang menggunakan konsol dapat memilih fungsi rotasi yang ada.
- `rds`— Memungkinkan kepala sekolah untuk menggambarkan cluster dan instance di Amazon RDS. Ini diperlukan agar prinsipal yang menggunakan konsol dapat memilih cluster atau instance Amazon RDS.
- `redshift`— Memungkinkan kepala sekolah untuk menggambarkan cluster di Amazon Redshift. Ini diperlukan agar prinsipal yang menggunakan konsol dapat memilih cluster Amazon Redshift.
- `redshift-serverless`— Memungkinkan kepala sekolah untuk mendeskripsikan ruang nama di Amazon Redshift Tanpa Server. Ini diperlukan agar prinsipal yang menggunakan konsol dapat memilih ruang nama Amazon Redshift Tanpa Server.
- `docdb-elastic`— Memungkinkan prinsipal untuk menggambarkan cluster elastis di Amazon DocumentDB. Ini diperlukan agar prinsipal yang menggunakan konsol dapat memilih cluster elastis Amazon DocumentDB.
- `tag`— Memungkinkan kepala sekolah untuk mendapatkan semua sumber daya di akun yang diberi tag.
- `serverlessrepo`— Memungkinkan kepala sekolah untuk membuat AWS CloudFormation set perubahan. Ini diperlukan agar prinsipal yang menggunakan konsol dapat membuat fungsi rotasi Lambda. Untuk informasi selengkapnya, lihat the section called “Bagaimana Secrets Manager menggunakan AWS CloudFormation”.
- `s3`— Memungkinkan prinsipal untuk mendapatkan objek dari bucket Amazon S3 yang dikelola oleh. AWS Ember ini berisi LambdaTemplat fungsi rotasi. Izin ini diperlukan agar prinsipal yang

menggunakan konsol dapat membuat fungsi rotasi Lambda berdasarkan templat di bucket. Untuk informasi selengkapnya, lihat the section called “Bagaimana Secrets Manager menggunakan AWS CloudFormation”.

Untuk melihat kebijakan, lihat dokumen kebijakan SecretsManagerReadWrite JSON.

Secrets Manager memperbarui kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Secrets Manager.

Perubahan	Deskripsi	Tanggal
SecretsManagerReadWrite — Permbaruan ke kebijakan yang sudah ada	Kebijakan ini diperbarui untuk mengizinkan akses deskripsi kan ke Amazon Redshift Tanpa Server sehingga pengguna konsol dapat memilih namespace Amazon Redshift Tanpa Server saat mereka membuat rahasia Amazon Redshift.	Maret 12, 2024
SecretsManagerReadWrite – Pembaruan ke kebijakan yang ada	Kebijakan ini diperbarui untuk memungkinkan akses deskripsikan ke klaster elastis Amazon DocumentD B sehingga pengguna konsol dapat memilih klaster elastis saat mereka membuat rahasia Amazon DocumentDB.	12 September 2023
SecretsManagerReadWrite – Pembaruan ke kebijakan yang ada	Kebijakan ini diperbarui untuk mengizinkan akses deskripsikan ke Amazon Redshift sehingga pengguna konsol dapat memilih klaster Amazon Redshift saat mereka membuat rahasia Amazon	24 Juni 2020

Perubahan	Deskripsi	Tanggal
	Redshift. Pembaruan juga menambahkan izin baru untuk memungkinkan akses baca ke bucket Amazon S3 yang dikelola AWS oleh yang menyimpan templat fungsi rotasi Lambda.	
SecretsManagerReadWrite – Pembaruan ke kebijakan yang ada	Kebijakan ini diperbarui untuk mengizinkan akses deskripsi kan ke kluster Amazon RDS sehingga pengguna konsol dapat memilih klaster saat mereka membuat rahasia Amazon RDS.	3 Mei 2018
SecretsManagerReadWrite – Kebijakan baru	Secrets Manager membuat kebijakan untuk memberika n izin yang diperlukan untuk menggunakan konsol dengan semua akses baca/tulis ke Secrets Manager.	4 April 2018
Secrets Manager mulai melacak perubahan	Secrets Manager mulai melacak perubahan untuk kebijakan yang AWS dikelola.	4 April 2018

Tentukan siapa yang memiliki izin untuk rahasia Anda AWS Secrets Manager

Secara default, identitas IAM tidak memiliki izin untuk mengakses rahasia. Saat mengotorisasi akses ke rahasia, Secrets Manager mengevaluasi kebijakan berbasis sumber daya yang dilampirkan pada rahasia dan semua kebijakan berbasis identitas yang dilampirkan pada pengguna IAM atau peran yang mengirim permintaan. Untuk melakukan ini, Secrets Manager menggunakan proses yang mirip

dengan yang dijelaskan dalam Menentukan apakah permintaan diizinkan atau ditolak dalam Panduan Pengguna IAM.

Jika beberapa kebijakan berlaku untuk permintaan, Secrets Manager menggunakan hierarki untuk mengontrol izin:

1. Jika pernyataan dalam kebijakan apa pun dengan eksplisit `deny` cocok dengan tindakan permintaan dan sumber daya:

   Eksplisit `deny` mengesampingkan semua yang lain dan memblokir tindakan.

2. Jika tidak ada eksplisit`deny`, tetapi pernyataan dengan eksplisit `allow` cocok dengan tindakan permintaan dan sumber daya:

   Eksplisit `allow` memberikan tindakan dalam permintaan akses ke sumber daya dalam pernyataan.

   Jika identitas dan rahasia ada dalam dua akun yang berbeda, harus ada kebijakan sumber daya untuk rahasia dan kebijakan yang dilampirkan pada identitas, jika tidak AWS menolak permintaan tersebut. `allow` Untuk informasi selengkapnya, lihat Akses lintas akun.

3. Jika tidak ada pernyataan dengan eksplisit `allow` yang cocok dengan tindakan permintaan dan sumber daya:

   AWSmenolak permintaan secara default, yang disebut penolakan implisit.

Untuk melihat kebijakan berbasis sumber daya untuk rahasia

- Lakukan salah satu dari berikut:
  - Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/. Di halaman detail rahasia untuk rahasia Anda, di bagian Izin sumber daya, pilih Edit izin.
  - Gunakan AWS CLI to call `get-resource-policy`atau AWS SDK untuk menelepon `GetResourcePolicy`.

Untuk menentukan siapa yang memiliki akses melalui kebijakan berbasis identitas

- Gunakan simulator kebijakan IAM. Lihat Menguji kebijakan IAM dengan simulator kebijakan IAM

Izin untuk AWS Secrets Manager rahasia untuk pengguna di akun yang berbeda

Untuk memungkinkan pengguna dalam satu akun mengakses rahasia di akun lain (akses lintas akun), Anda harus mengizinkan akses baik dalam kebijakan sumber daya maupun dalam kebijakan identitas. Ini berbeda dengan memberikan akses ke identitas di akun yang sama dengan rahasia.

Anda juga harus mengizinkan identitas untuk menggunakan kunci KMS yang rahasianya dienkripsi. Ini karena Anda tidak dapat menggunakan Kunci yang dikelola AWS (`aws/secretsmanager`) untuk akses lintas akun. Sebagai gantinya, Anda harus mengenkripsi rahasia Anda dengan kunci KMS yang Anda buat, lalu lampirkan kebijakan kunci ke dalamnya. Ada biaya untuk membuat kunci KMS. Untuk mengubah kunci enkripsi untuk rahasia, lihatthe section called “Merubah rahasia”.

Contoh kebijakan berikut mengasumsikan Anda memiliki kunci rahasia dan enkripsi di Account1, dan identitas di Account2 yang ingin Anda izinkan untuk mengakses nilai rahasia.

Langkah 1: Lampirkan kebijakan sumber daya ke rahasia di Akun1

- *Kebijakan berikut memungkinkan ApplicationRoledi Account2 untuk mengakses rahasia di Account1.* Untuk menggunakan kebijakan ini, lihatthe section called “Lampirkan kebijakan izin ke rahasia”.

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

Langkah 2: Tambahkan pernyataan ke kebijakan kunci untuk kunci KMS di Akun1

- *Pernyataan kebijakan kunci berikut memungkinkan ApplicationRoledi Account2 untuk menggunakan kunci KMS di Account1 untuk mendekripsi rahasia di Account1.* Untuk menggunakan pernyataan ini, tambahkan ke kebijakan kunci untuk kunci KMS Anda. Untuk informasi selengkapnya, lihat Mengubah kebijakan utama.

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

Langkah 3: Lampirkan kebijakan identitas ke identitas di Akun2

- *Kebijakan berikut memungkinkan ApplicationRoledi Account2 untuk mengakses rahasia di Account1 dan mendekripsi nilai rahasia dengan menggunakan kunci enkripsi yang juga ada di Account1.* Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”. Anda dapat menemukan ARN untuk rahasia Anda di konsol Secrets Manager di halaman detail rahasia di bawah Rahasia ARN. Atau, Anda dapat menelepon `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"
```

```
    }
  ]
}
```

Izin peran eksekusi fungsi rotasi Lambda untuk AWS Secrets Manager

Secrets Manager menggunakan fungsi Lambda untuk memutar rahasia. Agar fungsi Lambda berjalan, Lambda mengasumsikan peran eksekusi IAM dan memberikan kredensil tersebut ke kode fungsi Lambda. Untuk petunjuk tentang cara mengatur rotasi otomatis, lihat:

- Rotasi otomatis untuk rahasia database (konsol)
- Rotasi otomatis (konsol)
- Rotasi otomatis (AWS CLI)

Contoh berikut menunjukkan kebijakan inline untuk peran eksekusi fungsi rotasi Lambda. Untuk membuat peran eksekusi dan melampirkan kebijakan izin, lihat peran AWS Lambda eksekusi.

Contoh:

- Kebijakan untuk peran eksekusi fungsi rotasi Lambda
- Pernyataan kebijakan untuk kunci yang dikelola pelanggan
- Pernyataan kebijakan untuk strategi pengguna bergantian

Kebijakan untuk peran eksekusi fungsi rotasi Lambda

Contoh kebijakan berikut memungkinkan fungsi rotasi untuk:

- Jalankan operasi Secrets Manager untuk *SecretArn*.
- Buat kata sandi baru.
- Siapkan konfigurasi yang diperlukan jika database atau layanan Anda berjalan di VPC. Lihat Mengonfigurasi fungsi Lambda untuk mengakses sumber daya di 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"
        }
    ]
}
```

Pernyataan kebijakan untuk kunci yang dikelola pelanggan

Jika rahasia dienkripsi dengan kunci KMS selain Kunci yang dikelola AWS`aws/secretsmanager`, maka Anda perlu memberikan izin peran eksekusi Lambda untuk menggunakan kunci tersebut. Anda dapat menggunakan konteks enkripsi secretArn untuk membatasi penggunaan fungsi dekripsi, sehingga peran fungsi rotasi hanya memiliki akses untuk mendekripsi rahasia yang bertanggung jawab untuk berputar. Contoh berikut menunjukkan pernyataan untuk ditambahkan ke kebijakan peran eksekusi untuk mendekripsi rahasia menggunakan kunci KMS.

```
        {
            "Effect": "Allow",
```

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

Untuk menggunakan fungsi rotasi untuk beberapa rahasia yang dienkripsi dengan kunci yang dikelola pelanggan, tambahkan pernyataan seperti contoh berikut untuk memungkinkan peran eksekusi mendekripsi rahasia.

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

Pernyataan kebijakan untuk strategi pengguna bergantian

Untuk informasi tentang strategi rotasi pengguna bergantian, lihatthe section called “Strategi rotasi”.

Untuk rahasia yang berisi kredensi Amazon RDS, jika Anda menggunakan strategi pengguna bergantian dan rahasia pengguna super dikelola oleh Amazon RDS, maka Anda juga harus mengizinkan fungsi rotasi untuk memanggil API hanya-baca di Amazon RDS sehingga bisa

mendapatkan informasi koneksi untuk database. Kami sarankan Anda melampirkan kebijakan AWS terkelola ReadOnlyAccessAmazonRDS.

Contoh kebijakan berikut memungkinkan fungsi untuk:

- Jalankan operasi Secrets Manager untuk *SecretArn*.
- Ambil kredensialnya di rahasia superuser. Secrets Manager menggunakan kredensil dalam rahasia superuser untuk memperbarui kredensil dalam rahasia yang diputar.
- Buat kata sandi baru.
- Siapkan konfigurasi yang diperlukan jika database atau layanan Anda berjalan di VPC. Untuk informasi selengkapnya, lihat Mengonfigurasi fungsi Lambda untuk mengakses sumber daya di 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"
        }
    ]
}
```

Contoh kebijakan izin untuk AWS Secrets Manager

Kebijakan izin adalah teks terstruktur JSON. Lihat Struktur dokumen kebijakan JSON.

Kebijakan izin yang Anda lampirkan ke sumber daya dan identitas sangat mirip. Beberapa elemen yang Anda sertakan dalam kebijakan untuk akses ke rahasia meliputi:

- `Principal`: siapa yang memberikan akses ke. Lihat Menentukan prinsipal dalam Panduan Pengguna IAM. Ketika Anda melampirkan kebijakan ke identitas, Anda tidak menyertakan `Principal` elemen dalam kebijakan.
- `Action`: apa yang bisa mereka lakukan. Lihat the section called “Tindakan Secrets Manager”.
- `Resource`: rahasia mana yang dapat mereka akses. Lihat the section called “Sumber daya Secrets Manager”.

  Karakter wildcard (*) memiliki arti yang berbeda tergantung pada apa yang Anda lampirkan kebijakan ke:

  - Dalam kebijakan yang dilampirkan pada rahasia, `*` berarti kebijakan tersebut berlaku untuk rahasia ini.
  - Dalam kebijakan yang dilampirkan pada identitas, `*` berarti kebijakan tersebut berlaku untuk semua sumber daya, termasuk rahasia, di akun.

Untuk melampirkan kebijakan ke rahasia, lihatthe section called “Lampirkan kebijakan izin ke rahasia”.

Untuk melampirkan kebijakan ke identitas, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

Topik

Contoh: Izin untuk mengambil nilai rahasia individu

Untuk memberikan izin untuk mengambil nilai rahasia, Anda dapat melampirkan kebijakan ke rahasia atau identitas. Untuk bantuan menentukan jenis kebijakan yang akan digunakan, lihat Kebijakan berbasis identitas dan kebijakan berbasis sumber daya. Untuk informasi tentang cara melampirkan kebijakan, lihat the section called “Lampirkan kebijakan izin ke rahasia” danthe section called “Melampirkan kebijakan izin ke identitas”.

Contoh berikut menunjukkan dua cara berbeda untuk memberikan akses ke rahasia. Contoh pertama adalah kebijakan berbasis sumber daya yang dapat Anda lampirkan ke rahasia. Contoh ini berguna ketika Anda ingin memberikan akses ke satu rahasia ke beberapa pengguna atau peran. Contoh kedua adalah kebijakan berbasis identitas yang dapat Anda lampirkan ke pengguna atau peran di IAM. Contoh ini berguna ketika Anda ingin memberikan akses ke grup IAM. Untuk memberikan izin untuk mengambil sekelompok rahasia dalam panggilan API batch, lihatthe section called “Izin untuk mengambil sekelompok nilai rahasia dalam batch”.

Example Baca satu rahasia (lampirkan ke rahasia)

Anda dapat memberikan akses ke rahasia dengan melampirkan kebijakan berikut ke rahasia. Untuk menggunakan kebijakan ini, lihatthe section called “Lampirkan kebijakan izin ke rahasia”.

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

```
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:role/EC2RoleToAccessSecrets"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

Example Baca satu rahasia (lampirkan ke identitas)

Anda dapat memberikan akses ke rahasia dengan melampirkan kebijakan berikut ke identitas. Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”. Jika Anda melampirkan kebijakan ini ke peran *EC2 RoleToAccessSecrets*, kebijakan tersebut akan memberikan izin yang sama dengan kebijakan sebelumnya.

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

Example Baca rahasia yang dienkripsi menggunakan kunci yang dikelola pelanggan (lampirkan ke identitas)

Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan, Anda dapat memberikan akses untuk membaca rahasia dengan melampirkan kebijakan berikut ke identitas. Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

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

```
    },
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "KMSKeyARN"
    }
  ]
}
```

Izin untuk mengambil sekelompok nilai rahasia dalam batch

Example Baca sekelompok rahasia dalam batch (lampirkan ke identitas)

Anda dapat memberikan akses untuk mengambil grup rahasia dalam panggilan API batch dengan melampirkan kebijakan berikut ke identitas. Kebijakan membatasi pemanggil sehingga mereka hanya dapat mengambil rahasia yang ditentukan oleh *SecretArn1, SecretArn2, dan SecretArn3, bahkan jika panggilan* batch menyertakan *rahasia lain.* Jika penelepon juga meminta rahasia lain dalam panggilan API batch, Secrets Manager tidak akan mengembalikannya. Untuk informasi selengkapnya, lihat the section called “Ambil rahasia dalam batch”. Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:BatchGetSecretValue",
        "secretsmanager:ListSecrets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "SecretARN1",
        "SecretARN2",
        "SecretARN3"
        ]
```

```
    }
  ]
}
```

Contoh: Wildcard

Anda dapat menggunakan wildcard untuk menyertakan sekumpulan nilai dalam elemen kebijakan.

Example Akses semua rahasia di jalur (lampirkan ke identitas)

Kebijakan berikut memberikan akses untuk mengambil semua rahasia dengan nama yang diawali dengan "*TestEnv/*". Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

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

Example Akses metadata pada semua rahasia (lampirkan ke identitas)

Pemberian `DescribeSecret` dan izin kebijakan berikut dimulai dengan`List`: `ListSecrets` dan. `ListSecretVersionIds` Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

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

Example Cocokkan nama rahasia (lampirkan ke identitas)

Kebijakan berikut memberikan semua izin Secrets Manager untuk rahasia berdasarkan nama. Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

Untuk mencocokkan nama rahasia, Anda membuat ARN untuk rahasia dengan menyusun Region, ID Akun, nama rahasia, dan wildcard (?) untuk mencocokkan karakter acak individual. Secrets Manager menambahkan enam karakter acak ke nama rahasia sebagai bagian dari ARN mereka, sehingga Anda dapat menggunakan wildcard ini untuk mencocokkan karakter tersebut. Jika Anda menggunakan sintaks`"another_secret_name-*"`, Secrets Manager tidak hanya cocok dengan rahasia yang dimaksudkan dengan 6 karakter acak, tetapi juga cocok`"another_secret_name-<anything-here>a1b2c3"`.

Karena Anda dapat memprediksi semua bagian ARN rahasia kecuali 6 karakter acak, menggunakan `'??????'` sintaks karakter wildcard memungkinkan Anda memberikan izin dengan aman ke rahasia yang belum ada. Namun, ketahuilah, jika Anda menghapus rahasia dan membuatnya kembali dengan nama yang sama, pengguna secara otomatis menerima izin untuk rahasia baru, meskipun 6 karakter berubah.

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

Contoh: Izin untuk membuat rahasia

Untuk memberikan izin pengguna untuk membuat rahasia, kami sarankan Anda melampirkan kebijakan izin ke grup IAM milik pengguna. Lihat grup pengguna IAM.

Example Buat rahasia (lampirkan ke identitas)

Kebijakan berikut memberikan izin untuk membuat rahasia dan melihat daftar rahasia. Untuk menggunakan kebijakan ini, lihatthe section called “Melampirkan kebijakan izin ke identitas”.

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

Contoh: Izin dan VPC

Jika Anda perlu mengakses Secrets Manager dari dalam VPC, Anda dapat memastikan bahwa permintaan ke Secrets Manager berasal dari VPC dengan menyertakan kondisi dalam kebijakan izin Anda. Untuk informasi selengkapnya, lihat Kondisi titik akhir VPC dan Titik akhir VPC.

Pastikan bahwa permintaan untuk mengakses rahasia dari AWS layanan lain juga berasal dari VPC, jika tidak kebijakan ini akan menolak akses mereka.

Example Memerlukan permintaan untuk datang melalui titik akhir VPC (lampirkan ke rahasia)

Kebijakan berikut memungkinkan pengguna untuk melakukan operasi Secrets Manager hanya ketika permintaan datang melalui titik akhir VPC. *vpce-1234a5678b9012c* Untuk menggunakan kebijakan ini, lihatthe section called “Lampirkan kebijakan izin ke rahasia”.

```
{
  "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 Memerlukan permintaan untuk datang dari VPC (lampirkan ke rahasia)

Kebijakan berikut memungkinkan perintah untuk membuat dan mengelola rahasia hanya ketika mereka berasal *vpc-12345678*. Selain itu, kebijakan memungkinkan operasi yang menggunakan akses nilai terenkripsi rahasia hanya ketika permintaan berasal. vpc-2b2b2b2b Anda mungkin menggunakan kebijakan seperti ini jika Anda menjalankan aplikasi dalam satu VPC, tetapi Anda menggunakan VPC kedua yang terisolasi untuk fungsi manajemen. Untuk menggunakan kebijakan ini, lihatthe section called “Lampirkan kebijakan izin ke rahasia”.

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

Contoh: Kontrol akses ke rahasia menggunakan tag

Anda dapat menggunakan tag untuk mengontrol akses ke rahasia Anda. Menggunakan tag untuk mengontrol izin sangat membantu di lingkungan yang berkembang pesat dan membantu situasi di mana manajemen kebijakan menjadi rumit. Salah satu strategi adalah melampirkan tag ke rahasia dan kemudian memberikan izin ke identitas ketika rahasia memiliki tag tertentu.

Example Izinkan akses ke rahasia dengan tag tertentu (lampirkan ke identitas)

Kebijakan berikut memungkinkan `DescribeSecret` pada rahasia dengan tag dengan kunci "" dan nilai *`ServerName`*"*`ServeABC`*". Untuk menggunakan kebijakan ini, lihatthe section called "Melampirkan kebijakan izin ke identitas".

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

```
        "secretsmanager:ResourceTag/ServerName": "ServerABC"
      }
    }
  }
}
```

Contoh: Batasi akses ke identitas dengan tag yang cocok dengan tag rahasia

Salah satu strateginya adalah melampirkan tag ke rahasia dan identitas IAM. Kemudian Anda membuat kebijakan izin untuk mengizinkan operasi ketika tag identitas cocok dengan tag rahasia. Untuk tutorial selengkapnya, lihat Menentukan izin untuk mengakses rahasia berdasarkan tag.

Menggunakan tag untuk mengontrol izin sangat membantu di lingkungan yang berkembang pesat dan membantu situasi di mana manajemen kebijakan menjadi rumit. Untuk informasi lebih lanjut, lihat Apa fungsi ABAC untuk AWS?

Example Izinkan akses ke peran yang memiliki tag yang sama dengan rahasia (lampirkan ke rahasia)

Kebijakan berikut *123456789012*hanya `GetSecretValue` memberikan akun jika tag *AccessProject*memiliki nilai yang sama untuk rahasia dan peran. Untuk menggunakan kebijakan ini, lihatthe section called “Lampirkan kebijakan izin ke rahasia”.

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

Contoh: Prinsipal layanan

Jika kebijakan sumber daya yang dilampirkan ke rahasia Anda menyertakan prinsip AWS layanan, kami sarankan Anda menggunakan kunci kondisi SourceAccount global aws: SourceArn dan aws:. ARN dan nilai akun disertakan dalam konteks otorisasi hanya ketika permintaan datang ke Secrets Manager dari layanan lain. AWS Kombinasi kondisi ini menghindari skenario wakil yang berpotensi membingungkan.

Jika ARN sumber daya menyertakan karakter yang tidak diizinkan dalam kebijakan sumber daya, Anda tidak dapat menggunakan ARN sumber daya tersebut dalam nilai kunci kondisi. `aws:SourceArn` Sebagai gantinya, gunakan tombol `aws:SourceAccount` kondisi. Untuk informasi selengkapnya, lihat persyaratan IAM.

Prinsipal layanan biasanya tidak digunakan sebagai prinsipal dalam kebijakan yang melekat pada rahasia, tetapi beberapa layanan memerlukannya. AWS Untuk informasi tentang kebijakan sumber daya yang diharuskan layanan untuk Anda lampirkan ke rahasia, lihat dokumentasi layanan.

Example Izinkan layanan mengakses rahasia menggunakan kepala layanan (lampirkan ke rahasia)

```
{
  "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"
        }
      }

    }
```

```
  ]
}
```

Referensi izin untuk AWS Secrets Manager

Untuk melihat elemen yang membentuk kebijakan izin, lihat Struktur dokumen kebijakan JSON dan referensi elemen kebijakan IAM JSON.

Untuk mulai menulis kebijakan izin Anda sendiri, lihatthe section called “Contoh kebijakan izin”.

Tindakan Secrets Manager

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
CancelRotateSecret	Memberikan izin untuk membatalkan rotasi rahasia yang sedang berlangsung	Tulis	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey}	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
CreateSecret	Memberikan izin untuk membuat rahasia yang menyimpan data terenkripsi yang dapat ditanyakan dan diputar	Tulis	Secret*		
				secretsmanager:Name secretsmanager:Description secretsmanager:KmsKeyId aws:RequestTag/${TagKey} aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:AddReplicaRegions	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:ForceOverwriteReplicaSecret	
DeleteResourcePolicy	Memberikan izin untuk menghapus kebijakan sumber daya yang dilampirkan pada rahasia	Manajemen izin	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
DeleteSecret	Memberikan izin untuk menghapus rahasia	Tulis	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:RecoveryWindowInDays secretsmanager:ForceDeleteWithoutRecovery secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:Sec	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				retPrimaryRegion	
DescribeSecret	Memberikan izin untuk mengambil metadata tentang rahasia, tetapi bukan data terenkripsi	Baca	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetRandomPassword	Memberikan izin untuk menghasilkan string acak untuk digunakan dalam pembuatan kata sandi	Baca			

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
GetResourcePolicy	Memberikan izin untuk mendapatkan kebijakan sumber daya yang melekat pada rahasia	Baca	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetSecretValue	Memberikan izin untuk mengambil dan mendekripsi data terenkripsi	Baca	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:VersionId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
ListSecretVersionIds	Memberikan izin untuk membuat daftar versi rahasia yang tersedia	Baca	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ListSecrets	Memberikan izin untuk membuat daftar rahasia yang tersedia	Daftar			
PutResourcePolicy	Memberikan izin untuk melampirkan kebijakan sumber daya ke rahasia	Manajemen izin	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:BlockPublicPolicy secretsmanager:SecretPrimaryRegion	
PutSecretValue	Memberikan izin untuk membuat versi baru rahasia dengan data terenkripsi baru	Tulis	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RemoveRegionsFromReplication	Memberikan izin untuk menghapus wilayah dari replikasi	Tulis	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
ReplicateSecretToRegions	Memberikan izin untuk mengubah rahasia yang ada menjadi rahasia Multi-wilayah dan mulai mereplikasi rahasia ke daftar wilayah baru	Tulis	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:AddReplicaRegions secretsmanager:For	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				ceOverwriteReplicaSecret	
RestoreSecret	Memberikan izin untuk membatalkan penghapusan rahasia	Tulis	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RotateSecret	Memberikan izin untuk memulai rotasi rahasia	Tulis	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:RotationLambdaARN secretsmanager:resource/AllowRotationLambdaAr n secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:Mod	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				ifyRotationRules secretsmanager:RotateImmediately	
StopReplicationToReplica	Memberikan izin untuk menghapus rahasia dari replikasi dan mempromosikan rahasia ke rahasia regional di wilayah replika	Tulis	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
TagResource	Memberikan izin untuk menambahkan tag ke rahasia	Penandaan	Secret*		
				secretsmanager:SecretId aws:RequestTag/${TagKey} aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
UntagResource	Memberikan izin untuk menghapus tag dari rahasia	Penandaan	Secret*		
				secretsmanager:SecretId aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
UpdateSecret	Memberikan izin untuk memperbarui rahasia dengan metadata baru atau dengan versi baru dari data terenkripsi	Tulis	Secret*		

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
				secretsmanager:SecretId secretsmanager:Description secretsmanager:KmsKeyId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
UpdateSecretVersionStage	Memberikan izin untuk memindahkan panggung dari satu rahasia ke rahasia lainnya	Tulis	Secret*		
				secretsmanager:SecretId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Tindakan	Deskripsi	Tingkat akses	Jenis sumber daya (*wajib)	Kunci kondisi	Tindakan bergantung
ValidateResourcePolicy	Memberikan izin untuk memvalidasi kebijakan sumber daya sebelum melampirkan kebijakan	Manajemen izin	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Sumber daya Secrets Manager

Jenis sumber daya	ARN	Kunci syarat
Secret	`arn:${Partition}:secretsmanager:${Region}:${Account}:secret:${SecretId}`	aws:RequestTag/${TagKey} aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:resource/AllowRotationLambdaArn

Secrets Manager membangun bagian terakhir dari ARN rahasia dengan menambahkan tanda hubung dan enam karakter alfanumerik acak di akhir nama rahasia. Jika Anda menghapus rahasia dan kemudian membuat ulang rahasia lain dengan nama yang sama, pemformatan ini membantu memastikan bahwa individu dengan izin ke rahasia asli tidak secara otomatis mendapatkan akses ke rahasia baru karena Secrets Manager menghasilkan enam karakter acak baru.

Anda dapat menemukan ARN untuk rahasia di konsol Secrets Manager di halaman detail rahasia atau dengan menelepon. `DescribeSecret`

Kunci syarat

Jika Anda menyertakan kondisi string dari tabel berikut dalam kebijakan izin, penelepon ke Secrets Manager harus meneruskan parameter yang cocok atau mereka ditolak aksesnya. Untuk menghindari penolakan penelepon untuk parameter yang hilang, tambahkan `IfExists` ke akhir nama operator kondisi, misalnya. `StringLikeIfExists` Untuk informasi selengkapnya, lihat elemen kebijakan IAM JSON: Operator kondisi.

Kunci syarat	Deskripsi	Tipe
aws:RequestTag/${TagKey}	Memfilter akses dengan kunci yang ada dalam permintaan yang dibuat pengguna ke layanan Secrets Manager	String
aws:ResourceTag/${TagKey}	Memfilter akses dengan tag yang terkait dengan sumber daya	String
aws:TagKeys	Memfilter akses berdasarkan daftar semua nama kunci tag yang ada dalam permintaan yang dibuat pengguna ke layanan Secrets Manager	ArrayOfString
secretsmanager:AddReplicaRegions	Memfilter akses berdasarkan daftar Wilayah untuk mereplikasi rahasia	ArrayOfString
secretsmanager:BlockPublicPolicy	Memfilter akses berdasarkan apakah kebijakan sumber daya memblokir Akun AWS akses luas	Bool
secretsmanager:Description	Memfilter akses dengan teks deskripsi dalam permintaan	String
secretsmanager:ForceDeleteWithoutRecovery	Memfilter akses dengan apakah rahasianya akan segera dihapus tanpa jendela pemulihan	Bool
secretsmanager:ForceOverwriteReplicaSecret	Memfilter akses berdasarkan apakah akan menimpa rahasia dengan nama yang sama di Wilayah tujuan	Bool
secretsmanager:KmsKeyId	Memfilter akses oleh ARN dari kunci KMS dalam permintaan	String

Kunci syarat	Deskripsi	Tipe
secretsmanager:ModifyRotationRules	Memfilter akses dengan apakah aturan rotasi rahasia akan dimodifikasi	Bool
secretsmanager:Name	Memfilter akses dengan nama rahasia yang ramah dalam permintaan	String
secretsmanager:RecoveryWindowInDays	Memfilter akses berdasarkan jumlah hari yang menunggu Secrets Manager sebelum dapat menghapus rahasia	Numerik
secretsmanager:ResourceTag/tag-key	Memfilter akses dengan kunci tag dan pasangan nilai	String
secretsmanager:RotateImmediately	Memfilter akses dengan apakah rahasianya akan segera diputar	Bool
secretsmanager:RotationLambdaARN	Memfilter akses oleh ARN dari fungsi Lambda rotasi dalam permintaan	ARN
secretsmanager:SecretId	Memfilter akses berdasarkan nilai secretID dalam permintaan	ARN
secretsmanager:SecretPrimaryRegion	Memfilter akses berdasarkan wilayah utama tempat rahasia dibuat	String
secretsmanager:VersionId	Memfilter akses oleh pengenal unik dari versi rahasia dalam permintaan	String

Kunci syarat	Deskripsi	Tipe
secretsmanager:VersionStage	Memfilter akses berdasarkan daftar tahapan versi dalam permintaan	String
secretsmanager:resource/AllowRotationLambdaArn	Memfilter akses oleh ARN dari fungsi rotasi Lambda yang terkait dengan rahasia	ARN

Blokir akses luas ke rahasia dengan **`BlockPublicPolicy`** kondisi

Dalam kebijakan identitas yang memungkinkan tindakan`PutResourcePolicy`, kami sarankan Anda menggunakannya`BlockPublicPolicy: true`. Kondisi ini berarti bahwa pengguna hanya dapat melampirkan kebijakan sumber daya ke rahasia jika kebijakan tidak mengizinkan akses luas.

Secrets Manager menggunakan penalaran otomatis Zelkova untuk menganalisis kebijakan sumber daya untuk akses luas. Untuk informasi selengkapnya tentang Zelkova, lihat Cara AWS menggunakan penalaran otomatis untuk membantu Anda mencapai keamanan dalam skala besar di Blog Keamanan. AWS

Contoh berikut menunjukkan cara menggunakan`BlockPublicPolicy`.

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

Kondisi alamat IP

Berhati-hatilah saat Anda menentukan operator kondisi alamat IP atau kunci `aws:SourceIp` kondisi dalam pernyataan kebijakan yang mengizinkan atau menolak akses ke Secrets Manager. Misalnya, jika Anda melampirkan kebijakan yang membatasi AWS tindakan untuk permintaan dari rentang alamat IP jaringan perusahaan Anda ke rahasia, maka permintaan Anda sebagai pengguna IAM yang menjalankan permintaan dari jaringan perusahaan berfungsi seperti yang diharapkan. Namun, jika Anda mengaktifkan layanan lain untuk mengakses rahasia atas nama Anda, seperti saat Anda mengaktifkan rotasi dengan fungsi Lambda, fungsi tersebut akan memanggil operasi Secrets Manager dari ruang alamat AWS -internal. Permintaan yang terkena dampak kebijakan dengan filter alamat IP gagal.

Selain itu, kunci `aws:sourceIP` kondisi kurang efektif ketika permintaan berasal dari titik akhir Amazon VPC. Untuk membatasi permintaan ke titik akhir VPC tertentu, gunakan. the section called “Kondisi titik akhir VPC”

Kondisi titik akhir VPC

Untuk mengizinkan atau menolak akses ke permintaan dari titik akhir VPC atau VPC tertentu, gunakan `aws:SourceVpc` untuk membatasi akses ke permintaan dari VPC yang ditentukan atau `aws:SourceVpce` untuk membatasi akses ke permintaan dari titik akhir VPC yang ditentukan. Lihat the section called “Contoh: Izin dan VPC”.

- `aws:SourceVpc` membatasi akses ke permintaan dari VPC yang ditentukan.
- `aws:SourceVpce` membatasi akses ke permintaan dari VPC endpoint yang ditentukan.

Jika Anda menggunakan kunci kondisi ini dalam pernyataan kebijakan sumber daya yang mengizinkan atau menolak akses ke rahasia Secrets Manager, Anda dapat secara tidak sengaja menolak akses ke layanan yang menggunakan Secrets Manager untuk mengakses rahasia atas nama Anda. Hanya beberapa AWS layanan yang dapat berjalan dengan titik akhir dalam VPC Anda. Jika Anda membatasi permintaan rahasia ke titik akhir VPC atau VPC, maka panggilan ke Secrets Manager dari layanan yang tidak dikonfigurasi untuk layanan dapat gagal.

Lihat Titik akhir VPC.

Buat dan kelola rahasia dengan AWS Secrets Manager

Rahasia dapat berupa kata sandi, seperangkat kredensil seperti nama pengguna dan kata sandi, token OAuth, atau informasi rahasia lainnya yang Anda simpan dalam bentuk terenkripsi di Secrets Manager.

Topik

Buat rahasia AWS Secrets Manager database

Setelah Anda membuat pengguna di Amazon RDS, Amazon Aurora, Amazon Redshift, atau Amazon DocumentDB, Anda dapat menyimpan kredensialnya di Secrets Manager dengan mengikuti langkah-langkah ini. Ketika Anda menggunakan AWS CLI atau salah satu SDK untuk menyimpan rahasia, Anda harus memberikan rahasia dalam struktur JSON yang benar. Saat Anda menggunakan konsol untuk menyimpan rahasia database, Secrets Manager secara otomatis membuatnya dalam struktur JSON yang benar.

> **Tip**
>
> Untuk kredensyal pengguna admin Amazon RDS dan Amazon Redshift, kami sarankan Anda menggunakan rahasia terkelola. Anda membuat rahasia terkelola melalui servce pengelolaan, dan kemudian Anda dapat menggunakan rotasi terkelola.

Ketika Anda menyimpan kredensyal database untuk database sumber yang direplikasi ke Wilayah lain, rahasia berisi informasi koneksi untuk database sumber. Jika Anda kemudian mereplikasi rahasia, replika adalah salinan dari rahasia sumber dan berisi informasi koneksi yang sama. Anda dapat menambahkan pasangan kunci/nilai tambahan ke rahasia untuk informasi koneksi regional.

Untuk membuat rahasia, Anda memerlukan izin yang diberikan oleh. SecretsManagerReadWriteAWS kebijakan terkelola

Secrets Manager menghasilkan entri CloudTrail log saat Anda membuat rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk membuat rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih jenis rahasia, lakukan hal berikut:
   a. Untuk tipe Rahasia, pilih jenis kredensyal database yang akan disimpan:
      - Basis data Amazon RDS (termasuk Aurora)
      - Basis data Amazon DocumentDB
      - Gudang data Amazon Redshift
   b. Untuk Kredensyal, masukkan kredensyal untuk database.
   c. Untuk kunci Enkripsi, pilih Secrets Manager AWS KMS key yang digunakan untuk mengenkripsi nilai rahasia. Untuk informasi selengkapnya, lihat Enkripsi rahasia dan dekripsi.
      - Untuk kebanyakan kasus, pilih aws/secretsmanager untuk menggunakan for Kunci yang dikelola AWS Secrets Manager. Tidak ada biaya untuk menggunakan kunci ini.
      - Jika Anda perlu mengakses rahasia dari yang lain Akun AWS, atau jika Anda ingin menggunakan kunci KMS Anda sendiri sehingga Anda dapat memutarnya atau

menerapkan kebijakan kunci untuk itu, pilih kunci yang dikelola pelanggan dari daftar atau pilih Tambahkan kunci baru untuk membuatnya. Untuk informasi tentang biaya penggunaan kunci yang dikelola pelanggan, lihatHarga.

Anda harus memilikithe section called “Izin untuk kunci KMS”. Untuk informasi tentang akses lintas akun, lihatthe section called “Akses lintas akun”.

d. Untuk Database, pilih database Anda.

e. Pilih Berikutnya.

4. Pada halaman Konfigurasi rahasia, lakukan hal berikut:

a. Masukkan nama Rahasia deskriptif dan Deskripsi. Nama rahasia harus berisi 1-512 karakter Unicode.

b. (Opsional) Di bagian Tag, tambahkan tag ke rahasia Anda. Untuk strategi penandaan, lihatthe section called “Rahasia tag”. Jangan menyimpan informasi sensitif dalam tag karena tidak dienkripsi.

c. (Opsional) Di Izin sumber daya, untuk menambahkan kebijakan sumber daya ke rahasia Anda, pilih Edit izin. Untuk informasi selengkapnya, lihat the section called “Lampirkan kebijakan izin ke rahasia”.

d. (Opsional) Dalam rahasia Replikasi, untuk mereplikasi rahasia Anda ke yang lain Wilayah AWS, pilih Replikasi rahasia. Anda dapat mereplikasi rahasia Anda sekarang atau kembali dan mereplikasi nanti. Untuk informasi selengkapnya, lihat Replikasi rahasia ke Wilayah lain.

e. Pilih Berikutnya.

5. (Opsional) Pada halaman Konfigurasi rotasi, Anda dapat mengaktifkan rotasi otomatis. Anda juga dapat mematikan rotasi untuk saat ini dan kemudian menyalakannya nanti. Untuk informasi selengkapnya, lihat Putar rahasia. Pilih Berikutnya.

6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Secrets Manager kembali ke daftar rahasia. Jika rahasia baru Anda tidak muncul, pilih tombol refresh.

AWS CLI

Saat Anda memasukkan perintah di shell perintah, ada risiko riwayat perintah diakses atau utilitas memiliki akses ke parameter perintah Anda. Lihat the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

Example Buat rahasia dari kredensyal dalam file JSON

`create-secret`Contoh berikut membuat rahasia dari kredensyal dalam file. Untuk informasi selengkapnya, lihat Memuat AWS CLI parameter dari file di Panduan AWS CLI Pengguna.

Agar Secrets Manager dapat memutar rahasia, Anda harus memastikan JSON cocok dengan. Struktur JSON dari sebuah rahasia

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

Isi mycreds.json:

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

AWS SDK

Untuk membuat rahasia dengan menggunakan salah satu AWS SDK, gunakan `CreateSecret`tindakan. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Struktur rahasia JSON AWS Secrets Manager

Anda dapat menyimpan teks atau biner apa pun dalam rahasia Secrets Manager. Jika Anda ingin mengaktifkan rotasi otomatis untuk rahasia Secrets Manager, itu harus dalam struktur JSON yang benar. Selama rotasi, Secrets Manager menggunakan informasi dalam rahasia untuk terhubung ke sumber kredensi dan memperbarui kredensil di sana. Nama kunci JSON peka huruf besar/kecil.

Perhatikan bahwa ketika Anda menggunakan konsol untuk menyimpan rahasia database, Secrets Manager secara otomatis membuatnya dalam struktur JSON yang benar.

Anda dapat menambahkan lebih banyak pasangan kunci/nilai ke rahasia, misalnya dalam rahasia database, untuk memuat informasi koneksi untuk database replika di Wilayah lain.

Topik

Struktur rahasia Amazon RDS Db2

Untuk instans Amazon RDS Db2, karena pengguna tidak dapat mengubah kata sandi mereka sendiri, Anda harus memberikan kredensi admin dalam rahasia terpisah.

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

Struktur rahasia Amazon RDS 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>
}
```

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Amazon RDS dan Amazon Aurora MySQL struktur rahasia

```
{
  "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>
}
```

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Struktur rahasia Amazon RDS Oracle

```
{
  "engine": "oracle",
```

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

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Amazon RDS dan Amazon Aurora PostgreSQL struktur rahasia

```
{
  "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>
}
```

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

```
  "masterarn": "<the ARN of the elevated secret>"
}
```

Amazon RDS Microsoft SQLServer struktur rahasia

```
{
  "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>
}
```

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Struktur rahasia Amazon DocumentDB

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

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Struktur rahasia Amazon Redshift

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

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Amazon Redshift Struktur rahasia tanpa server

```
{
  "engine": "redshift",
  "host": "<instance host name/resolvable DNS name>",
  "username": "<username>",
```

```
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "namespaceName": <namespace name>,
  "port": <TCP port number. If not specified, defaults to 5439>
}
```

Untuk menggunakanthe section called “Pengguna bergantian”, Anda menyertakan `masterarn` untuk rahasia yang berisi kredensi admin atau superuser.

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

Struktur ElastiCache rahasia Amazon

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

Untuk informasi selengkapnya, lihat Memutar kata sandi secara otomatis untuk pengguna di Panduan ElastiCache Pengguna Amazon.

Buat AWS Secrets Manager rahasia

Untuk menyimpan kunci API, token akses, kredensyal yang bukan untuk database, dan rahasia lainnya di Secrets Manager, ikuti langkah-langkah berikut. Untuk ElastiCache rahasia Amazon, jika Anda ingin mengaktifkan rotasi, Anda harus menyimpan rahasia dalam struktur JSON yang diharapkan.

Untuk membuat rahasia, Anda memerlukan izin yang diberikan oleh. SecretsManagerReadWriteAWS kebijakan terkelola

Secrets Manager menghasilkan entri CloudTrail log saat Anda membuat rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk membuat rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Pada halaman Pilih jenis rahasia, lakukan hal berikut:

   a. Untuk tipe Rahasia, pilih Jenis rahasia lainnya.

   b. Dalam pasangan kunci/nilai, masukkan rahasia Anda di pasangan kunci/Nilai JSON, atau pilih tab Plaintext dan masukkan rahasia dalam format apa pun. Anda dapat menyimpan hingga 65536 byte dalam rahasia.

   c. Untuk kunci Enkripsi, pilih Secrets Manager AWS KMS key yang digunakan untuk mengenkripsi nilai rahasia. Untuk informasi selengkapnya, lihat Enkripsi rahasia dan dekripsi.

      - Untuk kebanyakan kasus, pilih aws/secretsmanager untuk menggunakan for Kunci yang dikelola AWS Secrets Manager. Tidak ada biaya untuk menggunakan kunci ini.
      - Jika Anda perlu mengakses rahasia dari yang lainAkun AWS, atau jika Anda ingin menggunakan kunci KMS Anda sendiri sehingga Anda dapat memutarnya atau menerapkan kebijakan kunci untuk itu, pilih kunci yang dikelola pelanggan dari daftar atau pilih Tambahkan kunci baru untuk membuatnya. Untuk informasi tentang biaya penggunaan kunci yang dikelola pelanggan, lihatHarga.

        Anda harus memilikithe section called “Izin untuk kunci KMS”. Untuk informasi tentang akses lintas akun, lihatthe section called “Akses lintas akun”.

   d. Pilih Selanjutnya.

4. Pada halaman Konfigurasi rahasia, lakukan hal berikut:

   a. Masukkan nama Rahasia deskriptif dan Deskripsi. Nama rahasia harus berisi 1-512 karakter Unicode.

   b. (Opsional) Di bagian Tag, tambahkan tag ke rahasia Anda. Untuk strategi penandaan, lihatthe section called “Rahasia tag”. Jangan menyimpan informasi sensitif dalam tag karena tidak dienkripsi.

c. (Opsional) Di Izin sumber daya, untuk menambahkan kebijakan sumber daya ke rahasia Anda, pilih Edit izin. Untuk informasi selengkapnya, lihat the section called “Lampirkan kebijakan izin ke rahasia”.

d. (Opsional) Dalam rahasia Replikasi, untuk mereplikasi rahasia Anda ke yang lainWilayah AWS, pilih Replikasi rahasia. Anda dapat mereplikasi rahasia Anda sekarang atau kembali dan mereplikasi nanti. Untuk informasi selengkapnya, lihat Replikasi rahasia ke Wilayah lain.

e. Pilih Next (Berikutnya).

5. (Opsional) Pada halaman Konfigurasi rotasi, Anda dapat mengaktifkan rotasi otomatis. Anda juga dapat mematikan rotasi untuk saat ini dan kemudian menyalakannya nanti. Untuk informasi selengkapnya, lihat Putar rahasia. Pilih Next (Berikutnya).
6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

   Secrets Manager kembali ke daftar rahasia. Jika rahasia baru Anda tidak muncul, pilih tombol refresh.

AWS CLI

Saat Anda memasukkan perintah di shell perintah, ada risiko riwayat perintah diakses atau utilitas memiliki akses ke parameter perintah Anda. Lihat the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

Example Buat rahasia

`create-secret`Contoh berikut menciptakan rahasia dengan dua pasangan kunci-nilai.

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

Example Buat rahasia dari kredensyal dalam file JSON

`create-secret`Contoh berikut membuat rahasia dari kredensyal dalam file. Untuk informasi selengkapnya, lihat Memuat AWS CLI parameter dari file di Panduan AWS CLI Pengguna.

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

Isi mycreds.json:

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

AWS SDK

Untuk membuat rahasia dengan menggunakan salah satu AWS SDK, gunakan `CreateSecret`tindakan. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Perbarui nilai untuk AWS Secrets Manager rahasia

Untuk memperbarui nilai rahasia Anda, Anda dapat menggunakan konsol, CLI, atau SDK. Saat Anda memperbarui nilai rahasia, Secrets Manager membuat versi baru rahasia dengan label `AWSCURRENT` pementasan. Anda masih dapat mengakses versi lama, yang memiliki label`AWSPREVIOUS`. Anda juga dapat menambahkan label Anda sendiri. Untuk informasi selengkapnya, lihat Pembuatan versi Secrets Manager.

Untuk memperbarui nilai rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dari daftar rahasia, pilih rahasia Anda.
3. Pada halaman detail rahasia, pada tab Ikhtisar, di bagian Nilai rahasia, pilih Ambil nilai rahasia dan kemudian pilih Edit.

AWS CLI

Untuk memperbarui nilai rahasia (AWS CLI)

- Saat Anda memasukkan perintah di shell perintah, ada risiko riwayat perintah diakses atau utilitas memiliki akses ke parameter perintah Anda. Lihat the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

  Berikut ini `put-secret-value`menciptakan versi baru dari rahasia dengan dua pasangan kunci-nilai.

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

Berikut ini `put-secret-value`membuat versi baru dengan label pementasan kustom. Versi baru akan memiliki label `MyLabel` dan`AWSCURRENT`.

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

AWS SDK

Kami menyarankan Anda menghindari menelepon `PutSecretValue` atau dengan `UpdateSecret` kecepatan berkelanjutan lebih dari sekali setiap 10 menit. Saat Anda `UpdateSecret` menelepon `PutSecretValue` atau memperbarui nilai rahasia, Secrets Manager membuat versi baru dari rahasia tersebut. Secrets Manager menghapus versi yang tidak berlabel ketika ada lebih dari 100, tetapi tidak menghapus versi yang dibuat kurang dari 24 jam yang lalu. Jika Anda memperbarui nilai rahasia lebih dari sekali setiap 10 menit, Anda membuat lebih banyak versi daripada yang dihapus Secrets Manager, dan Anda akan mencapai kuota untuk versi rahasia.

Untuk memperbarui nilai rahasia, gunakan tindakan berikut: `UpdateSecret`atau `PutSecretValue`. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Ubah kunci enkripsi untuk AWS Secrets Manager rahasia

Secrets Manager menggunakan enkripsi amplop dengan AWS KMS kunci dan kunci data untuk melindungi setiap nilai rahasia. Untuk setiap rahasia, Anda dapat memilih kunci KMS mana yang akan digunakan. Anda dapat menggunakan Kunci yang dikelola AWS aws/secretsmanager, atau Anda dapat menggunakan kunci yang dikelola pelanggan. Untuk kebanyakan kasus, kami sarankan menggunakan aws/secretsmanager, dan tidak ada biaya untuk menggunakannya. Jika Anda perlu mengakses rahasia dari yang lain Akun AWS, atau jika Anda ingin menggunakan kunci KMS Anda sendiri sehingga Anda dapat memutarnya atau menerapkan kebijakan kunci untuk itu, gunakan file. kunci yang dikelola pelanggan Anda harus memilikithe section called “Izin untuk kunci KMS”. Untuk informasi tentang biaya penggunaan kunci yang dikelola pelanggan, lihatHarga.

Anda dapat mengubah kunci enkripsi untuk rahasia Anda. Misalnya, jika Anda ingin mengakses rahasia dari akun lain, dan rahasia saat ini dienkripsi menggunakan kunci yang AWS dikelolaaws/secretsmanager, Anda dapat beralih ke file. kunci yang dikelola pelanggan

> **Tip**
>
> Jika Anda ingin memutar kunci yang dikelola pelanggan, kami sarankan menggunakan rotasi tombol AWS KMS otomatis. Untuk informasi selengkapnya, lihat Memutar AWS KMS tombol.

Saat Anda mengubah kunci enkripsi, Secrets Manager mengenkripsi ulang AWSCURRENTAWSPENDING, dan AWSPREVIOUS versi dengan kunci baru. Untuk menghindari mengunci Anda dari rahasia, Secrets Manager menyimpan semua versi yang ada dienkripsi dengan kunci sebelumnya. Itu berarti Anda dapat mendekripsiAWSCURRENT,AWSPENDING, dan AWSPREVIOUS versi dengan kunci sebelumnya atau kunci baru.

Untuk membuatnya sehingga hanya AWSCURRENT dapat didekripsi oleh kunci enkripsi baru, buat versi baru rahasia dengan kunci baru. Kemudian untuk dapat mendekripsi versi AWSCURRENT rahasia, Anda harus memiliki izin untuk kunci baru.

Jika Anda menonaktifkan kunci enkripsi sebelumnya, Anda tidak akan dapat mendekripsi versi rahasia apa pun kecualiAWSCURRENT,, AWSPENDING dan. AWSPREVIOUS Jika Anda memiliki versi rahasia berlabel lain yang ingin Anda pertahankan aksesnya, Anda perlu membuat ulang versi tersebut dengan kunci enkripsi baru menggunakan. the section called “AWS CLI”

Untuk mengubah kunci enkripsi untuk rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dari daftar rahasia, pilih rahasia Anda.
3. Pada halaman detail rahasia, di bagian Detail rahasia, pilih Tindakan, lalu pilih Edit kunci enkripsi.

AWS CLI

Jika Anda mengubah kunci enkripsi untuk rahasia dan kemudian menonaktifkan kunci enkripsi sebelumnya, Anda tidak akan dapat mendekripsi versi rahasia apa pun kecualiAWSCURRENT,, AWSPENDING dan. AWSPREVIOUS Jika Anda memiliki versi rahasia berlabel lain yang ingin Anda

pertahankan aksesnya, Anda perlu membuat ulang versi tersebut dengan kunci enkripsi baru menggunakan. the section called “AWS CLI”

Untuk mengubah kunci enkripsi untuk secret (AWS CLI)

1. `update-secret`Contoh berikut memperbarui kunci KMS yang digunakan untuk mengenkripsi nilai rahasia. Kunci KMS harus berada di wilayah yang sama dengan rahasia.

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

2. (Opsional) Jika Anda memiliki versi rahasia yang memiliki label khusus, untuk mengenkripsi ulang menggunakan kunci baru, Anda harus membuat ulang versi tersebut.

   Saat Anda memasukkan perintah di shell perintah, ada risiko riwayat perintah diakses atau utilitas memiliki akses ke parameter perintah Anda. Lihat the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

   a. Dapatkan nilai dari versi rahasia.

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

      Catat nilai rahasianya.

   b. Buat versi baru dengan nilai itu.

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

Memodifikasi AWS Secrets Manager rahasia

Anda dapat memodifikasi metadata rahasia setelah dibuat, tergantung pada siapa yang membuat rahasia. Untuk rahasia yang dibuat oleh layanan lain, Anda mungkin perlu menggunakan layanan lain untuk memperbarui atau memutarnya.

Untuk menentukan siapa yang mengelola rahasia, Anda dapat meninjau nama rahasia. Rahasia yang dikelola oleh layanan lain diawali dengan ID layanan tersebut. Atau, diAWS CLI, panggil deskripsikan-rahasia, dan kemudian tinjau bidangnya. `OwningService` Untuk informasi selengkapnya, lihat Rahasia yang dikelola oleh layanan lain.

Untuk rahasia yang Anda kelola, Anda dapat mengubah deskripsi, kebijakan berbasis sumber daya, kunci enkripsi, dan tag. Anda juga dapat mengubah nilai rahasia terenkripsi; namun, kami sarankan Anda menggunakan rotasi untuk memperbarui nilai rahasia yang berisi kredensil. Rotasi memperbarui rahasia di Secrets Manager dan kredensil pada database atau layanan. Ini membuat rahasia disinkronkan secara otomatis sehingga ketika klien meminta nilai rahasia, mereka selalu mendapatkan seperangkat kredensil yang berfungsi. Untuk informasi selengkapnya, lihat Putar rahasia.

Secrets Manager menghasilkan entri CloudTrail log saat Anda memodifikasi rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk memperbarui rahasia yang Anda kelola (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dari daftar rahasia, pilih rahasia Anda.
3. Pada halaman rahasia, lakukan salah satu hal berikut:

   Perhatikan bahwa Anda tidak dapat mengubah nama atau ARN rahasia.

   - Untuk memperbarui deskripsi, di bagian Detail rahasia, pilih Tindakan, lalu pilih Edit deskripsi.
   - Untuk memperbarui kunci enkripsi, lihatthe section called “Ubah kunci enkripsi untuk rahasia”.
   - Untuk memperbarui tag, pada tab Tag, pilih Edit tag. Lihat the section called “Rahasia tag”.
   - Untuk memperbarui nilai rahasia, lihatthe section called “Perbarui nilai rahasia”.
   - Untuk memperbarui izin rahasia Anda, pada tab Ikhtisar, pilih Edit izin. Lihat the section called “Lampirkan kebijakan izin ke rahasia”.

   - Untuk memperbarui rotasi rahasia Anda, pada tab Rotasi, pilih Edit rotasi. Lihat Putar rahasia.
   - Untuk mereplikasi rahasia Anda ke Wilayah lain, lihatReplikasi rahasia ke Wilayah lain.
   - Jika rahasia Anda memiliki replika, Anda dapat mengubah kunci enkripsi untuk replika. Pada tab Replikasi, pilih tombol radio untuk replika, dan kemudian pada menu Tindakan, pilih Edit kunci enkripsi. Lihat the section called “Enkripsi rahasia dan dekripsi”.

- Untuk mengubah rahasia sehingga dikelola oleh layanan lain, Anda perlu membuat ulang rahasia dalam layanan itu. Lihat Rahasia yang dikelola oleh layanan lain.

AWS CLI

Example Perbarui deskripsi rahasia

`update-secret`Contoh berikut memperbarui deskripsi rahasia.

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

AWS SDK

Kami menyarankan Anda menghindari menelepon `PutSecretValue` atau dengan `UpdateSecret` kecepatan berkelanjutan lebih dari sekali setiap 10 menit. Saat Anda `UpdateSecret` menelepon `PutSecretValue` atau memperbarui nilai rahasia, Secrets Manager membuat versi baru dari rahasia tersebut. Secrets Manager menghapus versi yang tidak berlabel ketika ada lebih dari 100, tetapi tidak menghapus versi yang dibuat kurang dari 24 jam yang lalu. Jika Anda memperbarui nilai rahasia lebih dari sekali setiap 10 menit, Anda membuat lebih banyak versi daripada yang dihapus Secrets Manager, dan Anda akan mencapai kuota untuk versi rahasia.

Untuk memperbarui rahasia, gunakan tindakan berikut: `UpdateSecret`atau `ReplicateSecretToRegions`. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Temukan rahasia di AWS Secrets Manager

Saat Anda mencari rahasia tanpa filter, Secrets Manager mencocokkan kata kunci dalam nama rahasia, deskripsi, kunci tag, dan nilai tag. Pencarian tanpa filter tidak peka huruf besar/kecil dan mengabaikan karakter khusus, seperti spasi,/, _, =, #, dan hanya menggunakan angka dan huruf. Saat Anda mencari tanpa filter, Secrets Manager menganalisis string pencarian untuk mengubahnya menjadi kata-kata terpisah. Kata-kata dipisahkan oleh perubahan dari huruf besar ke huruf kecil, dari huruf ke angka, atau dari angka/huruf ke tanda baca. Misalnya, memasukkan istilah `credsDatabase#892` pencarian mencari`creds`,, dan 892 dalam nama`Database`, deskripsi, dan kunci tag dan nilai.

Secrets Manager menghasilkan entri CloudTrail log saat Anda mencantumkan rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Anda dapat menerapkan filter berikut ke pencarian Anda:

Nama

Cocokkan awal nama rahasia; peka huruf besar/kecil. Misalnya, Nama: **`Data`** mengembalikan rahasia bernama`DatabaseSecret`, tetapi tidak `databaseSecret` atau`MyData`.

Deskripsi

Cocokkan kata-kata dalam deskripsi rahasia, tidak peka huruf besar/kecil. Misalnya, Deskripsi: **`My Description`** mencocokkan rahasia dengan deskripsi berikut:

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

Memiliki layanan

Cocokkan awal awalan ID layanan pengelolaan, tidak peka huruf besar/kecil. Misalnya, **`my-ser`** mencocokkan rahasia yang dikelola oleh layanan dengan awalan `my-serv` dan`my-service`. Untuk informasi selengkapnya, lihat Rahasia yang dikelola oleh layanan lain.

Rahasia yang direplikasi

Anda dapat memfilter rahasia utama, rahasia replika, atau rahasia yang tidak direplikasi.

Tag kunci

Cocokkan awal kunci tag; peka huruf besar/kecil. Misalnya, kunci Tag: **`Prod`** mengembalikan rahasia dengan tag `Production` dan`Prod1`, tetapi bukan rahasia dengan tag `prod` atau`1 Prod`.

Nilai tag

Cocokkan awal nilai tag; peka huruf besar/kecil. Misalnya, nilai Tag: **`Prod`** mengembalikan rahasia dengan tag `Production` dan`Prod1`, tetapi bukan rahasia dengan nilai tag `prod` atau`1 Prod`.

Secrets Manager adalah layanan regional dan hanya rahasia dalam wilayah yang dipilih dikembalikan.

AWS CLI

Example Daftar rahasia di akun Anda

`list-secrets`Contoh berikut mendapatkan daftar rahasia di akun Anda.

```
aws secretsmanager list-secrets
```

Example Filter daftar rahasia di akun Anda

`list-secrets`Contoh berikut mendapatkan daftar rahasia di akun Anda yang memiliki Test dalam nama. Pemfilteran berdasarkan nama peka huruf besar/kecil.

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

Example Temukan rahasia yang dikelola oleh AWS layanan lain

`list-secrets`Contoh berikut mendapatkan daftar rahasia yang dikelola oleh layanan. Anda menentukan layanan dengan ID. Untuk informasi selengkapnya, lihat Rahasia yang dikelola oleh layanan lain.

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

AWS SDK

Untuk menemukan rahasia dengan menggunakan salah satu AWS SDK, gunakan `ListSecrets`. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Hapus AWS Secrets Manager rahasia

Karena sifat kritis rahasia, AWS Secrets Manager sengaja membuat penghapusan rahasia menjadi sulit. Secrets Manager tidak segera menghapus rahasia. Sebagai gantinya, Secrets Manager segera membuat rahasia tidak dapat diakses dan dijadwalkan untuk dihapus setelah jendela pemulihan minimal tujuh hari. Sampai jendela pemulihan berakhir, Anda dapat memulihkan rahasia yang sebelumnya Anda hapus. Tidak ada biaya untuk rahasia yang telah Anda tandai untuk dihapus.

Anda tidak dapat menghapus rahasia utama jika direplikasi ke Wilayah lain. Pertama hapus replika, lalu hapus rahasia utama. Saat Anda menghapus replika, replika segera dihapus.

Anda tidak dapat langsung menghapus versi rahasia. Sebagai gantinya, Anda menghapus semua label pementasan dari versi menggunakan AWS CLI atau AWS SDK. Ini menandai versi sebagai usang, dan kemudian Secrets Manager dapat secara otomatis menghapus versi di latar belakang.

Jika Anda tidak tahu apakah suatu aplikasi masih menggunakan rahasia, Anda dapat membuat CloudWatch alarm Amazon untuk mengingatkan Anda tentang upaya apa pun untuk mengakses rahasia selama jendela pemulihan. Untuk informasi selengkapnya, lihat Memantau AWS Secrets Manager rahasia yang dijadwalkan untuk dihapus dengan menggunakan Amazon CloudWatch.

Untuk menghapus rahasia, Anda harus memiliki `secretsmanager:ListSecrets` dan `secretsmanager:DeleteSecret` izin.

Secrets Manager menghasilkan entri CloudTrail log saat Anda menghapus rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk menghapus rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dalam daftar rahasia, pilih rahasia yang ingin Anda hapus.
3. Di bagian Detail rahasia, pilih Tindakan, lalu pilih Hapus rahasia.
4. Dalam kotak dialog Nonaktifkan rahasia dan jadwal penghapusan, di Masa tunggu, masukkan jumlah hari untuk menunggu sebelum penghapusan menjadi permanen. Secrets Manager melampirkan bidang yang disebut `DeletionDate` dan menetapkan bidang ke tanggal dan waktu saat ini, ditambah jumlah hari yang ditentukan untuk jendela pemulihan.
5. Pilih Jadwalkan penghapusan.

Untuk melihat rahasia yang dihapus

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pada halaman Rahasia, pilih Preferensi ().
3. Di kotak dialog Preferensi, pilih Tampilkan rahasia yang dijadwalkan untuk dihapus, lalu pilih Simpan.

Untuk menghapus rahasia replika

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih rahasia utama.
3. Di bagian Rahasia Replikasi, pilih rahasia replika.
4. Dari menu Tindakan, pilih Hapus Replika.

AWS CLI

Example Hapus rahasia

`delete-secret`Contoh berikut menghapus rahasia. Anda dapat memulihkan rahasia dengan `restore-secret`sampai tanggal dan waktu di bidang DeletionDate respons. Untuk menghapus rahasia yang direplikasi ke wilayah lain, pertama-tama hapus replika dengan `remove-regions-from-replication`, lalu panggil. `delete-secret`

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

Example Hapus rahasia segera

`delete-secret`Contoh berikut menghapus rahasia segera tanpa jendela pemulihan. Anda tidak dapat memulihkan rahasia ini.

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

Example Hapus rahasia replika

`remove-regions-from-replication`Contoh berikut menghapus rahasia replika di eu-west-3. Untuk menghapus rahasia utama yang direplikasi ke wilayah lain, pertama-tama hapus replika dan kemudian panggil. `delete-secret`

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

AWS SDK

Untuk menghapus rahasia, gunakan `DeleteSecret`perintah. Untuk menghapus versi rahasia, gunakan `UpdateSecretVersionStage`perintah. Untuk menghapus replika, gunakan `StopReplicationToReplica`perintah. Untuk informasi selengkapnya, lihat the section called "AWS SDK".

Kembalikan AWS Secrets Manager rahasia

Secrets Manager menganggap rahasia yang dijadwalkan untuk penghapusan sudah usang dan Anda tidak dapat lagi mengaksesnya secara langsung. Setelah jendela pemulihan berlalu, Secrets Manager menghapus rahasia secara permanen. Setelah Secrets Manager menghapus rahasia, Anda tidak dapat memulihkannya. Sebelum akhir jendela pemulihan, Anda dapat memulihkan rahasia dan membuatnya dapat diakses lagi. Ini menghapus `DeletionDate` bidang, yang membatalkan penghapusan permanen terjadwal.

Untuk mengembalikan rahasia dan metadata di konsol, Anda harus memiliki `secretsmanager:ListSecrets` dan `secretsmanager:RestoreSecret` izin.

Secrets Manager menghasilkan entri CloudTrail log saat Anda memulihkan rahasia. Untuk informasi selengkapnya, lihat the section called "Log dengan AWS CloudTrail ".

Untuk mengembalikan rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dalam daftar rahasia, pilih rahasia yang ingin Anda pulihkan.

   Jika rahasia yang dihapus tidak muncul dalam daftar rahasia Anda, pilih Preferensi ().
   Di kotak dialog Preferensi, pilih Tampilkan rahasia yang dijadwalkan untuk dihapus, lalu pilih Simpan.
3. Pada halaman Detail rahasia, pilih Batalkan penghapusan.
4. Dalam kotak dialog Batalkan penghapusan rahasia, pilih Batalkan penghapusan.

AWS CLI

Example Kembalikan rahasia yang sebelumnya dihapus

`restore-secret`Contoh berikut mengembalikan rahasia yang sebelumnya dijadwalkan untuk dihapus.

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

AWS SDK

Untuk mengembalikan rahasia yang ditandai untuk dihapus, gunakan perintah. `RestoreSecret` Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Replikasi AWS Secrets Manager rahasia ke yang lain Wilayah AWS

Anda dapat mereplikasi rahasia Anda dalam beberapa Wilayah AWS untuk mendukung aplikasi yang tersebar di seluruh Wilayah tersebut untuk memenuhi akses Regional dan persyaratan latensi rendah. Jika nanti perlu, Anda dapat mempromosikan rahasia replika ke standalone dan kemudian mengaturnya untuk replikasi secara independen. Secrets Manager mereplikasi data rahasia terenkripsi dan metadata seperti tag dan kebijakan sumber daya di seluruh Wilayah tertentu.

ARN untuk rahasia yang direplikasi sama dengan rahasia utama kecuali untuk Wilayah, misalnya:

- Rahasia utama: `arn:aws:secretsmanager:Region1:123456789012:secret:MySecret-a1b2c3`
- Rahasia replika: `arn:aws:secretsmanager:Region2:123456789012:secret:MySecret-a1b2c3`

Untuk informasi harga untuk rahasia replika, lihat AWS Secrets ManagerHarga.

Ketika Anda menyimpan kredensyal database untuk database sumber yang direplikasi ke Wilayah lain, rahasia berisi informasi koneksi untuk database sumber. Jika Anda kemudian mereplikasi rahasia, replika adalah salinan dari rahasia sumber dan berisi informasi koneksi yang sama. Anda dapat menambahkan pasangan kunci/nilai tambahan ke rahasia untuk informasi koneksi regional.

Jika Anda mengaktifkan rotasi untuk rahasia utama Anda, Secrets Manager memutar rahasia di Wilayah utama, dan nilai rahasia baru menyebar ke semua rahasia replika terkait. Anda tidak perlu mengelola rotasi satu per satu untuk semua rahasia replika.

Anda dapat mereplikasi rahasia di semua AWS Wilayah yang diaktifkan. Namun, jika Anda menggunakan Secrets Manager di AWS Wilayah khusus seperti AWS GovCloud (US) atau Wilayah China, Anda hanya dapat mengonfigurasi rahasia dan replika di dalam AWS Wilayah khusus ini. Anda tidak dapat mereplikasi rahasia di AWS Wilayah yang diaktifkan ke Wilayah khusus atau mereplikasi rahasia dari wilayah khusus ke wilayah komersial.

Sebelum Anda dapat mereplikasi rahasia ke Wilayah lain, Anda harus mengaktifkan Wilayah itu. Untuk informasi selengkapnya, lihat Mengelola AWS Wilayah.

Dimungkinkan untuk menggunakan rahasia di beberapa Wilayah tanpa mereplikasi dengan memanggil titik akhir Secrets Manager di Wilayah tempat rahasia disimpan. Untuk daftar titik akhir, lihatthe section called “Titik akhir Secrets Manager”. Untuk menggunakan replikasi untuk meningkatkan ketahanan beban kerja Anda, lihat Arsitektur Pemulihan Bencana (DR) diAWS, Bagian I: Strategi untuk Pemulihan di Cloud.

Secrets Manager menghasilkan entri CloudTrail log saat Anda mereplikasi rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk mereplikasi rahasia ke Wilayah lain (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dari daftar rahasia, pilih rahasia Anda.
3. Pada halaman detail rahasia, pada tab Replikasi, lakukan salah satu hal berikut:
   - Jika rahasia Anda tidak direplikasi, pilih Replikasi rahasia.
   - Jika rahasia Anda direplikasi, di bagian Rahasia Replikasi, pilih Tambah Wilayah.
4. Dalam kotak dialog Tambahkan wilayah replika, lakukan hal berikut:
   a. Untuk AWSWilayah, pilih Wilayah yang ingin Anda tiru rahasianya.
   b. (Opsional) Untuk kunci Enkripsi, pilih kunci KMS untuk mengenkripsi rahasia. Kuncinya harus ada di Region replika.
   c. (Opsional) Untuk menambahkan Wilayah lain, pilih Tambahkan lebih banyak wilayah.
   d. Pilih Replikasi.

Anda kembali ke halaman detail rahasia. Di bagian rahasia Replikasi, status Replikasi ditampilkan untuk setiap Wilayah.

AWS CLI

Example Replikasi rahasia ke wilayah lain

`replicate-secret-to-regions`Contoh berikut mereplikasi rahasia eu-west-3. Replika dienkripsi dengan kunci AWS terkelola aws/secretsmanager.

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

AWS SDK

Untuk mereplikasi rahasia, gunakan `ReplicateSecretToRegions`perintah. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Memecahkan masalah

Berikut ini adalah beberapa alasan mengapa replikasi bisa gagal.

Rahasia dengan nama yang sama ada di Wilayah yang dipilih

Untuk mengatasi masalah ini, Anda dapat menimpa rahasia nama duplikat di Region replika. Coba lagi replikasi, dan kemudian di kotak dialog Retry replikasi, pilih Timpa.

Tidak ada izin yang tersedia pada kunci KMS untuk menyelesaikan replikasi

Secrets Manager pertama kali mendekripsi rahasia sebelum mengenkripsi ulang dengan kunci KMS baru di Region replika. Jika Anda tidak memiliki `kms:Decrypt` izin untuk kunci enkripsi di Wilayah utama, Anda akan mengalami kesalahan ini. Untuk mengenkripsi rahasia yang direplikasi dengan kunci KMS selain`aws/secretsmanager`, Anda perlu `kms:GenerateDataKey` dan `kms:Encrypt` ke kunci. Lihat the section called “Izin untuk kunci KMS”.

Kunci KMS dinonaktifkan atau tidak ditemukan

Jika kunci enkripsi di Wilayah utama dinonaktifkan atau dihapus, Secrets Manager tidak dapat mereplikasi rahasia. Kesalahan ini dapat terjadi bahkan jika Anda telah mengubah kunci enkripsi, jika rahasia memiliki versi berlabel khusus yang dienkripsi dengan kunci enkripsi yang dinonaktifkan atau dihapus. Untuk informasi tentang cara Secrets Manager melakukan enkripsi, lihatthe section called “Enkripsi rahasia dan dekripsi”. Untuk mengatasi masalah ini, Anda dapat membuat ulang versi rahasia sehingga Secrets Manager mengenkripsi mereka dengan kunci enkripsi saat ini. Untuk informasi selengkapnya, lihat Mengubah kunci enkripsi untuk rahasia. Kemudian coba lagi replikasi.

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

Anda belum mengaktifkan Wilayah tempat replikasi terjadi

Untuk informasi tentang cara mengaktifkan Wilayah, lihat Mengelola AWS Wilayah. dalam Panduan Referensi Manajemen AWS Akun.

Promosikan rahasia replika ke rahasia mandiri di AWS Secrets Manager

Rahasia replika adalah rahasia yang direplikasi dari primer di yang lain. Wilayah AWS Ini memiliki nilai rahasia dan metadata yang sama dengan yang utama, tetapi dapat dienkripsi dengan kunci KMS yang berbeda. Rahasia replika tidak dapat diperbarui secara independen dari rahasia utamanya, kecuali untuk kunci enkripsi. Mempromosikan rahasia replika memutus rahasia replika dari rahasia utama dan membuat rahasia replika rahasia mandiri. Perubahan pada rahasia utama tidak akan mereplikasi ke rahasia mandiri.

Anda mungkin ingin mempromosikan rahasia replika ke rahasia mandiri sebagai solusi pemulihan bencana jika rahasia utama menjadi tidak tersedia. Atau Anda mungkin ingin mempromosikan replika ke rahasia mandiri jika Anda ingin mengaktifkan rotasi untuk replika.

Jika Anda mempromosikan replika, pastikan untuk memperbarui aplikasi yang sesuai untuk menggunakan rahasia mandiri.

Secrets Manager menghasilkan entri CloudTrail log saat Anda mempromosikan rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk mempromosikan rahasia replika (konsol)

1. Masuk ke Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Arahkan ke wilayah replika.
3. Pada halaman Rahasia, pilih rahasia replika.
4. Pada halaman detail rahasia replika, pilih Promosikan ke rahasia mandiri.
5. Di kotak dialog Promosikan replika ke rahasia mandiri, masukkan Wilayah dan kemudian pilih Promosikan replika.

AWS CLI

Example Promosikan rahasia replika ke primer

`stop-replication-to-replica`Contoh berikut menghapus link antara rahasia replika ke primer. Rahasia replika dipromosikan menjadi rahasia utama di wilayah replika. Anda harus menelepon `stop-replication-to-replica`dari dalam wilayah replika.

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

AWS SDK

Untuk mempromosikan replika ke rahasia mandiri, gunakan perintah. `StopReplicationToReplica` Anda harus memanggil perintah ini dari replika Region rahasia. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Tag AWS Secrets Manager rahasia

Secrets Manager mendefinisikan tag sebagai label yang terdiri dari kunci yang Anda tentukan dan nilai opsional. Anda dapat menggunakan tag untuk memudahkan mengelola, mencari, dan memfilter rahasia dan sumber daya lainnya di AWS akun Anda. Saat Anda menandai rahasia Anda, gunakan skema penamaan standar pada semua sumber daya Anda. Untuk informasi selengkapnya, lihat whitepaper Praktik Terbaik Penandaan.

Anda dapat memberikan atau menolak akses ke rahasia dengan memeriksa tag yang dilampirkan pada rahasia. Untuk informasi selengkapnya, lihat the section called “Contoh: Kontrol akses ke rahasia menggunakan tag”.

Anda dapat menemukan rahasia dengan tag di konsol,AWS CLI, dan SDK. AWSjuga menyediakan alat Resource Groups untuk membuat konsol khusus yang menggabungkan dan mengatur sumber daya Anda berdasarkan tag mereka. Untuk menemukan rahasia dengan tag tertentu, lihatthe section called “Temukan rahasia”. Secrets Manager tidak mendukung alokasi biaya berbasis tag.

Jangan pernah menyimpan informasi sensitif untuk rahasia dalam tag.

Untuk kuota tag dan batasan penamaan, lihat Kuota layanan untuk Penandaan di panduan Referensi AWSUmum. Tag peka terhadap huruf besar dan kecil.

Secrets Manager menghasilkan entri CloudTrail log saat Anda menandai atau menghapus tag rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Untuk mengubah tag untuk rahasia Anda (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dari daftar rahasia, pilih rahasia Anda.
3. Di halaman detail rahasia, pada tab Tag, pilih Edit tag. Nama dan nilai kunci tag peka huruf besar/kecil, dan kunci tag harus unik.

AWS CLI

Example Penambahan sebuah tag ke sebuah rahasia

`tag-resource`Contoh berikut menunjukkan cara melampirkan tag dengan sintaks singkatan.

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

Example Penambahan beberapa tag ke sebuah rahasia

`tag-resource`Contoh berikut melampirkan dua tag kunci-nilai ke rahasia.

```
aws secretsmanager tag-resource \
            --secret-id MyTestSecret \
            --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",
 "Value": "SecondValue"}]'
```

Example Menghapus tag dari sebuah rahasia

`untag-resource`Contoh berikut menghapus dua tag dari rahasia. Untuk setiap tag, kunci dan nilai dihapus.

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

AWS SDK

Untuk mengubah tag untuk rahasia Anda, gunakan `TagResource`atau `UntagResource`. Untuk informasi selengkapnya, lihat the section called “AWS SDK”.

Ambil rahasia dari AWS Secrets Manager

Anda dapat mengambil rahasia Anda:

- Dalam kode
- Di layanan lain
- Di AWS CLI
- Di AWS konsol

Secrets Manager menghasilkan entri CloudTrail log saat Anda mengambil rahasia. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Dalam kode

Dalam aplikasi, Anda dapat mengambil rahasia Anda dengan menelepon `GetSecretValue` atau `BatchGetSecretValue` di salah satu AWS SDK. Sebagai contoh, lihat Mendapatkan nilai rahasia di Pustaka Contoh Kode AWS SDK. Namun, kami menyarankan Anda menyimpan nilai rahasia Anda dengan menggunakan caching sisi klien. Rahasia caching meningkatkan kecepatan dan mengurangi biaya Anda.

- Untuk aplikasi Java:
  - Jika Anda menyimpan kredensi database dalam rahasia, gunakan driver koneksi SQL Secrets Manager untuk terhubung ke database menggunakan kredensil dalam rahasia.
  - Untuk jenis rahasia lainnya, gunakan komponen caching berbasis Java Secrets Manager atau panggil SDK secara langsung. `GetSecretValue`
- Untuk aplikasi Python, gunakan komponen caching berbasis Secrets Manager Python atau panggil SDK langsung dengan atau. `get_secret_value``batch_get_secret_value`
- Untuk aplikasi.NET, gunakan komponen caching berbasis Secrets Manager .NET atau panggil SDK secara langsung dengan atau. `GetSecretValue``BatchGetSecretValue`
- Untuk aplikasi Go, gunakan komponen caching berbasis Secrets Manager Go atau panggil SDK langsung dengan atau. `GetSecretValue``BatchGetSecretValue`
- Untuk JavaScript aplikasi, hubungi SDK secara langsung dengan `getSecretValue`atau `batchGetSecretValue`.

- Untuk aplikasi PHP, hubungi SDK langsung dengan `GetSecretValue`atau `BatchGetSecretValue`.
- Untuk aplikasi Ruby, hubungi SDK langsung dengan `get_secret_value`atau. `batch_get_secret_value`
- Untuk GitHub Tindakan, lihatthe section called “GitHub Lowongan”.

Dalam sistem dan AWS layanan lain

Anda juga dapat mengambil rahasia dalam hal berikut:

- UntukAWS Batch, Anda dapat mereferensikan rahasia dalam definisi pekerjaan.
- UntukAWS CloudFormation, Anda dapat membuat rahasia dan referensi rahasia dalam CloudFormation tumpukan.
- Untuk Amazon ECS, Anda dapat mereferensikan rahasia dalam definisi wadah.
- Untuk Amazon EKS, Anda dapat menggunakan Penyedia AWS Rahasia dan Konfigurasi (ASCP) untuk memasang rahasia sebagai file di Amazon EKS.
- Untuk GitHub, Anda dapat menggunakan GitHub tindakan Secrets Manager untuk menambahkan rahasia sebagai variabel lingkungan dalam GitHub pekerjaan Anda.
- UntukAWS IoT Greengrass, Anda dapat mereferensikan rahasia dalam grup Greengrass.
- UntukAWS Lambda, Anda dapat mereferensikan rahasia dalam fungsi Lambda.
- Untuk Parameter Store, Anda dapat mereferensikan rahasia dalam parameter.

AWS CLI

Example Ambil nilai rahasia terenkripsi dari sebuah rahasia

`get-secret-value`Contoh berikut mendapatkan nilai rahasia saat ini.

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

Example Ambil nilai rahasia sebelumnya

`get-secret-value`Contoh berikut mendapatkan nilai rahasia sebelumnya.

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

Konsol AWS

Untuk mengambil rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Dalam daftar rahasia, pilih rahasia yang ingin Anda ambil.
3. Di bagian Nilai rahasia, pilih Ambil nilai rahasia.

   Secrets Manager menampilkan versi saat ini (`AWSCURRENT`) dari rahasia. Untuk melihat versi rahasia lainnya, seperti `AWSPREVIOUS` atau versi berlabel khusus, gunakan file. the section called “AWS CLI”

Ambil sekelompok rahasia dalam batch dari AWS Secrets Manager

Secrets Manager menawarkan API batch `BatchGetSecretValue`untuk mengambil sekelompok rahasia dalam satu panggilan API. Untuk memilih rahasia mana yang akan diambil, Anda dapat menentukan daftar rahasia berdasarkan nama atau ARN, atau Anda dapat menggunakan filter. Jika Secrets Manager menemukan kesalahan seperti `AccessDeniedException` saat mencoba untuk mengambil salah satu rahasia, Anda dapat melihat kesalahan `Errors` dalam respon.

Izin untuk mengambil rahasia dalam batch

Anda harus memiliki `secretsmanager:GetSecretValue` izin untuk setiap rahasia yang ingin Anda ambil. Anda juga harus memiliki `secretsmanager:BatchGetSecretValue` izin. Jika Anda menggunakan filter, Anda juga harus memilikinya`secretsmanager:ListSecrets`. Untuk contoh kebijakan izin, lihatthe section called “Izin untuk mengambil sekelompok nilai rahasia dalam batch”.

> **Important**
>
> Jika Anda memiliki kebijakan VPCE yang menolak izin untuk mengambil rahasia individu dalam grup yang Anda ambil, tidak `BatchGetSecretValue` akan mengembalikan nilai rahasia apa pun, dan itu akan mengembalikan kesalahan.

AWS CLI

Example Ambil nilai rahasia untuk sekelompok rahasia yang terdaftar dengan nama

`batch-get-secret-value`Contoh berikut mendapatkan nilai rahasia untuk tiga rahasia.

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

Example Ambil nilai rahasia untuk sekelompok rahasia yang dipilih oleh filter

`batch-get-secret-value`Contoh berikut mendapatkan nilai rahasia untuk rahasia yang memiliki tag bernama “Test”.

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

Connect ke database SQL dengan kredensi dalam rahasia AWS Secrets Manager

Dalam aplikasi Java, Anda dapat menggunakan driver Secrets Manager SQL Connection untuk terhubung ke MySQL, PostgreSQL, Oracle, MSSQLServer, Db2, dan database Redshift menggunakan kredensil yang disimpan di Secrets Manager. Setiap driver membungkus driver JDBC dasar, sehingga Anda dapat menggunakan panggilan JDBC untuk mengakses database Anda. Namun, alih-alih memberikan nama pengguna dan kata sandi untuk koneksi, Anda memberikan ID rahasia. Pengemudi memanggil Secrets Manager untuk mengambil nilai rahasia, dan kemudian menggunakan kredensil dalam rahasia untuk terhubung ke database. Driver juga menyimpan kredensialnya menggunakan pustaka caching sisi klien Java, sehingga koneksi future tidak memerlukan panggilan ke Secrets Manager. Secara default, cache diperbarui setiap jam dan juga ketika rahasia diputar. Untuk mengkonfigurasi cache, lihatthe section called “SecretCacheConfiguration”.

Anda dapat mengunduh kode sumber dari GitHub.

Untuk menggunakan driver Secrets Manager SQL Connection:

- Aplikasi Anda harus di Java 8 atau lebih tinggi.

- Rahasia Anda harus salah satu dari yang berikut:
  - Sebuah rahasia database dalam struktur JSON diharapkan. Untuk memeriksa format, di konsol Secrets Manager, lihat rahasia Anda dan pilih Ambil nilai rahasia. Atau, dalam AWS CLI, panggilan get-secret-value.
  - Rahasia yang dikelola Amazon RDS. Untuk jenis rahasia ini, Anda harus menentukan titik akhir dan port saat Anda membuat koneksi.
  - Rahasia yang dikelola Amazon Redshift. Untuk jenis rahasia ini, Anda harus menentukan titik akhir dan port saat Anda membuat koneksi.

Jika database Anda direplikasi ke Wilayah lain, untuk menyambung ke database replika di Wilayah lain, Anda menentukan titik akhir dan port regional saat Anda membuat koneksi. Anda dapat menyimpan informasi koneksi regional dalam rahasia sebagai pasangan kunci/nilai tambahan, dalam parameter Penyimpanan Parameter SSM, atau dalam konfigurasi kode Anda.

Untuk menambahkan driver ke proyek Anda, dalam file build Maven Andapom.xml, tambahkan dependensi berikut untuk driver. Untuk informasi selengkapnya, lihat Secrets Manager SQL Connection Library di situs web Maven Central Repository.

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

Pengemudi menggunakan rantai penyedia kredensi default. Jika Anda menjalankan driver di Amazon EKS, itu mungkin mengambil kredensil node yang dijalankannya alih-alih peran akun layanan. Untuk mengatasinya, tambahkan versi 1 `com.amazonaws:aws-java-sdk-sts` ke file proyek Gradle atau Maven Anda sebagai dependensi.

Untuk mengatur URL endpoint AWS PrivateLink DNS dan wilayah dalam file: `secretsmanager.properties`

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

Untuk mengganti wilayah primer, atur variabel `AWS_SECRET_JDBC_REGION` lingkungan atau buat perubahan berikut ke `secretsmanager.properties` file:

```
drivers.region = region
```

Izin yang diperlukan:

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

Untuk informasi selengkapnya, lihat Referensi izin.

Contoh:

- Membangun koneksi ke database
- Buat koneksi dengan menentukan titik akhir dan port
- Gunakan penyatuan koneksi c3p0 untuk membuat koneksi
- Gunakan penyatuan koneksi c3p0 untuk membuat koneksi dengan menentukan titik akhir dan port

Membangun koneksi ke database

Contoh berikut menunjukkan cara membuat koneksi ke database menggunakan kredensil dan informasi koneksi secara rahasia. Setelah Anda memiliki koneksi, Anda dapat menggunakan panggilan JDBC untuk mengakses database. Untuk informasi selengkapnya, lihat JDBC Basics di situs web dokumentasi 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);
```

Buat koneksi dengan menentukan titik akhir dan port

Contoh berikut menunjukkan cara membuat koneksi ke database menggunakan kredensil secara rahasia dengan titik akhir dan port yang Anda tentukan.

Rahasia terkelola Amazon RDS tidak menyertakan titik akhir dan port database. Untuk menyambung ke database menggunakan kredensi master dalam rahasia yang dikelola oleh Amazon RDS, Anda menentukannya dalam kode Anda.

Rahasia yang direplikasi ke Wilayah lain dapat meningkatkan latensi untuk koneksi ke database regional, tetapi mereka tidak mengandung informasi koneksi yang berbeda dari rahasia sumber. Setiap replika adalah salinan rahasia sumber. Untuk menyimpan informasi koneksi regional secara rahasia, tambahkan lebih banyak pasangan kunci/nilai untuk informasi titik akhir dan port untuk Wilayah.

Setelah Anda memiliki koneksi, Anda dapat menggunakan panggilan JDBC untuk mengakses database. Untuk informasi selengkapnya, lihat JDBC Basics di situs web dokumentasi 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);
```

Gunakan penyatuan koneksi c3p0 untuk membuat koneksi

Contoh berikut menunjukkan cara membuat kolam koneksi dengan `c3p0.properties` file yang menggunakan driver untuk mengambil kredensi dan informasi koneksi dari rahasia. Untuk `user` dan`jdbcUrl`, masukkan ID rahasia untuk mengkonfigurasi kumpulan koneksi. Kemudian Anda dapat mengambil koneksi dari kolam dan menggunakannya sebagai koneksi database lainnya. Untuk informasi selengkapnya, lihat JDBC Basics di situs web dokumentasi Java.

Untuk informasi lebih lanjut tentang c3p0, lihat c3p0 di situs 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
```

Gunakan penyatuan koneksi c3p0 untuk membuat koneksi dengan menentukan titik akhir dan port

Contoh berikut menunjukkan cara membuat kumpulan koneksi dengan `c3p0.properties` file yang menggunakan driver untuk mengambil kredensil secara rahasia dengan titik akhir dan port yang Anda tentukan. Kemudian Anda dapat mengambil koneksi dari kolam dan menggunakannya sebagai koneksi database lainnya. Untuk informasi selengkapnya, lihat JDBC Basics di situs web dokumentasi Java.

Rahasia terkelola Amazon RDS tidak menyertakan titik akhir dan port database. Untuk menyambung ke database menggunakan kredensi master dalam rahasia yang dikelola oleh Amazon RDS, Anda menentukannya dalam kode Anda.

Rahasia yang direplikasi ke Wilayah lain dapat meningkatkan latensi untuk koneksi ke database regional, tetapi mereka tidak mengandung informasi koneksi yang berbeda dari rahasia sumber. Setiap replika adalah salinan rahasia sumber. Untuk menyimpan informasi koneksi regional secara rahasia, tambahkan lebih banyak pasangan kunci/nilai untuk informasi titik akhir dan port untuk Wilayah.

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
```

Mengambil AWS Secrets Manager rahasia dalam aplikasi Java

Ketika Anda mengambil rahasia, Anda dapat menggunakan Secrets Manager komponen caching berbasis Java untuk cache untuk digunakan di masa mendatang. Mengambil rahasia yang di-cache lebih cepat daripada mengambilnya dari Secrets Manager. Karena ada biaya untuk memanggil Secrets Manager API, menggunakan cache dapat mengurangi biaya Anda. Untuk semua cara Anda dapat mengambil rahasia, lihatAmbil rahasia.

Kebijakan cache adalah Least Recently Used (LRU), jadi ketika cache harus membuang rahasia, ia membuang rahasia yang paling jarang digunakan. Secara default, cache menyegarkan rahasia setiap jam. Anda dapat mengonfigurasi seberapa sering rahasia disegarkan dalam cache, dan Anda dapat menghubungkan ke pengambilan rahasia untuk menambahkan lebih banyak fungsionalitas.

Cache tidak memaksa pengumpulan sampah setelah referensi cache dibebaskan. Implementasi cache tidak termasuk pembatalan cache. Implementasi cache difokuskan di sekitar cache itu sendiri, dan tidak dikeraskan atau difokuskan keamanan. Jika Anda memerlukan keamanan tambahan seperti mengenkripsi item dalam cache, gunakan antarmuka dan metode abstrak yang disediakan.

Untuk menggunakan komponen tersebut, Anda harus memiliki yang berikut:

- Java 8 atau lingkungan pengembangan yang lebih tinggi. Lihat Unduhan Java SE di situs web Oracle.

- AWSSDK 1.x untuk Java. Anda dapat menggunakan kedua versi AWS SDK for Java dalam proyek Anda. Untuk informasi selengkapnya, lihat What is What is a Java 1.x and 2.x. side-by-side

Untuk mengunduh kode sumber, lihat Secrets Manager komponen klien caching berbasis Java pada. GitHub

Untuk menambahkan komponen ke proyek Anda, dalam file pom.xml Maven Anda, sertakan dependensi berikut. Untuk informasi selengkapnya tentang Maven, lihat Panduan Memulai di situs Apache Maven Project.

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

Izin yang diperlukan:

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

Untuk informasi selengkapnya, lihat Referensi izin.

Referensi

- SecretCache
- SecretCacheConfiguration
- SecretCacheHook

Example Ambil rahasia

Contoh kode berikut menunjukkan fungsi Lambda yang mengambil string rahasia. Ini mengikuti praktik terbaik untuk membuat instance cache di luar penangan fungsi, sehingga tidak terus memanggil API jika Anda memanggil fungsi Lambda lagi.

```
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

Cache dalam memori untuk rahasia yang diminta dari Secrets Manager. Anda menggunakan the section called “getSecretString” atau the section called “getSecretBinary” untuk mengambil rahasia dari cache. Anda dapat mengkonfigurasi pengaturan cache dengan meneruskan the section called “SecretCacheConfiguration” objek di konstruktor.

Untuk informasi selengkapnya, termasuk contoh, lihatthe section called “Aplikasi Java”.

Konstruktor

`public SecretCache()`

Konstruktor default untuk `SecretCache` objek.

`public SecretCache(AWSSecretsManagerClientBuilder builder)`

Membangun cache baru menggunakan klien Secrets Manager yang dibuat menggunakan yang disediakan `AWSSecretsManagerClientBuilder`. Gunakan konstruktor ini untuk menyesuaikan klien Secrets Manager, misalnya untuk menggunakan wilayah atau titik akhir tertentu.

```
public SecretCache(AWSSecretsManager client)
```

Membangun cache rahasia baru menggunakan yang disediakan `AWSSecretsManagerClient`. Gunakan konstruktor ini untuk menyesuaikan klien Secrets Manager, misalnya untuk menggunakan wilayah atau titik akhir tertentu.

```
public SecretCache(SecretCacheConfiguration config)
```

Membangun cache rahasia baru menggunakan yang disediakanthe section called "SecretCacheConfiguration".

Metode

getSecretString

```
public String getSecretString(final String secretId)
```

Mengambil rahasia string dari Secrets Manager. Mengembalikan `String`.

getSecretBinary

```
public ByteBuffer getSecretBinary(final String secretId)
```

Mengambil rahasia biner dari Secrets Manager. Mengembalikan `ByteBuffer`.

RefreshNow

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

Memaksa cache untuk menyegarkan. Kembali `true` jika refresh selesai tanpa kesalahan, jika tidak`false`.

tutup

```
public void close()
```

Menutup cache.

SecretCacheConfiguration

Opsi konfigurasi cache untukthe section called "SecretCache", seperti ukuran cache maks dan Time to Live (TTL) untuk rahasia cache.

Konstruktor

`public SecretCacheConfiguration`

Konstruktor default untuk `SecretCacheConfiguration` objek.

Metode

GetClient

`public AWSSecretsManager getClient()`

Mengembalikan `AWSSecretsManagerClient`bahwa cache mengambil rahasia dari.

SetClient

`public void setClient(AWSSecretsManager client)`

Menetapkan `AWSSecretsManagerClient`klien tempat cache mengambil rahasia.

getCacheHook

`public SecretCacheHook getCacheHook()`

Mengembalikan the section called “SecretCacheHook” antarmuka yang digunakan untuk mengaitkan pembaruan cache.

setCacheHook

`public void setCacheHook(SecretCacheHook cacheHook)`

Mengatur the section called “SecretCacheHook” antarmuka yang digunakan untuk mengaitkan pembaruan cache.

getMaxCacheUkuran

`public int getMaxCacheSize()`

Mengembalikan ukuran cache maksimum. Defaultnya adalah 1024 rahasia.

setMaxCacheUkuran

`public void setMaxCacheSize(int maxCacheSize)`

Menetapkan ukuran cache maksimum. Defaultnya adalah 1024 rahasia.

getCacheItemTTL

```
public long getCacheItemTTL()
```

Mengembalikan TTL dalam milidetik untuk item cache. Ketika rahasia yang di-cache melebihi TTL ini, cache mengambil salinan rahasia baru dari file. `AWSSecretsManagerClient` Defaultnya adalah 1 jam dalam milidetik.

Cache menyegarkan rahasia secara serempak ketika rahasia diminta setelah TTL. Jika penyegaran sinkron gagal, cache mengembalikan rahasia basi.

setCacheItemTTL

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

Mengatur TTL dalam milidetik untuk item yang di-cache. Ketika rahasia yang di-cache melebihi TTL ini, cache mengambil salinan rahasia baru dari file. `AWSSecretsManagerClient` Defaultnya adalah 1 jam dalam milidetik.

getVersionStage

```
public String getVersionStage()
```

Mengembalikan versi rahasia yang ingin Anda cache. Untuk informasi selengkapnya, lihat What is a Secret version. Defaultnya adalah `"AWSCURRENT"`.

setVersionStage

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

Menetapkan versi rahasia yang ingin Anda cache. Untuk informasi selengkapnya, lihat What is a Secret version. Defaultnya adalah `"AWSCURRENT"`.

SecretCacheConfiguration denganKlien

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

Menetapkan `AWSSecretsManagerClient`untuk mengambil rahasia dari. Mengembalikan `SecretCacheConfiguration` objek diperbarui dengan pengaturan baru.

SecretCacheConfiguration withCacheHook

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

Mengatur antarmuka yang digunakan untuk mengaitkan cache dalam memori. Mengembalikan `SecretCacheConfiguration` objek diperbarui dengan pengaturan baru.

SecretCacheConfiguration withMaxCacheUkuran

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

Menetapkan ukuran cache maksimum. Mengembalikan `SecretCacheConfiguration` objek diperbarui dengan pengaturan baru.

SecretCacheConfiguration withCacheItemTTL

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

Mengatur TTL dalam milidetik untuk item yang di-cache. Ketika rahasia yang di-cache melebihi TTL ini, cache mengambil salinan rahasia baru dari file. `AWSSecretsManagerClient` Defaultnya adalah 1 jam dalam milidetik. Mengembalikan `SecretCacheConfiguration` objek diperbarui dengan pengaturan baru.

SecretCacheConfiguration withVersionStage

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

Menetapkan versi rahasia yang ingin Anda cache. Untuk informasi selengkapnya, lihat What is a Secret version. Mengembalikan `SecretCacheConfiguration` objek diperbarui dengan pengaturan baru.

SecretCacheHook

Antarmuka untuk menghubungkan the section called “SecretCache” ke dalam untuk melakukan tindakan pada rahasia yang disimpan dalam cache.

menempatkan

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

Siapkan objek untuk disimpan dalam cache.

Mengembalikan objek untuk menyimpan dalam cache.

memperoleh

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

Turunkan objek dari objek yang di-cache.

Mengembalikan objek untuk kembali dari cache

Ambil AWS Secrets Manager rahasia dalam aplikasi Python

Saat Anda mengambil rahasia, Anda dapat menggunakan komponen caching berbasis Secrets Manager Python untuk men-cache untuk digunakan di masa mendatang. Mengambil rahasia yang di-cache lebih cepat daripada mengambilnya dari Secrets Manager. Karena ada biaya untuk memanggil Secrets Manager API, menggunakan cache dapat mengurangi biaya Anda. Untuk semua cara Anda dapat mengambil rahasia, lihatAmbil rahasia.

Kebijakan cache adalah Least Recently Used (LRU), jadi ketika cache harus membuang rahasia, ia membuang rahasia yang paling jarang digunakan. Secara default, cache menyegarkan rahasia setiap jam. Anda dapat mengonfigurasi seberapa sering rahasia disegarkan dalam cache, dan Anda dapat menghubungkan ke pengambilan rahasia untuk menambahkan lebih banyak fungsionalitas.

Cache tidak memaksa pengumpulan sampah setelah referensi cache dibebaskan. Implementasi cache tidak termasuk pembatalan cache. Implementasi cache difokuskan di sekitar cache itu sendiri, dan tidak dikeraskan atau difokuskan keamanan. Jika Anda memerlukan keamanan tambahan seperti mengenkripsi item dalam cache, gunakan antarmuka dan metode abstrak yang disediakan.

Untuk menggunakan komponen, Anda harus memiliki yang berikut:

- Python 3.6 atau yang lebih baru.
- botocore 1.12 atau lebih tinggi. Lihat AWSSDK untuk Python dan Botocore.
- setuptools_scm 3.2 atau lebih tinggi. Lihat https://pypi.org/project/setuptools-scm/.

Untuk mengunduh kode sumber, lihat Secrets Manager Python berbasis komponen klien caching di. GitHub

Untuk menginstal komponen, gunakan perintah berikut.

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

Izin yang diperlukan:

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

Untuk informasi selengkapnya, lihat Referensi izin.

Referensi

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

Example Ambil rahasia

Contoh berikut menunjukkan cara mendapatkan rahasia bernama *`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

Cache dalam memori untuk rahasia yang diambil dari Secrets Manager. Anda menggunakan the section called “get_secret_string” atau the section called “get_secret_binary” untuk mengambil rahasia dari cache. Anda dapat mengkonfigurasi pengaturan cache dengan meneruskan the section called “SecretCacheConfig” objek di konstruktor.

Untuk informasi selengkapnya, termasuk contoh, lihatthe section called “Aplikasi Python”.

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

Ini adalah metode yang tersedia:

- get_secret_string
- get_secret_binary

get_secret_string

Mengambil nilai string rahasia.

Sintaksis Permintaan

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

Parameter

- `secret_id`(string) -- [Diperlukan] Nama atau ARN rahasia.
- `version_stage`(string) -- Versi rahasia yang ingin Anda ambil. Untuk informasi selengkapnya, lihat Versi rahasia. Defaultnya adalah 'AWSCURRENT'.

Jenis pengembalian

string

get_secret_binary

Mengambil nilai biner rahasia.

Sintaksis Permintaan

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

Parameter

- `secret_id`(string) -- [Diperlukan] Nama atau ARN rahasia.
- `version_stage`(string) -- Versi rahasia yang ingin Anda ambil. Untuk informasi selengkapnya, lihat Versi rahasia. Defaultnya adalah 'AWSCURRENT'.

Jenis pengembalian

string yang dikodekan base64

SecretCacheConfig

Opsi konfigurasi cache untuk ukuran cache maks dan Time to Live (TTL) untuk rahasia cache. the section called “SecretCache”

Parameter

`max_cache_size`(int)

Ukuran cache maksimum. Defaultnya adalah `1024` rahasia.

`exception_retry_delay_base`(int)

Jumlah detik untuk menunggu setelah pengecualian ditemukan sebelum mencoba kembali permintaan. Defaultnya adalah `1`.

`exception_retry_growth_factor`(int) pur

Faktor pertumbuhan yang digunakan untuk menghitung waktu tunggu antara percobaan ulang permintaan yang gagal. Defaultnya adalah `2`.

`exception_retry_delay_max`(int)

Jumlah waktu maksimum dalam hitungan detik untuk menunggu di antara permintaan yang gagal. Defaultnya adalah `3600`.

`default_version_stage`(str)

Versi rahasia yang ingin Anda cache. Untuk informasi selengkapnya, lihat Versi rahasia. Defaultnya adalah `'AWSCURRENT'`.

`secret_refresh_interval`(int)

Jumlah detik untuk menunggu antara menyegarkan informasi rahasia yang di-cache. Defaultnya adalah `3600`.

`secret_cache_hook` (SecretCacheHook)

Implementasi kelas `SecretCacheHook` abstrak. Nilai default-nya adalah `None`.

SecretCacheHook

Antarmuka untuk menghubungkan ke a the section called “SecretCache” untuk melakukan tindakan pada rahasia yang disimpan dalam cache.

Ini adalah metode yang tersedia:

- menempatkan
- memperoleh

menempatkan

Mempersiapkan objek untuk disimpan dalam cache.

Sintaksis Permintaan

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

Parameter

- `obj`(objek) -- [Diperlukan] Rahasia atau objek yang berisi rahasia.

Jenis pengembalian

objek

memperoleh

Mendapatkan objek dari objek yang di-cache.

Sintaksis Permintaan

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

```
)
```

Parameter

- `obj`(objek) -- [Diperlukan] Rahasia atau objek yang berisi rahasia.

Jenis pengembalian

objek

@InjectSecretString

Dekorator ini mengharapkan string ID rahasia dan the section called “SecretCache” sebagai argumen pertama dan kedua. Dekorator mengembalikan nilai string rahasia. Rahasianya harus berisi string.

```
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

Dekorator ini mengharapkan string ID rahasia dan the section called “SecretCache” sebagai argumen pertama dan kedua. Argumen yang tersisa memetakan parameter dari fungsi yang dibungkus ke kunci JSON dalam rahasia. Rahasia harus berisi string dalam struktur JSON.

Untuk rahasia yang berisi JSON ini:

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

Contoh berikut menunjukkan cara mengekstrak nilai JSON for `username` and `password` from the secret.

```
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')
```

Ambil AWS Secrets Manager rahasia di aplikasi.NET

Saat Anda mengambil rahasia, Anda dapat menggunakan komponen caching berbasis Secrets Manager .net untuk men-cache untuk digunakan di masa mendatang. Mengambil rahasia yang di-cache lebih cepat daripada mengambilnya dari Secrets Manager. Karena ada biaya untuk memanggil Secrets Manager API, menggunakan cache dapat mengurangi biaya Anda. Untuk semua cara Anda dapat mengambil rahasia, lihatAmbil rahasia.

Kebijakan cache adalah Least Recently Used (LRU), jadi ketika cache harus membuang rahasia, ia membuang rahasia yang paling jarang digunakan. Secara default, cache menyegarkan rahasia setiap jam. Anda dapat mengonfigurasi seberapa sering rahasia disegarkan dalam cache, dan Anda dapat menghubungkan ke pengambilan rahasia untuk menambahkan lebih banyak fungsionalitas.

Cache tidak memaksa pengumpulan sampah setelah referensi cache dibebaskan. Implementasi cache tidak termasuk pembatalan cache. Implementasi cache difokuskan di sekitar cache itu sendiri, dan tidak dikeraskan atau difokuskan keamanan. Jika Anda memerlukan keamanan tambahan seperti mengenkripsi item dalam cache, gunakan antarmuka dan metode abstrak yang disediakan.

Untuk menggunakan komponen, Anda harus memiliki yang berikut:

- .NET Framework 4.6.2 atau lebih tinggi, atau .NET Standard 2.0 atau lebih tinggi. Lihat Mengunduh.NET di situs web Microsoft .NET.
- AWSSDK for .NET. Lihat the section called “AWS SDK”.

Untuk mengunduh kode sumber, lihat Caching client untuk.NET di GitHub.

Untuk menggunakan cache, pertama buat instance, lalu ambil rahasia Anda dengan menggunakan atau. `GetSecretString` `GetSecretBinary` Pada pengambilan berturut-turut, cache mengembalikan salinan rahasia yang di-cache.

Untuk mendapatkan paket caching

- Lakukan salah satu dari berikut:
  - Jalankan perintah.NET CLI berikut di direktori proyek Anda.

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

  - Tambahkan referensi paket berikut ke `.csproj` file Anda.

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

Izin yang diperlukan:

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

Untuk informasi selengkapnya, lihat Referensi izin.

Referensi

- SecretsManagerCache
- SecretCacheConfiguration
- Aku SecretCacheHook

Example Mengambil rahasia

Contoh kode berikut menunjukkan metode yang mengambil rahasia bernama *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 Konfigurasikan durasi refresh cache time to live (TTL)

Contoh kode berikut menunjukkan metode yang mengambil rahasia bernama *MySecret*dan menetapkan durasi penyegaran cache TTL menjadi 24 jam.

```
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

Cache dalam memori untuk rahasia yang diminta dari Secrets Manager. Anda menggunakan the section called “GetSecretString” atau the section called “GetSecretBinary” untuk mengambil rahasia dari cache. Anda dapat mengkonfigurasi pengaturan cache dengan meneruskan the section called “SecretCacheConfiguration” objek di konstruktor.

Untuk informasi selengkapnya, termasuk contoh, lihatthe section called “Aplikasi .NET”.

Konstruktor

```
public SecretsManagerCache()
```

Konstruktor default untuk `SecretsManagerCache` objek.

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

Membangun cache baru menggunakan klien Secrets Manager yang dibuat menggunakan yang disediakan AmazonSecretsManagerClient. Gunakan konstruktor ini untuk menyesuaikan klien Secrets Manager, misalnya untuk menggunakan wilayah atau titik akhir tertentu.

Parameter

Rahasia Manajer

AmazonSecretsManagerClientUntuk mengambil rahasia dari.

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

Membangun cache rahasia baru menggunakan yang disediakanthe section called “SecretCacheConfiguration”. Gunakan konstruktor ini untuk mengkonfigurasi cache, misalnya jumlah rahasia untuk cache dan seberapa sering itu menyegarkan.

Parameter

config

A the section called “SecretCacheConfiguration” yang berisi informasi konfigurasi untuk cache.

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

Membangun cache baru menggunakan klien Secrets Manager yang dibuat menggunakan yang disediakan AmazonSecretsManagerClientdan file. the section called “SecretCacheConfiguration”

Gunakan konstruktor ini untuk menyesuaikan klien Secrets Manager, misalnya untuk menggunakan wilayah atau titik akhir tertentu serta mengkonfigurasi cache, misalnya jumlah rahasia untuk cache dan seberapa sering itu menyegarkan.

Parameter

Rahasia Manajer

AmazonSecretsManagerClientUntuk mengambil rahasia dari.

config

A the section called “SecretCacheConfiguration” yang berisi informasi konfigurasi untuk cache.

Metode

GetSecretString

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

Mengambil rahasia string dari Secrets Manager.

Parameter

secretID

ARN atau nama rahasia untuk mengambil.

GetSecretBinary

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

Mengambil rahasia biner dari Secrets Manager.

Parameter

secretID

ARN atau nama rahasia untuk mengambil.

RefreshNowAsync

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

Meminta nilai rahasia dari Secrets Manager dan memperbarui cache dengan perubahan apa pun. Jika tidak ada entri cache yang ada, buat yang baru. Kembali `true` jika refresh berhasil.

Parameter

secretID

ARN atau nama rahasia untuk mengambil.

GetCachedSecret

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

Mengembalikan entri cache untuk rahasia tertentu jika ada dalam cache. Jika tidak, mengambil rahasia dari Secrets Manager dan menciptakan cache baru.

Parameter

secretID

ARN atau nama rahasia untuk mengambil.

SecretCacheConfiguration

Opsi konfigurasi cache untukthe section called “SecretsManagerCache”, seperti ukuran cache maksimum dan Time to Live (TTL) untuk rahasia cache.

Properti

CacheItemTTL

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

TTL item cache dalam milidetik. Standarnya adalah `3600000` ms atau 1 jam. Maksimum adalah `4294967295` ms, yaitu sekitar 49,7 hari.

MaxCacheSize

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

Ukuran cache maksimum. Defaultnya adalah 1024 rahasia. Maksimum adalah 65.535.

VersionStage

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

Versi rahasia yang ingin Anda cache. Untuk informasi selengkapnya, lihat Versi rahasia. Defaultnya adalah `"AWSCURRENT"`.

Klien

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

AmazonSecretsManagerClientUntuk mengambil rahasia dari. Jika yan`ull`, cache membuat instance klien baru. Defaultnya adalah `null`.

CacheHook

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

Athe section called “Aku SecretCacheHook”.

Aku SecretCacheHook

Antarmuka untuk menghubungkan the section called “SecretsManagerCache” ke dalam untuk melakukan tindakan pada rahasia yang disimpan dalam cache.

Metode

Masukan

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

Siapkan objek untuk disimpan dalam cache.

Mengembalikan objek untuk menyimpan dalam cache.

Dapatkan

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

Turunkan objek dari objek yang di-cache.

Mengembalikan objek untuk kembali dari cache

Ambil AWS Secrets Manager rahasia di aplikasi Go

Saat Anda mengambil rahasia, Anda dapat menggunakan komponen caching berbasis Secrets Manager Go untuk men-cache untuk digunakan di masa mendatang. Mengambil rahasia yang di-cache lebih cepat daripada mengambilnya dari Secrets Manager. Karena ada biaya untuk memanggil Secrets Manager API, menggunakan cache dapat mengurangi biaya Anda. Untuk semua cara Anda dapat mengambil rahasia, lihatAmbil rahasia.

Kebijakan cache adalah Least Recently Used (LRU), jadi ketika cache harus membuang rahasia, ia membuang rahasia yang paling jarang digunakan. Secara default, cache menyegarkan rahasia setiap jam. Anda dapat mengonfigurasi seberapa sering rahasia disegarkan dalam cache, dan Anda dapat menghubungkan ke pengambilan rahasia untuk menambahkan lebih banyak fungsionalitas.

Cache tidak memaksa pengumpulan sampah setelah referensi cache dibebaskan. Implementasi cache tidak termasuk pembatalan cache. Implementasi cache difokuskan di sekitar cache itu sendiri, dan tidak dikeraskan atau difokuskan keamanan. Jika Anda memerlukan keamanan tambahan seperti mengenkripsi item dalam cache, gunakan antarmuka dan metode abstrak yang disediakan.

Untuk menggunakan komponen, Anda harus memiliki yang berikut:

- AWSSDK for Go. Lihat the section called “AWS SDK”.

Untuk mengunduh kode sumber, lihat Secrets Manager Go caching client di GitHub.

Untuk menyiapkan lingkungan pengembangan Go, lihat Golang Memulai di situs web Go Programming Language.

Izin yang diperlukan:

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

Untuk informasi selengkapnya, lihat Referensi izin.

Referensi

- jenis Cache
- jenis CacheConfig
- jenis CacheHook

Example Ambil rahasia

Contoh kode berikut menunjukkan fungsi Lambda yang mengambil rahasia.

```
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)
}
```

jenis Cache

Cache dalam memori untuk rahasia yang diminta dari Secrets Manager. Anda menggunakan the section called “GetSecretString” atau the section called “GetSecretBinary” untuk mengambil rahasia dari cache.

Contoh berikut menunjukkan cara mengonfigurasi pengaturan.

```
// 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 },
)
```

Untuk informasi selengkapnya, termasuk contoh, lihatthe section called “Aplikasi Go”.

Metode

Baru

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

Baru membangun cache rahasia menggunakan opsi fungsional, menggunakan default sebaliknya. Menginisialisasi SecretsManager Klien dari sesi baru. Menginisialisasi CacheConfig ke nilai default. Menginisialisasi cache LRU dengan ukuran maks default.

GetSecretString

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

GetSecretString mendapatkan nilai string rahasia dari cache untuk ID rahasia yang diberikan. Mengembalikan sengatan rahasia dan kesalahan jika operasi gagal.

GetSecretStringWithStage

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

GetSecretStringWithStage mendapatkan nilai string rahasia dari cache untuk ID rahasia dan tahap versi yang diberikan. Mengembalikan sengatan rahasia dan kesalahan jika operasi gagal.

GetSecretBinary

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

GetSecretBinary mendapatkan nilai biner rahasia dari cache untuk ID rahasia yang diberikan. Mengembalikan biner rahasia dan kesalahan jika operasi gagal.

GetSecretBinaryWithStage

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

GetSecretBinaryWithStage mendapatkan nilai biner rahasia dari cache untuk ID rahasia dan tahap versi yang diberikan. Mengembalikan biner rahasia dan kesalahan jika operasi gagal.

jenis CacheConfig

Opsi konfigurasi cache untuk Cache, seperti ukuran cache maksimum, tahap versi default, dan Time to Live (TTL) untuk rahasia cache.

```
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
}
```

jenis CacheHook

Antarmuka untuk menghubungkan ke Cache untuk melakukan tindakan pada rahasia yang disimpan dalam cache.

Metode

Masukan

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

Mempersiapkan objek untuk disimpan dalam cache.

Dapatkan

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

Mendapatkan objek dari objek yang di-cache.

Gunakan AWS Secrets Manager rahasia di AWS Batch

AWS Batchmembantu Anda menjalankan beban kerja komputasi batch di. AWS Cloud Dengan AWS Batch, Anda dapat menyuntikkan data sensitif ke dalam tugas Anda dengan menyimpan data sensitif Anda di rahasia AWS Secrets Manager, lalu mereferensikannya dalam ketentuan tugas Anda. Untuk informasi selengkapnya, lihat Menentukan data sensitif menggunakan Secrets Manager.

Ambil AWS Secrets Manager rahasia di sumber daya AWS CloudFormation

DenganAWS CloudFormation, Anda dapat mengambil rahasia untuk digunakan di AWS CloudFormation sumber lain. Skenario umum adalah pertama-tama membuat rahasia dengan kata sandi yang dihasilkan oleh Secrets Manager, dan kemudian mengambil nama pengguna dan kata sandi dari rahasia untuk digunakan sebagai kredensyal untuk database baru. Untuk informasi tentang membuat rahasia denganAWS CloudFormation, lihatAWS CloudFormation.

Untuk mengambil rahasia dalam AWS CloudFormation template, Anda menggunakan referensi dinamis. Saat Anda membuat tumpukan, referensi dinamis menarik nilai rahasia ke AWS CloudFormation sumber daya, jadi Anda tidak perlu melakukan hardcode informasi rahasia. Sebaliknya, Anda merujuk ke rahasia dengan nama atau ARN. Anda dapat menggunakan referensi dinamis untuk rahasia di properti sumber daya apa pun. Anda tidak dapat menggunakan referensi dinamis untuk rahasia dalam metadata sumber daya seperti `AWS::CloudFormation::Init`karena itu akan membuat nilai rahasia terlihat di konsol.

Referensi dinamis untuk rahasia memiliki pola berikut:

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

rahasia-id

Nama atau ARN rahasianya. Untuk mengakses rahasia di AWS akun Anda, Anda dapat menggunakan nama rahasia. Untuk mengakses rahasia di AWS akun yang berbeda, gunakan ARN rahasia.

json-key (Opsional)

Nama kunci dari pasangan kunci-nilai yang nilainya ingin Anda ambil. Jika Anda tidak menentukan `json-key`, AWS CloudFormation mengambil seluruh teks rahasia. Segmen ini mungkin tidak memasukkan karakter titik dua (`:`).

tahap versi (Opsional)

Versi rahasia untuk digunakan. Secrets Manager menggunakan label pementasan untuk melacak versi yang berbeda selama proses rotasi. Jika Anda menggunakan `version-stage` maka jangan tentukan `version-id`. Jika Anda tidak menentukan salah satu `version-stage` atau`version-id`, maka defaultnya adalah `AWSCURRENT` versinya. Segmen ini mungkin tidak memasukkan karakter titik dua (`:`).

version-id (Opsional)

Pengidentifikasi unik dari versi rahasia yang akan digunakan. Jika Anda menentukan `version-id`, jangan tentukan `version-stage`. Jika Anda tidak menentukan salah satu `version-stage` atau`version-id`, maka defaultnya adalah `AWSCURRENT` versinya. Segmen ini mungkin tidak memasukkan karakter titik dua (`:`).

Untuk informasi selengkapnya, lihat Menggunakan referensi dinamis untuk menentukan rahasia Secrets Manager.

> **Note**
>
> Jangan membuat referensi dinamis menggunakan garis miring terbalik (`\`) sebagai nilai akhir. AWS CloudFormationtidak dapat menyelesaikan referensi tersebut, yang menyebabkan kegagalan sumber daya.

Gunakan AWS Secrets Manager rahasia di Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) adalah layanan orkestrasi kontainer terkelola penuh yang membantu Anda menerapkan, mengelola, dan menskalakan aplikasi kontainer dengan mudah. Anda dapat menyuntikkan data sensitif ke dalam kontainer dengan mereferensikan rahasia Secrets Manager. Untuk informasi selengkapnya, lihat halaman berikut di Panduan Pengembang Layanan Kontainer Elastis Amazon:

- Tutorial: Menentukan data sensitif menggunakan rahasia Secrets Manager
- Ambil rahasia secara terprogram melalui aplikasi Anda
- Ambil rahasia melalui variabel lingkungan
- Ambil rahasia untuk konfigurasi logging

Gunakan AWS Secrets Manager rahasia di Amazon Elastic Kubernetes Service

Untuk menampilkan rahasia dari Secrets Manager sebagai file yang dipasang di pod Amazon EKS, Anda dapat menggunakan AWS Secrets and Configuration Provider (ASCP) untuk Kubernetes Secrets Store CSI Driver. ASCP bekerja dengan Amazon Elastic Kubernetes Service (Amazon EKS) 1.17+ yang menjalankan grup node Amazon EC2. AWS Fargate grup node tidak didukung. Dengan ASCP, Anda dapat menyimpan dan mengelola rahasia Anda di Secrets Manager dan kemudian mengambilnya melalui beban kerja Anda yang berjalan di Amazon EKS. Jika rahasia Anda berisi beberapa pasangan kunci/nilai dalam format JSON, Anda dapat memilih mana yang akan dipasang di Amazon EKS. ASCP menggunakan sintaks JMESPath untuk menanyakan pasangan kunci/nilai dalam rahasia Anda. ASCP juga bekerja dengan parameter Parameter Store.

Anda menggunakan peran dan kebijakan IAM untuk memberikan akses ke rahasia Anda ke pod Amazon EKS tertentu dalam sebuah klaster.

Untuk menjelaskan file mana yang akan dibuat di pod Amazon EKS dan rahasia mana yang dimasukkan ke dalamnya, Anda membuat file the section called “`SecretProviderClass`” YAMAL. `SecretProviderClass`Harus berada di namespace yang sama dengan pod Amazon EKS yang direferensikannya.

Jika Anda menggunakan kluster Amazon EKS pribadi, pastikan VPC tempat klaster berada memiliki titik akhir Secrets Manager. Secrets Store CSI Driver menggunakan endpoint untuk melakukan panggilan ke Secrets Manager. Untuk informasi tentang membuat endpoint di VPC, lihat. Titik akhir VPC

Jika Anda menggunakan rotasi otomatis Secrets Manager untuk rahasia Anda, Anda juga dapat menggunakan fitur reconciler rotasi Driver Secrets Store CSI untuk memastikan Anda mengambil rahasia terbaru dari Secrets Manager. Untuk informasi selengkapnya, lihat Rotasi otomatis konten yang dipasang dan Rahasia Kubernetes yang disinkronkan.

Untuk tutorial tentang cara menggunakan ASCP, lihatthe section called “Tutorial”.

Instal ASCP

ASCP tersedia GitHub di repositori secrets-store-csi-provider-aws. Repo juga berisi contoh file YAMAL untuk membuat dan memasang rahasia.

Untuk menginstal ASCP

- Untuk menginstal Secrets Store CSI Driver dan ASCP dengan menggunakan Helm, gunakan perintah berikut. Untuk memastikan repo menunjuk ke bagan terbaru, gunakan `helm repo update`.

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

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

  Atau, untuk menginstal dengan menggunakan file YAMAL di direktori deployment, gunakan perintah berikut.

```
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
```

Langkah 1: Siapkan kontrol akses

Untuk memberikan akses pod Amazon EKS ke rahasia di Secrets Manager, pertama-tama Anda membuat kebijakan `secretsmanager:DescribeSecret` izin yang memberikan `secretsmanager:GetSecretValue` dan mengizinkan rahasia yang perlu diakses oleh pod. Untuk kebijakan-kebijakan contoh, lihat Contoh kebijakan izin.

Kemudian Anda membuat peran IAM untuk akun layanan dan melampirkan kebijakan ke dalamnya. Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan.

ASCP mengambil identitas pod dan menukarnya dengan peran IAM. ASCP mengasumsikan peran IAM dari pod, yang memberinya akses ke rahasia yang Anda otorisasi. Kontainer lain tidak dapat mengakses rahasia kecuali Anda juga mengaitkannya dengan peran IAM.

Jika Anda menggunakan kluster Amazon EKS pribadi, pastikan VPC tempat klaster berada memiliki AWS STS titik akhir. Untuk informasi tentang membuat titik akhir, lihat Titik akhir VPC Antarmuka diAWS Identity and Access Management Panduan Pengguna.

Langkah 2: Identifikasi rahasia mana yang akan dipasang

Untuk menentukan rahasia mana yang dipasang ASCP di Amazon EKS sebagai file di sistem file, Anda membuat file YAMAL. `SecretProviderClass` `SecretProviderClass`YAMM mencantumkan rahasia untuk dipasang dan nama file untuk dipasang sebagai. `SecretProviderClass`Harus berada di namespace yang sama dengan pod Amazon EKS yang direferensikannya.

Contoh berikut menunjukkan cara menggunakan `SecretProviderClass` untuk mendeskripsikan rahasia yang ingin Anda pasang dan apa nama file yang dipasang di pod Amazon EKS. Untuk informasi selengkapnya, lihat the section called "`SecretProviderClass`".

Contoh:

- Contoh: Pasang rahasia dengan nama atau ARN
- Contoh: Pasang pasangan kunci/nilai dari rahasia
- Contoh: Tentukan Wilayah failover untuk rahasia Multi-wilayah
- Contoh: Pilih rahasia failover untuk dipasang

Contoh: Pasang rahasia dengan nama atau ARN

Contoh berikut menunjukkan `SecretProviderClass` yang memasang tiga file di Amazon EKS:

1. Rahasia yang ditentukan oleh ARN lengkap.
2. Rahasia yang ditentukan oleh nama.
3. Versi rahasia tertentu.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
```

```
    objects: |
        - objectName: "arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret2-
d4e5f6"
        - objectName: "MySecret3"
          objectType: "secretsmanager"
        - objectName: "MySecret4"
          objectType: "secretsmanager"
          objectVersionLabel: "AWSCURRENT"
```

Contoh: Pasang pasangan kunci/nilai dari rahasia

Contoh berikut menunjukkan `SecretProviderClass` yang memasang tiga file di Amazon EKS:

1. Rahasia yang ditentukan oleh ARN lengkap.
2. Pasangan `username` kunci/nilai dari rahasia yang sama.
3. Pasangan `password` kunci/nilai dari rahasia yang sama.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
    objects: |
      - objectName: "arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret-
a1b2c3"
        jmesPath:
            - path: username
              objectAlias: dbusername
            - path: password
              objectAlias: dbpassword
```

Contoh: Tentukan Wilayah failover untuk rahasia Multi-wilayah

Untuk menyediakan ketersediaan selama pemadaman konektivitas atau untuk konfigurasi pemulihan bencana, ASCP mendukung fitur failover otomatis untuk mengambil rahasia dari wilayah sekunder.

Contoh berikut menunjukkan `SecretProviderClass` yang mengambil rahasia yang direplikasi ke beberapa Wilayah. Dalam contoh ini, ASCP mencoba untuk mengambil rahasia dari keduanya

danus-east-1. us-east-2 Jika salah satu Wilayah mengembalikan kesalahan 4xx, misalnya untuk masalah otentikasi, ASCP tidak memasang salah satu rahasia. Jika rahasia berhasil diambilus-east-1, maka ASCP memasang nilai rahasia itu. Jika rahasia tidak berhasil diambil darius-east-1, tetapi berhasil diambil darius-east-2, maka ASCP memasang nilai rahasia itu.

```
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"
```

Contoh: Pilih rahasia failover untuk dipasang

Contoh berikut menunjukkan SecretProviderClass yang menentukan rahasia mana yang akan dipasang jika terjadi failover. Rahasia failover bukanlah replika. Dalam contoh ini, ASCP mencoba untuk mengambil dua rahasia yang ditentukan oleh. objectName Jika salah satu mengembalikan kesalahan 4xx, misalnya untuk masalah otentikasi, ASCP tidak memasang salah satu rahasia. Jika rahasia berhasil diambilus-east-1, maka ASCP memasang nilai rahasia itu. Jika rahasia tidak berhasil diambil darius-east-1, tetapi berhasil diambil darius-east-2, maka ASCP memasang nilai rahasia itu. File yang dipasang di Amazon EKS diberi namaMyMountedSecret.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: aws-secrets
spec:
  provider: aws
  parameters:
    region: us-east-1
    failoverRegion: us-east-2
    objects: |
      - objectName: "arn:aws:secretsmanager:us-east-1:111122223333:secret:MySecret-
a1b2c3"
        objectAlias: "MyMountedSecret"
        failoverObject:
```

```
          - objectName: "arn:aws:secretsmanager:us-
east-2:111122223333:secret:MyFailoverSecret-d4e5f6"
```

Pemecahan Masalah

Anda dapat melihat sebagian besar kesalahan dengan menjelaskan penerapan pod.

Untuk melihat pesan galat untuk penampung Anda

1. Dapatkan daftar nama pod dengan perintah berikut. Jika Anda tidak menggunakan namespace default, gunakan. `-n <NAMESPACE>`

   ```
   kubectl get pods
   ```

2. Untuk mendeskripsikan pod, dalam perintah berikut, <PODID>gunakan ID pod dari pod yang Anda temukan di langkah sebelumnya. Jika Anda tidak menggunakan namespace default, gunakan. `-n <NAMESPACE>`

   ```
   kubectl describe pod/<PODID>
   ```

Untuk melihat kesalahan untuk ASCP

- Untuk menemukan informasi selengkapnya di log penyedia, dalam perintah berikut, <PODID>gunakan ID pod csi-secrets-store-provider-aws.

  ```
  kubectl -n kube-system get pods
  kubectl -n kube-system logs pod/<PODID>
  ```

Tutorial: Membuat dan memasang AWS Secrets Manager rahasia di pod Amazon EKS

Dalam tutorial ini, Anda membuat contoh rahasia di Secrets Manager, dan kemudian Anda memasang rahasia di pod Amazon EKS dan menerapkannya.

Sebelum Anda mulai, instal ASCP:the section called “Instal ASCP”.

Untuk membuat dan memasang rahasia

1. Atur Wilayah AWS dan nama cluster Anda sebagai variabel shell sehingga Anda dapat menggunakannya dalam perintah bash. Untuk<REGION>, masukkan Wilayah AWS tempat kluster Amazon EKS Anda berjalan. Untuk<CLUSTERNAME>, masukkan nama cluster Anda.

```
REGION=<REGION>
CLUSTERNAME=<CLUSTERNAME>
```

2. Buat rahasia tes. Untuk informasi selengkapnya, lihat Buat dan kelola rahasia.

```
aws --region "$REGION" secretsmanager  create-secret --name MySecret --secret-
string '{"username":"lijuan", "password":"hunter2"}'
```

3. Buat kebijakan sumber daya untuk pod yang membatasi aksesnya ke rahasia yang Anda buat di langkah sebelumnya. Untuk<SECRETARN>, gunakan ARN rahasia. Simpan ARN kebijakan dalam variabel shell.

```
POLICY_ARN=$(aws --region "$REGION" --query Policy.Arn --output text iam create-
policy --policy-name nginx-deployment-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [ {
        "Effect": "Allow",
        "Action": ["secretsmanager:GetSecretValue",
 "secretsmanager:DescribeSecret"],
        "Resource": ["<SECRETARN>"]
    } ]
}')
```

4. Buat penyedia IAM OIDC untuk cluster jika Anda belum memilikinya. Untuk informasi selengkapnya, lihat Membuat penyedia IAM OIDC untuk klaster Anda.

```
eksctl utils associate-iam-oidc-provider --region="$REGION" --
cluster="$CLUSTERNAME" --approve # Only run this once
```

5. Buat akun layanan yang digunakan pod dan kaitkan kebijakan sumber daya yang Anda buat di langkah 3 dengan akun layanan tersebut. Untuk tutorial ini, untuk nama akun layanan, Anda gunakan nginx-deployment-sa. Untuk informasi selengkapnya, lihat Membuat peran IAM untuk akun layanan.

```
eksctl create iamserviceaccount --name nginx-deployment-sa --region="$REGION" --
cluster "$CLUSTERNAME" --attach-policy-arn "$POLICY_ARN" --approve --override-
existing-serviceaccounts
```

6. Buat `SecretProviderClass` untuk menentukan rahasia mana yang akan dipasang di pod. Perintah berikut digunakan `ExampleSecretProviderClass.yaml` dalam direktori contoh GitHub repo ASCP untuk me-mount rahasia yang Anda buat di langkah 2. Untuk informasi tentang membuat milik Anda sendiri`SecretProviderClass`, lihatthe section called "`SecretProviderClass`".

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

7. Terapkan pod Anda. Perintah berikut digunakan `ExampleDeployment.yaml` dalam direktori contoh GitHub repo ASCP untuk me-mount rahasia di `/mnt/secrets-store` dalam pod.

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

8. Untuk memverifikasi rahasia telah dipasang dengan benar, gunakan perintah berikut dan konfirmasikan bahwa nilai rahasia Anda muncul.

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

Nilai rahasia muncul.

```
{"username":"lijuan", "password":"hunter2"}
```

`SecretProviderClass`

Anda menggunakan YAMAL untuk menjelaskan rahasia mana yang akan dipasang di Amazon EKS menggunakan ASCP. Sebagai contoh, lihat Identifikasi rahasia mana yang akan dipasang.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: <NAME>
```

```
spec:
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    objects:
```

Bidang `parameters` berisi rincian permintaan pemasangan:

region

(Opsional) Wilayah AWS Rahasianya. Jika Anda tidak menggunakan bidang ini, ASCP mencari Region dari anotasi pada node. Pencarian ini menambahkan overhead ke permintaan mount, jadi sebaiknya Anda menyediakan Region untuk cluster yang menggunakan pod dalam jumlah besar.

Jika Anda juga menentukan`failoverRegion`, ASCP mencoba untuk mengambil rahasia dari kedua Wilayah. Jika salah satu Wilayah mengembalikan kesalahan 4xx, misalnya untuk masalah otentikasi, ASCP tidak memasang salah satu rahasia. Jika rahasia berhasil diambil`region`, maka ASCP memasang nilai rahasia itu. Jika rahasia tidak berhasil diambil dari`region`, tetapi berhasil diambil dari`failoverRegion`, maka ASCP memasang nilai rahasia itu.

FailOverRegion

(Opsional) Jika Anda menyertakan bidang ini, ASCP mencoba mengambil rahasia dari Wilayah yang ditentukan dalam `region` dan bidang ini. Jika salah satu Wilayah mengembalikan kesalahan 4xx, misalnya untuk masalah otentikasi, ASCP tidak memasang salah satu rahasia. Jika rahasia berhasil diambil`region`, maka ASCP memasang nilai rahasia itu. Jika rahasia tidak berhasil diambil dari`region`, tetapi berhasil diambil dari`failoverRegion`, maka ASCP memasang nilai rahasia itu. Untuk contoh cara menggunakan bidang ini, lihatTentukan Wilayah failover untuk rahasia Multi-wilayah.

PathTranslation

(Opsional) Karakter substitusi tunggal untuk digunakan jika nama file di Amazon EKS akan berisi karakter pemisah jalur, seperti garis miring (/) di Linux. ASCP tidak dapat membuat file yang dipasang yang berisi karakter pemisah jalur. Sebagai gantinya, ASCP menggantikan karakter pemisah jalur dengan karakter yang berbeda. Jika Anda tidak menggunakan bidang ini, karakter pengganti adalah garis bawah (_), jadi misalnya, `My/Path/Secret` dipasang sebagai. `My_Path_Secret`

Untuk mencegah substitusi karakter, masukkan string`False`.

objek

String yang berisi deklarasi YAMAL tentang rahasia yang akan dipasang. Sebaiknya gunakan karakter string atau pipe (|) multi-line YAMM.

objectName

Nama atau ARN lengkap rahasianya. Jika Anda menggunakan ARN, Anda dapat menghilangkannya. `objectType` Bidang ini menjadi nama file rahasia di pod Amazon EKS kecuali Anda menentukan`objectAlias`. Jika Anda menggunakan ARN, Wilayah di ARN harus cocok dengan bidang. `region` Jika Anda menyertakan `afailoverRegion`, bidang ini mewakili primer`objectName`.

objectType

Diperlukan jika Anda tidak menggunakan Secrets Manager ARN untuk. `objectName` Bisa salah satu `secretsmanager` atau`ssmparameter`.

Objectalias

(Opsional) Nama file rahasia di pod Amazon EKS. Jika Anda tidak menentukan bidang ini, `objectName` muncul sebagai nama file.

ObjectVersion

(Opsional) ID versi rahasia. Tidak disarankan karena Anda harus memperbarui ID versi setiap kali Anda memperbarui rahasia. Secara default versi terbaru digunakan. Jika Anda menyertakan `afailoverRegion`, bidang ini mewakili primer`objectVersion`.

objectVersionLabel

(Opsional) Alias untuk versi. Defaultnya adalah versi terbaru AWSCURRENT. Untuk informasi selengkapnya, lihat the section called “Versi”. Jika Anda menyertakan `afailoverRegion`, bidang ini mewakili primer`objectVersionLabel`.

JMESPath

(Opsional) Peta kunci dalam rahasia file yang akan dipasang di Amazon EKS. Untuk menggunakan bidang ini, nilai rahasia Anda harus dalam format JSON. Jika Anda menggunakan bidang ini, Anda harus menyertakan subbidang `path` dan`objectAlias`.

path

Kunci dari pasangan kunci/nilai di JSON dari nilai rahasia. Jika bidang berisi tanda hubung, gunakan tanda kutip tunggal untuk menghindarinya, misalnya: `path: '"hyphenated-path"'`

Objectalias

Nama file yang akan dipasang di pod Amazon EKS. Jika bidang berisi tanda hubung, gunakan tanda kutip tunggal untuk menghindarinya, misalnya: `objectAlias: '"hyphenated-alias"'`

FailOverObject

(Opsional) Jika Anda menentukan bidang ini, ASCP mencoba untuk mengambil kedua rahasia yang ditentukan dalam primer `objectName` dan rahasia yang ditentukan dalam sub-bidang. `failoverObject` `objectName` Jika salah satu mengembalikan kesalahan 4xx, misalnya untuk masalah otentikasi, ASCP tidak memasang salah satu rahasia. Jika rahasia berhasil diambil dari primer`objectName`, maka ASCP memasang nilai rahasia itu. Jika rahasia tidak berhasil diambil dari primer`objectName`, tetapi berhasil diambil dari failover`objectName`, maka ASCP memasang nilai rahasia itu. Jika Anda menyertakan bidang ini, Anda harus menyertakan bidang tersebut`objectAlias`. Untuk contoh cara menggunakan bidang ini, lihatPilih rahasia failover untuk dipasang.

Anda biasanya menggunakan bidang ini ketika rahasia failover bukan replika. Untuk contoh cara menentukan replika, lihatTentukan Wilayah failover untuk rahasia Multi-wilayah.

objectName

Nama atau ARN lengkap dari rahasia failover. Jika Anda menggunakan ARN, Wilayah di ARN harus cocok dengan bidang. `failoverRegion`

ObjectVersion

(Opsional) ID versi rahasia. Harus cocok dengan yang utama`objectVersion`. Tidak disarankan karena Anda harus memperbarui ID versi setiap kali Anda memperbarui rahasia. Secara default versi terbaru digunakan.

objectVersionLabel

(Opsional) Alias untuk versi. Defaultnya adalah versi terbaru AWSCURRENT. Lihat informasi yang lebih lengkap di the section called “Versi”.

Gunakan AWS Secrets Manager rahasia dalam GitHub pekerjaan

Untuk menggunakan rahasia dalam GitHub pekerjaan, Anda dapat menggunakan GitHub tindakan untuk mengambil rahasia dari AWS Secrets Manager dan menambahkannya sebagai variabel

Lingkungan bertopeng dalam alur kerja Anda GitHub . Untuk informasi selengkapnya tentang GitHub Tindakan, lihat Memahami GitHub Tindakan di GitHub Dokumen.

Ketika Anda menambahkan rahasia ke GitHub lingkungan Anda, itu tersedia untuk semua langkah lain dalam GitHub pekerjaan Anda. Ikuti panduan dalam Pengerasan Keamanan untuk GitHub Tindakan untuk membantu mencegah rahasia di lingkungan Anda disalahgunakan.

Anda dapat mengatur seluruh string dalam nilai rahasia sebagai nilai variabel lingkungan, atau jika string adalah JSON, Anda dapat mengurai JSON untuk mengatur variabel lingkungan individu untuk setiap pasangan nilai kunci JSON. Jika nilai rahasia adalah biner, tindakan mengubahnya menjadi string.

Untuk melihat variabel lingkungan yang dibuat dari rahasia Anda, aktifkan logging debug. Untuk informasi selengkapnya, lihat Mengaktifkan logging debug di Dokumen. GitHub

Untuk menggunakan variabel lingkungan yang dibuat dari rahasia Anda, lihat Variabel lingkungan di GitHub Dokumen.

Prasyarat

Untuk menggunakan tindakan ini, Anda harus terlebih dahulu mengonfigurasi AWS kredensil dan mengatur Wilayah AWS di GitHub lingkungan Anda dengan menggunakan langkah tersebut`configure-aws-credentials`. Ikuti petunjuk di Mengonfigurasi Tindakan AWS Kredensial Untuk GitHub Tindakan untuk Mengasumsikan peran secara langsung menggunakan penyedia GitHub OIDC. Ini memungkinkan Anda untuk menggunakan kredensil berumur pendek dan menghindari menyimpan kunci akses tambahan di luar Secrets Manager.

Peran IAM yang diasumsikan tindakan harus memiliki izin berikut:

- `GetSecretValue`pada rahasia yang ingin Anda ambil.
- `ListSecrets`pada semua rahasia.
- (Opsional) `Decrypt` pada KMS key jika rahasia dienkripsi dengan file. kunci yang dikelola pelanggan

Untuk informasi selengkapnya, lihat Kontrol autentikasi dan akses.

Penggunaan

Untuk menggunakan tindakan, tambahkan langkah ke alur kerja Anda yang menggunakan sintaks berikut.

```
- name: Step name
  uses: aws-actions/aws-secretsmanager-get-secrets@v2
  with:
    secret-ids: |
      secretId1
      ENV_VAR_NAME, secretId2
    parse-json-secrets: (Optional) true|false
```

Parameter-parameter

`secret-ids`

Rahasia ARNS, nama, dan awalan nama.

Secara default, langkah membuat setiap nama variabel lingkungan dari nama rahasia, diubah untuk menyertakan hanya huruf besar, angka, dan garis bawah, dan agar tidak dimulai dengan angka.

Untuk mengatur nama variabel lingkungan, masukkan sebelum ID rahasia, diikuti dengan koma. Misalnya `ENV_VAR_1, secretId` membuat variabel lingkungan bernama ENV_VAR_1 dari rahasia. `secretId` Nama variabel lingkungan dapat terdiri dari huruf besar, angka, dan garis bawah.

Untuk menggunakan awalan, masukkan setidaknya tiga karakter diikuti dengan tanda bintang. Misalnya `dev*` mencocokkan semua rahasia dengan nama yang dimulai di dev. Jumlah maksimum rahasia pencocokan yang dapat diambil adalah 100. Jika Anda menetapkan nama variabel, dan awalan cocok dengan beberapa rahasia, maka tindakan gagal.

`parse-json-secrets`

(Opsional) Secara default, tindakan menetapkan nilai variabel lingkungan ke seluruh string JSON dalam nilai rahasia. Atur `parse-json-secrets` `true` untuk membuat variabel lingkungan untuk setiap pasangan kunci/nilai di JSON.

Perhatikan bahwa jika JSON menggunakan kunci peka huruf besar/kecil seperti “nama” dan “Nama”, tindakan akan memiliki konflik nama duplikat. Dalam hal ini, atur `parse-json-secrets` ke `false` dan parse nilai rahasia JSON secara terpisah.

Penamaan variabel lingkungan

Variabel lingkungan yang dibuat oleh tindakan diberi nama yang sama dengan rahasia asalnya. Variabel lingkungan memiliki persyaratan penamaan yang lebih ketat daripada rahasia, sehingga tindakan mengubah nama rahasia untuk memenuhi persyaratan tersebut. Misalnya, tindakan mengubah huruf kecil menjadi huruf besar. Jika Anda mengurai JSON rahasia, maka nama variabel lingkungan mencakup nama rahasia dan nama kunci JSON, misalnya. `MYSECRET_KEYNAME`

Jika dua variabel lingkungan akan berakhir dengan nama yang sama, tindakan gagal. Dalam hal ini, Anda harus menentukan nama yang ingin Anda gunakan untuk variabel lingkungan sebagai alias.

Contoh kapan nama mungkin bertentangan:

- Sebuah rahasia bernama "MySecret" dan rahasia bernama “mysecret” keduanya akan menjadi variabel lingkungan bernama “MYSECRET”.
- Sebuah rahasia bernama “SECRET_KEYNAME” dan rahasia JSON-parsed bernama “Secret” dengan kunci bernama “keyname” keduanya akan menjadi variabel lingkungan bernama “SECRET_KEYNAME”.

Anda dapat mengatur nama variabel lingkungan dengan menentukan alias, seperti yang ditunjukkan dalam contoh berikut yang menciptakan variabel bernama. `ENV_VAR_NAME`

```
secret-ids: |
  ENV_VAR_NAME, secretId2
```

Alias kosong

- Jika Anda mengatur `parse-json-secrets: true` dan memasukkan alias kosong, diikuti dengan koma dan kemudian ID rahasia, tindakan tersebut memberi nama variabel lingkungan sama dengan kunci JSON yang diuraikan. Nama variabel tidak termasuk nama rahasia.

  Jika rahasia tidak berisi JSON yang valid, maka tindakan akan membuat satu variabel lingkungan dan menamainya sama dengan nama rahasia.
- Jika Anda mengatur `parse-json-secrets: false` dan memasukkan alias kosong, diikuti dengan koma dan ID rahasia, tindakan tersebut memberi nama variabel lingkungan seolah-olah Anda tidak menentukan alias.

Contoh berikut menunjukkan alias kosong.

```
,secret2
```

Contoh-contoh

Example 1 Dapatkan rahasia dengan nama dan oleh ARN

Contoh berikut menciptakan variabel lingkungan untuk rahasia diidentifikasi dengan nama dan oleh 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
```

Variabel lingkungan dibuat:

```
EXAMPLESECRETNAME: secretValue1
TEST1: secretValue2
_0_TEST_SECRET: secretValue3
_PROD_EXAMPLE_SECRET: secretValue4
SECRET_ALIAS_1: secretValue5
SECRET_ALIAS_2: secretValue6
SECRET2: secretValue7
```

Example 2 Dapatkan semua rahasia yang dimulai dengan awalan

Contoh berikut menciptakan variabel lingkungan untuk semua rahasia dengan nama yang dimulai dengan *beta*.

```
- name: Get Secret Names by Prefix
  uses: 2
  with:
    secret-ids: |
```

```
      beta*    # Retrieves all secrets that start with 'beta'
```

Variabel lingkungan dibuat:

```
BETASECRETNAME: secretValue1
BETATEST: secretValue2
BETA_NEWSECRET: secretValue3
```

Example 3 Parse JSON secara rahasia

Contoh berikut menciptakan variabel lingkungan dengan mengurai JSON dalam rahasia.

```
- 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
```

Rahasianya `test/secret` memiliki nilai rahasia berikut.

```
{
  "api_user": "user",
  "api_key": "key",
  "config": {
    "active": "true"
  }
}
```

Rahasianya `secret2` memiliki nilai rahasia berikut.

```
{
  "myusername": "alejandro_rosalez",
  "mypassword": "EXAMPLE_PASSWORD"
}
```

Variabel lingkungan dibuat:

```
TEST_SECRET_API_USER: "user"
TEST_SECRET_API_KEY: "key"
```

```
TEST_SECRET_CONFIG_ACTIVE: "true"
MYUSERNAME: "alejandro_rosalez"
MYPASSWORD: "EXAMPLE_PASSWORD"
```

Gunakan AWS Secrets Manager rahasia di AWS IoT Greengrass

AWS IoT Greengrass adalah perangkat lunak yang memperluas kemampuan cloud ke perangkat lokal. Hal ini memungkinkan perangkat untuk mengumpulkan dan menganalisis data lebih dekat ke sumber informasi, bereaksi secara mandiri terhadap acara lokal, dan berkomunikasi secara aman satu sama lain di jaringan lokal.

AWS IoT Greengrass memungkinkan Anda mengautentikasi dengan layanan dan aplikasi dari perangkat Greengrass tanpa kata sandi hard-coding, token, atau rahasia lainnya. Anda dapat menggunakannya AWS Secrets Manager untuk menyimpan dan mengelola rahasia Anda dengan aman di cloud. AWS IoT Greengrassmemperluas Secrets Manager ke perangkat inti Greengrass, sehingga konektor dan fungsi Lambda Anda dapat menggunakan rahasia lokal untuk berinteraksi dengan layanan dan aplikasi.

Untuk mengintegrasikan rahasia ke grup Greengrass, Anda membuat sumber daya grup yang mereferensi rahasia Secrets Manager. Sumber daya rahasia ini mereferensikan rahasia cloud dengan menggunakan ARN terkait. Untuk mempelajari cara membuat, mengelola, dan menggunakan sumber daya rahasia, lihat Bekerja dengan Sumber Daya Rahasia di Panduan AWS IoT Pengembang.

Untuk menyebarkan rahasia ke AWS IoT Greengrass Core, lihat Menyebarkan rahasia ke inti. AWS IoT Greengrass

Gunakan AWS Secrets Manager rahasia dalam AWS Lambda fungsi

Anda dapat menggunakan AWS Parameter dan Rahasia Lambda Ekstensi untuk mengambil dan menyimpan rahasia AWS Secrets Manager dalam fungsi Lambda tanpa menggunakan SDK. Mengambil rahasia yang di-cache lebih cepat daripada mengambilnya dari Secrets Manager. Karena ada biaya untuk memanggil Secrets Manager API, menggunakan cache dapat mengurangi biaya Anda. Ekstensi dapat mengambil rahasia Secrets Manager dan parameter Parameter Store. Untuk informasi tentang Parameter Store, lihat Integrasi Parameter Store dengan ekstensi Lambda di AWS Systems Manager Panduan Pengguna.

Ekstensi Lambda adalah proses pendamping yang menambah kemampuan fungsi Lambda. Untuk informasi selengkapnya, lihat Ekstensi Lambda di Panduan Pengembang Lambda. Untuk informasi tentang penggunaan ekstensi dalam gambar kontainer, lihat Bekerja dengan lapisan Lambda dan ekstensi dalam gambar kontainer. Lambda mencatat informasi eksekusi tentang ekstensi beserta fungsinya dengan menggunakan Amazon CloudWatch Logs. Secara default, ekstensi mencatat jumlah minimal informasi ke CloudWatch. Untuk mencatat detail lebih lanjut, atur variabel lingkungan `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` kedebug.

Untuk menyediakan cache dalam memori untuk parameter dan rahasia, ekstensi mengekspos titik akhir HTTP lokal, port localhost 2773, ke lingkungan Lambda. Anda dapat mengkonfigurasi port dengan mengatur variabel lingkungan`PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`.

Lambda membuat instance terpisah yang sesuai dengan tingkat konkurensi yang dibutuhkan fungsi Anda. Setiap instance diisolasi dan memelihara cache lokal sendiri dari data konfigurasi Anda. Untuk informasi selengkapnya tentang instans dan konkurensi Lambda, lihat Mengelola konkurensi untuk fungsi Lambda di Panduan Pengembang Lambda.

Untuk menambahkan ekstensi untuk ARM, Anda harus menggunakan `arm64` arsitektur untuk fungsi Lambda Anda. Untuk informasi selengkapnya, lihat Arsitektur set instruksi Lambda di Panduan Pengembang Lambda. Ekstensi ini mendukung ARM di Wilayah berikut: Asia Pasifik (Mumbai), AS Timur (Ohio), Eropa (Irlandia), Eropa (Frankfurt), Eropa (Zurich), AS Timur (Virginia N.), Eropa (London), Eropa (Spanyol), Asia Pasifik (Tokyo), AS Barat (Oregon), Asia Pasifik (Singapura), Asia Pasifik (Hyderabad), dan Asia Pasifik (Sydney).

Ekstensi menggunakan AWS klien. Untuk informasi tentang mengonfigurasi AWS klien, lihat Referensi pengaturan di AWS SDK dan Panduan Referensi Alat. Jika fungsi Lambda Anda berjalan di VPC, Anda perlu membuat titik akhir VPC sehingga ekstensi dapat melakukan panggilan ke Secrets Manager. Untuk informasi selengkapnya, lihat Titik akhir VPC.

Izin yang diperlukan:

- Peran eksekusi Lambda harus memiliki `secretsmanager:GetSecretValue` izin untuk rahasia.
- Jika rahasia dienkripsi dengan kunci yang dikelola pelanggan alih-alih Kunci yang dikelola AWS `aws/secretsmanager`, peran eksekusi juga memerlukan `kms:Decrypt` izin untuk kunci KMS.

Untuk menggunakan AWS Parameter dan Rahasia Ekstensi Lambda

1. Tambahkan layer ke fungsi Anda dengan melakukan salah satu hal berikut:
   - Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.

a. Pilih fungsi Anda, pilih Layers, dan kemudian pilih Add a layer.

b. Pada halaman Add layer, untuk AWS layer, pilih AWS Parameter dan Rahasia Lambda Extension, lalu pilih Add.

- Gunakan AWS CLI perintah berikut dengan ARN yang sesuai untuk Wilayah Anda. Untuk daftar ARN, lihat AWS Parameter dan Rahasia Lambda Extension ARN di AWS Systems Manager Panduan Pengguna.

```
aws lambda update-function-configuration \
    --function-name my-function \
    --layers LayerARN
```

2. Berikan izin ke peran eksekusi Lambda untuk dapat mengakses rahasia:
   - `secretsmanager:GetSecretValue`Izin untuk Rahasia Lihat the section called “Contoh: Izin untuk mengambil nilai rahasia individu”.
   - (Opsional) Jika rahasia dienkripsi dengan kunci yang dikelola pelanggan alih-alih Kunci yang dikelola AWS `aws/secretsmanager`, peran eksekusi juga memerlukan `kms:Decrypt` izin untuk kunci KMS.
   - Anda dapat menggunakan Attribute Based Access Control (ABAC) dengan peran Lambda untuk memungkinkan akses yang lebih terperinci ke rahasia di akun. Lihat informasi yang lebih lengkap di the section called “Contoh: Kontrol akses ke rahasia menggunakan tag” dan the section called “Contoh: Batasi akses ke identitas dengan tag yang cocok dengan tag rahasia”.
3. Konfigurasikan cache dengan variabel lingkungan Lambda.
4. Untuk mengambil rahasia dari cache ekstensi, Anda harus terlebih dahulu menambahkan `X-AWS-Parameters-Secrets-Token` ke header permintaan. Setel token ke`AWS_SESSION_TOKEN`, yang disediakan oleh Lambda untuk semua fungsi yang berjalan. Menggunakan header ini menunjukkan bahwa penelepon berada dalam lingkungan Lambda.

   Contoh Python berikut menunjukkan bagaimana menambahkan header.

```
import os
headers = {"X-Aws-Parameters-Secrets-Token": os.environ.get('AWS_SESSION_TOKEN')}
```

5. Untuk mengambil rahasia dalam fungsi Lambda, gunakan salah satu permintaan HTTP GET berikut:

   - Untuk mengambil rahasia, untuk`secretId`, gunakan ARN atau nama rahasia.

```
GET: /secretsmanager/get?secretId=secretId
```

- Untuk mengambil nilai rahasia sebelumnya atau versi tertentu dengan label pementasan, untuk`secretId`, gunakan ARN atau nama rahasia, dan untuk`versionStage`, gunakan label pementasan.

```
GET: /secretsmanager/get?secretId=secretId&versionStage=AWSPREVIOUS
```

- Untuk mengambil versi rahasia tertentu berdasarkan ID, untuk`secretId`, gunakan ARN atau nama rahasia, dan `versionId` untuk, gunakan ID versi.

```
GET: /secretsmanager/get?secretId=secretId&versionId=versionId
```

Example Mengambil rahasia (Python)

Contoh Python berikut menunjukkan bagaimana untuk mengambil rahasia dan mengurai hasil menggunakan. `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 Parameter dan Rahasia variabel lingkungan Ekstensi Lambda

Anda dapat mengkonfigurasi ekstensi dengan variabel lingkungan berikut.

Untuk informasi tentang cara menggunakan variabel lingkungan, lihat Menggunakan variabel lingkungan Lambda di Panduan Pengembang Lambda.

`PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED`

Setel ke true ke parameter cache dan rahasia. Setel ke false untuk tidak ada caching. Default adalah benar.

`PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE`

Jumlah maksimum rahasia dan parameter untuk cache. Harus berupa nilai dari 0 hingga 1000. Nilai 0 berarti tidak ada caching. Variabel ini diabaikan jika `SECRETS_MANAGER_TTL` keduanya `SSM_PARAMETER_STORE _TTL` dan 0. Defaultnya adalah 1000.

`PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`

Port untuk server HTTP lokal. Defaultnya adalah 2773.

`PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL`

Tingkat pencatatan ekstensi menyediakan:`debug`,, `info`warn,`error`, ataunone. Setel `debug` untuk melihat konfigurasi cache. Default-nya adalah `info`.

`PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS`

Jumlah maksimum koneksi untuk klien HTTP yang digunakan ekstensi untuk membuat permintaan ke Parameter Store atau Secrets Manager. Ini adalah konfigurasi per klien. Defaultnya adalah 3.

`SECRETS_MANAGER_TIMEOUT_MILLIS`

Batas waktu untuk permintaan ke Secrets Manager dalam milidetik. Nilai 0 berarti tidak ada batas waktu. Default-nya adalah 0.

`SECRETS_MANAGER_TTL`

TTL rahasia dalam cache dalam hitungan detik. Nilai 0 berarti tidak ada caching. Maksimumnya adalah 300 detik. Variabel ini diabaikan `PARAMETERS_SECRETS_CACHE_SIZE` jika 0. Defaultnya adalah 300 detik.

`SSM_PARAMETER_STORE_TIMEOUT_MILLIS`

Batas waktu untuk permintaan ke Parameter Store dalam milidetik. Nilai 0 berarti tidak ada batas waktu. Default-nya adalah 0.

`SSM_PARAMETER_STORE_TTL`

TTL parameter dalam cache dalam hitungan detik. Nilai 0 berarti tidak ada caching. Maksimumnya adalah 300 detik. Variabel ini diabaikan `PARAMETERS_SECRETS_CACHE_SIZE` jika 0. Defaultnya adalah 300 detik.

Gunakan AWS Secrets Manager rahasia di Parameter Store

AWSSystems Manager Parameter Store menyediakan penyimpanan hierarkis yang aman untuk manajemen data konfigurasi dan manajemen rahasia. Anda dapat menyimpan data seperti kata sandi, string database, dan kode lisensi sebagai nilai parameter. Namun, Parameter Store tidak menyediakan layanan rotasi otomatis untuk rahasia yang disimpan. Sebagai gantinya, Parameter Store memungkinkan Anda untuk menyimpan rahasia Anda di Secrets Manager, dan kemudian mereferensikan rahasia sebagai parameter Parameter Store.

Saat Anda mengonfigurasi Parameter Store dengan Secrets Manager, `secret-id` Parameter Store memerlukan garis miring (/) sebelum nama-string.

Untuk informasi selengkapnya, lihat Mereferensikan AWS Secrets Manager Rahasia dari Parameter Penyimpanan Parameter di Panduan AWS Systems Manager Pengguna.

Putar AWS Secrets Manager rahasia

Rotasi adalah proses memperbarui rahasia secara berkala. Ketika Anda memutar rahasia, Anda memperbarui kredensil di kedua rahasia dan database atau layanan. Di Secrets Manager, Anda dapat mengatur rotasi otomatis untuk rahasia Anda.

Topik

Cara kerja rotasi

> **Tip**
>
> Untuk beberapaRahasia yang dikelola oleh layanan lain, Anda menggunakan rotasi terkelola. Untuk menggunakanRotasi terkelola, Anda pertama kali membuat rahasia melalui layanan pengelolaan.

Rotasi Secrets Manager menggunakan AWS Lambda fungsi untuk memperbarui rahasia dan database atau layanan. Untuk informasi tentang biaya penggunaan fungsi Lambda, lihat. Harga

Untuk memutar rahasia, Secrets Manager memanggil fungsi Lambda sesuai dengan jadwal yang Anda atur. Anda dapat mengatur jadwal untuk memutar setelah periode waktu tertentu, misalnya setiap 30 hari, atau Anda dapat membuat ekspresi cron. Lihat Ekspresi jadwal. Jika Anda juga memperbarui nilai rahasia Anda secara manual saat rotasi otomatis diatur, maka Secrets Manager menganggap bahwa rotasi yang valid ketika menghitung tanggal rotasi berikutnya.

Untuk keamanan, Secrets Manager hanya mengizinkan fungsi rotasi Lambda untuk memutar rahasia secara langsung. Fungsi rotasi tidak dapat memanggil fungsi Lambda kedua untuk memutar rahasia.

Secrets Manager menggunakan label pementasan untuk memberi label versi rahasia selama rotasi. Selama rotasi, Secrets Manager memanggil fungsi yang sama beberapa kali, setiap kali dengan parameter yang berbeda. Secrets Manager memanggil fungsi dengan struktur permintaan parameter JSON berikut:

```
{
    "Step" : "request.type",
    "SecretId" : "string",
    "ClientRequestToken" : "string"
}
```

Fungsi rotasi melakukan pekerjaan memutar rahasia. Ada empat langkah untuk memutar rahasia, yang sesuai dengan empat langkah berikut dalam fungsi rotasi Lambda:

1. Buat versi baru dari secret (**`createSecret`**)

   Langkah pertama rotasi adalah membuat versi baru dari rahasia. Dalam template rotasi database yang disediakan oleh Secrets Manager, fungsi rotasi Lambda menghasilkan kata sandi 32 karakter untuk versi baru. Versi baru dapat berisi kata sandi baru, nama pengguna dan kata sandi baru, atau informasi rahasia lainnya. Fungsi rotasi Lambda memberi label pada versi baru. `AWSPENDING`

2. Mengubah kredensi dalam database atau layanan () **`setSecret`**

   Selanjutnya, fungsi rotasi Lambda mengubah kredensyal dalam database atau layanan agar sesuai dengan kredensyal baru dalam versi rahasia. `AWSPENDING` Bergantung pada strategi rotasi Anda, langkah ini dapat membuat pengguna baru dengan izin yang sama dengan pengguna yang ada.

   Fungsi rotasi untuk Amazon RDS (kecuali Oracle dan Db2) dan Amazon DocumentDB secara otomatis menggunakan Secure Socket Layer (SSL) atau Transport Layer Security (TLS) untuk terhubung ke database Anda, jika tersedia. Jika tidak, mereka menggunakan koneksi yang tidak terenkripsi.

   > **Note**
   >
   > Jika Anda mengatur rotasi rahasia otomatis sebelum 20 Desember 2021, fungsi rotasi Anda mungkin didasarkan pada templat lama yang tidak mendukung SSL/TLS. Lihat

Menentukan kapan fungsi rotasi Anda dibuat. Jika dibuat sebelum 20 Desember 2021, untuk mendukung koneksi yang menggunakan SSL/TLS, Anda perlu membuat ulang fungsi rotasi Anda.

3. Uji versi rahasia baru (**`testSecret`**)

   Selanjutnya, fungsi rotasi Lambda menguji `AWSPENDING` versi rahasia dengan menggunakannya untuk mengakses database atau layanan. Fungsi rotasi berdasarkan Templat fungsi rotasi uji rahasia baru dengan menggunakan akses baca. Bergantung pada jenis akses yang dibutuhkan aplikasi Anda, Anda dapat memperbarui fungsi untuk menyertakan akses lain seperti akses tulis.

4. Selesaikan rotasi (**`finishSecret`**)

   Terakhir, fungsi rotasi Lambda memindahkan label `AWSCURRENT` dari versi rahasia sebelumnya ke versi ini, yang juga menghapus `AWSPENDING` label dalam panggilan API yang sama. Anda tidak boleh menghapus `AWSPENDING` sebelum titik ini, dan Anda tidak boleh menghapusnya dengan menggunakan panggilan API terpisah, karena itu dapat menunjukkan kepada Secrets Manager bahwa rotasi tidak berhasil diselesaikan. Secrets Manager menambahkan label `AWSPREVIOUS` pementasan ke versi sebelumnya, sehingga Anda mempertahankan versi rahasia terakhir yang diketahui.

Selama rotasi, Secrets Manager mencatat peristiwa yang menunjukkan keadaan rotasi. Untuk informasi selengkapnya, lihat the section called “Log dengan AWS CloudTrail ”.

Jika ada langkah rotasi yang gagal, Secrets Manager mencoba ulang seluruh proses rotasi beberapa kali.

Ketika rotasi berhasil, label `AWSPENDING` pementasan mungkin dilampirkan ke versi yang sama dengan `AWSCURRENT` versi, atau mungkin tidak dilampirkan ke versi apa pun. Jika label `AWSPENDING` pementasan ada tetapi tidak dilampirkan ke versi yang sama dengan`AWSCURRENT`, maka pemanggilan rotasi selanjutnya mengasumsikan bahwa permintaan rotasi sebelumnya masih dalam proses dan mengembalikan kesalahan. Ketika rotasi tidak berhasil, label `AWSPENDING` pementasan mungkin dilampirkan ke versi rahasia kosong. Untuk informasi selengkapnya, lihat Memecahkan masalah rotasi.

Setelah rotasi berhasil, aplikasi yang Ambil rahasia dari AWS Secrets Manager dari Secrets Manager secara otomatis mendapatkan kredensyal yang diperbarui. Untuk detail lebih lanjut tentang cara kerja setiap langkah rotasi, lihatthe section called “Templat fungsi rotasi”.

Rotasi terkelola untuk AWS Secrets Manager rahasia

Beberapa layanan menawarkan rotasi terkelola, di mana layanan mengkonfigurasi dan mengelola rotasi untuk Anda. Dengan rotasi terkelola, Anda tidak menggunakan AWS Lambda fungsi untuk memperbarui rahasia dan kredensyal dalam database. Layanan berikut menawarkan rotasi terkelola:

- Amazon ECS Service Connect menawarkan rotasi terkelola untuk sertifikat AWS Private Certificate Authority TLS. Untuk informasi selengkapnya, lihat TLS dengan Service Connect di Panduan Pengembang Layanan Amazon Elastic Container.
- Amazon RDS menawarkan rotasi terkelola untuk kredensyal pengguna master. Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS dan AWS Secrets Manager di Panduan Pengguna Amazon RDS.
- Amazon Aurora menawarkan rotasi terkelola untuk kredensyal pengguna master. Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon Aurora dan AWS Secrets Manager di Panduan Pengguna Amazon Aurora.
- Amazon Redshift menawarkan rotasi terkelola untuk kata sandi admin. Untuk informasi selengkapnya, lihat Mengelola kata sandi admin Amazon Redshift menggunakan AWS Secrets Manager dalam Panduan Manajemen Amazon Redshift.

Untuk semua jenis rahasia lainnya, lihatPutar rahasia.

Rotasi untuk rahasia terkelola biasanya selesai dalam satu menit. Selama rotasi, koneksi baru yang mengambil rahasia mungkin mendapatkan versi kredensyal sebelumnya. Dalam aplikasi, kami sangat menyarankan agar Anda mengikuti praktik terbaik menggunakan pengguna database yang dibuat dengan hak istimewa minimal yang diperlukan untuk aplikasi Anda, daripada menggunakan pengguna utama. Untuk pengguna aplikasi, untuk ketersediaan tertinggi, Anda dapat menggunakan strategi rotasi pengguna Alternating.

Untuk mengubah jadwal rotasi terkelola (konsol)

1. Buka rahasia terkelola di konsol Secrets Manager. Anda dapat mengikuti tautan dari layanan pengelolaan, atau mencari rahasia di konsol Secrets Manager.
2. Di bawah Jadwal rotasi, masukkan jadwal Anda di zona waktu UTC baik di pembuat ekspresi Jadwal atau sebagai ekspresi Jadwal. Secrets Manager menyimpan jadwal Anda sebagai `cron()` ekspresi `rate()` atau. Jendela rotasi secara otomatis dimulai pada tengah malam kecuali Anda menentukan waktu Mulai. Anda dapat memutar rahasia sesering setiap empat jam. Untuk informasi selengkapnya, lihat Ekspresi jadwal.

3. (Opsional) Untuk durasi Jendela, pilih panjang jendela di mana Anda ingin Secrets Manager memutar rahasia Anda, **3h** misalnya untuk jendela tiga jam. Jendela tidak boleh meluas ke jendela rotasi berikutnya. Jika Anda tidak menentukan Durasi jendela, untuk jadwal rotasi dalam jam, jendela akan ditutup secara otomatis setelah satu jam. Untuk jadwal rotasi dalam beberapa hari, jendela secara otomatis ditutup pada akhir hari.
4. Pilih Simpan.

Untuk mengubah jadwal rotasi terkelola (AWS CLI)

- Panggil `rotate-secret`. Contoh berikut memutar rahasia antara pukul 16:00 dan 18:00 UTC pada hari pertama dan ke-15 setiap bulan. Lihat informasi yang lebih lengkap di Ekspresi jadwal.

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

Siapkan rotasi otomatis untuk rahasia Amazon RDS, Amazon Aurora, Amazon Redshift, atau Amazon DocumentDB menggunakan konsol

Rotasi adalah proses memperbarui rahasia secara berkala. Ketika Anda memutar rahasia, Anda memperbarui kredensil di kedua rahasia dan database. Di Secrets Manager, Anda dapat mengatur rotasi otomatis untuk rahasia database Anda.

Secrets Manager menggunakan fungsi Lambda untuk memutar rahasia. Untuk ikhtisar, lihat the section called “Cara kerja rotasi”.

> **Tip**
>
> Untuk beberapaRahasia yang dikelola oleh layanan lain, Anda menggunakan rotasi terkelola. Untuk menggunakanRotasi terkelola, Anda pertama kali membuat rahasia melalui layanan pengelolaan.

Untuk mengatur rotasi menggunakan konsol, Anda harus terlebih dahulu memilih strategi rotasi. Kemudian Anda mengonfigurasi rahasia rotasi, yang menciptakan fungsi rotasi Lambda jika Anda belum memilikinya. Konsol juga menetapkan izin untuk peran eksekusi fungsi Lambda. Langkah terakhir adalah memastikan bahwa fungsi rotasi Lambda dapat mengakses Secrets Manager dan database Anda melalui jaringan.

Untuk mengaktifkan rotasi otomatis, Anda harus memiliki izin untuk membuat peran eksekusi IAM dan melampirkan kebijakan izin padanya. Anda membutuhkan keduanya `iam:CreateRole` dan `iam:AttachRolePolicy` izin.

> **Warning**
>
> Pemberian identitas `iam:CreateRole` dan `iam:AttachRolePolicy` izin memungkinkan identitas untuk memberikan izin apa pun kepada diri mereka sendiri.

Langkah:

Langkah 1: Pilih strategi rotasi dan (opsional) buat rahasia superuser

Untuk Amazon RDS, Amazon Redshift, dan Amazon DocumentDB, Secrets Manager menawarkan dua strategi rotasi:

Strategi rotasi pengguna tunggal

Strategi ini memperbarui kredensil untuk satu pengguna dalam satu rahasia. Untuk instans Amazon RDS Db2, karena pengguna tidak dapat mengubah kata sandi mereka sendiri, Anda harus memberikan kredensi admin dalam rahasia terpisah. Ini adalah strategi rotasi paling sederhana, dan cocok untuk sebagian besar kasus penggunaan. Secara khusus, kami menyarankan Anda menggunakan strategi ini untuk kredensyal untuk pengguna satu kali (ad hoc) atau interaktif.

Ketika rahasia berputar, koneksi database terbuka tidak terputus. Sementara rotasi sedang terjadi, ada periode waktu singkat antara ketika kata sandi dalam database berubah dan ketika rahasia diperbarui. Selama waktu ini, ada risiko rendah database menolak panggilan yang menggunakan kredenal yang diputar. Anda dapat mengurangi risiko ini dengan strategi coba lagi yang tepat. Setelah rotasi, koneksi baru menggunakan kredensil baru.

Strategi rotasi pengguna bergantian

Strategi ini memperbarui kredensil untuk dua pengguna dalam satu rahasia. Anda membuat pengguna pertama, dan selama rotasi pertama, fungsi rotasi mengkloningnya untuk membuat pengguna kedua. Setiap kali rahasia berputar, fungsi rotasi mengganti kata sandi pengguna mana yang diperbarui. Karena sebagian besar pengguna tidak memiliki izin untuk mengkloning diri mereka sendiri, Anda harus memberikan kredensialnya untuk rahasia lain. `superuser` Sebaiknya gunakan strategi rotasi pengguna tunggal ketika pengguna kloning di database Anda tidak memiliki izin yang sama dengan pengguna asli, dan untuk kredensyal untuk pengguna satu kali (ad hoc) atau interaktif.

Strategi ini sesuai untuk database dengan model izin di mana satu peran memiliki tabel database dan peran kedua memiliki izin untuk mengakses tabel database. Ini juga sesuai untuk aplikasi yang membutuhkan ketersediaan tinggi. Jika aplikasi mengambil rahasia selama rotasi, aplikasi masih mendapatkan set kredensyal yang valid. Setelah rotasi, keduanya `user` dan `user_clone` kredensialnya valid. Bahkan ada lebih sedikit kemungkinan aplikasi mendapatkan penolakan selama jenis rotasi ini daripada rotasi pengguna tunggal. Jika database di-host di server farm di mana perubahan kata sandi membutuhkan waktu untuk menyebar ke semua server, ada risiko database menolak panggilan yang menggunakan kredensyal baru. Anda dapat mengurangi risiko ini dengan strategi coba lagi yang tepat.

Secrets Manager membuat pengguna kloning dengan izin yang sama dengan pengguna asli. Jika Anda mengubah izin pengguna asli setelah klon dibuat, Anda juga harus mengubah izin pengguna kloning.

> **Important**
>
> Jika Anda memilih strategi pengguna bergantian, Anda harus Buat rahasia database dan menyimpan kredensyal superuser database di dalamnya. Anda memerlukan rahasia dengan kredensyal superuser karena rotasi mengkloning pengguna pertama, dan sebagian besar pengguna tidak memiliki izin itu.

Langkah 2: Konfigurasikan rotasi dan buat fungsi rotasi

Fungsi rotasi untuk Amazon RDS (kecuali Oracle dan Db2) dan Amazon DocumentDB secara otomatis menggunakan Secure Socket Layer (SSL) atau Transport Layer Security (TLS) untuk terhubung ke database Anda, jika tersedia. Jika tidak, mereka menggunakan koneksi yang tidak terenkripsi.

Untuk mengaktifkan rotasi untuk rahasia Amazon RDS, Amazon DocumentDB, atau Amazon Redshift

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pada halaman Rahasia, pilih rahasia Anda.
3. Pada halaman Detail rahasia, di bagian konfigurasi Rotasi, pilih Edit rotasi.
4. Dalam kotak dialog Edit konfigurasi rotasi, lakukan hal berikut:

   a. Nyalakan Rotasi otomatis.

   b. Di bawah Jadwal rotasi, masukkan jadwal Anda di zona waktu UTC baik di pembuat ekspresi Jadwal atau sebagai ekspresi Jadwal. Secrets Manager menyimpan jadwal Anda sebagai `cron()` ekspresi `rate()` atau. Jendela rotasi secara otomatis dimulai pada tengah malam kecuali Anda menentukan waktu Mulai. Anda dapat memutar rahasia sesering setiap empat jam. Untuk informasi selengkapnya, lihat Ekspresi jadwal.

   c. (Opsional) Untuk durasi Jendela, pilih panjang jendela di mana Anda ingin Secrets Manager memutar rahasia Anda, **3h** misalnya untuk jendela tiga jam. Jendela tidak boleh meluas ke jendela rotasi berikutnya. Jika Anda tidak menentukan Durasi jendela, untuk jadwal rotasi dalam jam, jendela akan ditutup secara otomatis setelah satu jam. Untuk jadwal rotasi dalam beberapa hari, jendela secara otomatis ditutup pada akhir hari.

   d. (Opsional) Pilih Putar segera ketika rahasia disimpan untuk memutar rahasia Anda ketika Anda menyimpan perubahan Anda. Jika Anda menghapus kotak centang, maka rotasi pertama akan dimulai pada jadwal yang Anda tetapkan.

      Jika rotasi gagal, misalnya karena Langkah 3 dan 4 belum selesai, Secrets Manager mencoba ulang proses rotasi beberapa kali.

   e. Di bawah fungsi Rotasi, lakukan salah satu hal berikut:

      - Pilih Buat fungsi Lambda baru dan masukkan nama untuk fungsi baru Anda. Secrets Manager menambahkan `SecretsManager` ke awal nama fungsi. Secrets Manager membuat fungsi berdasarkan template yang sesuai dan menetapkan izin yang diperlukan untuk peran eksekusi Lambda.

- Pilih Gunakan fungsi Lambda yang ada untuk menggunakan kembali fungsi rotasi yang Anda gunakan untuk rahasia lain. Fungsi rotasi yang tercantum di bawah Konfigurasi VPC yang direkomendasikan memiliki VPC dan grup keamanan yang sama dengan database, yang membantu fungsi mengakses database.

f. Untuk strategi Rotasi, pilih strategi Single user atau Alternating users. Untuk informasi lebih lanjut, lihat the section called “Langkah 1: Pilih strategi rotasi dan (opsional) buat rahasia superuser”.

5. Pilih Simpan.

Langkah 3: (Opsional) Tetapkan kondisi izin tambahan pada fungsi rotasi

Dalam kebijakan sumber daya untuk fungsi rotasi Anda, sebaiknya sertakan kunci konteks `aws:SourceAccount`untuk membantu mencegah Lambda digunakan sebagai wakil yang bingung. Untuk beberapa AWS layanan, untuk menghindari skenario wakil yang membingungkan, AWS merekomendasikan agar Anda menggunakan kunci kondisi `aws:SourceAccount`global `aws:SourceArn`dan global. Namun, jika Anda menyertakan `aws:SourceArn` kondisi dalam kebijakan fungsi rotasi Anda, fungsi rotasi hanya dapat digunakan untuk memutar rahasia yang ditentukan oleh ARN tersebut. Kami menyarankan Anda hanya menyertakan kunci konteks `aws:SourceAccount` sehingga Anda dapat menggunakan fungsi rotasi untuk beberapa rahasia.

Untuk memperbarui kebijakan sumber daya fungsi rotasi

1. Di konsol Secrets Manager, pilih rahasia Anda, dan kemudian pada halaman detail, di bawah konfigurasi Rotasi, pilih fungsi rotasi Lambda. Konsol Lambda terbuka.
2. Ikuti petunjuk di Menggunakan kebijakan berbasis sumber daya untuk Lambda untuk menambahkan kondisi. `aws:sourceAccount`

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

Jika rahasia dienkripsi dengan kunci KMS selain, Secrets Kunci yang dikelola AWS `aws/secretsmanager` Manager memberikan izin peran eksekusi Lambda untuk menggunakan kunci tersebut. Anda dapat menggunakan konteks enkripsi secretArn untuk membatasi penggunaan

fungsi dekripsi, sehingga peran fungsi rotasi hanya memiliki akses untuk mendekripsi rahasia yang bertanggung jawab untuk berputar.

Untuk memperbarui peran eksekusi fungsi rotasi

1. Dari fungsi rotasi Lambda, pilih Konfigurasi, lalu di bawah Peran eksekusi, pilih nama Peran.
2. Ikuti petunjuk di Memodifikasi kebijakan izin peran untuk menambahkan kondisi. `kms:EncryptionContext:SecretARN`

```
"Condition": {
    "StringEquals": {
        "kms:EncryptionContext:SecretARN": "SecretARN"
    }
},
```

Langkah 4: Siapkan akses jaringan untuk fungsi rotasi

Untuk dapat memutar rahasia, fungsi rotasi Lambda harus dapat mengakses rahasia dan database atau layanan.

Untuk mengakses rahasia

Fungsi rotasi Lambda Anda harus dapat mengakses titik akhir Secrets Manager. Jika fungsi Lambda Anda dapat mengakses internet, maka Anda dapat menggunakan titik akhir publik. Untuk menemukan titik akhir, lihatthe section called “Titik akhir Secrets Manager”.

Jika fungsi Lambda Anda berjalan di VPC yang tidak memiliki akses internet, kami sarankan Anda mengonfigurasi titik akhir pribadi layanan Secrets Manager dalam VPC Anda. VPC Anda kemudian dapat mencegat permintaan yang ditujukan ke titik akhir regional publik dan mengarahkannya ke titik akhir pribadi. Untuk informasi selengkapnya, lihat Titik akhir VPC.

Atau, Anda dapat mengaktifkan fungsi Lambda Anda untuk mengakses titik akhir publik Secrets Manager dengan menambahkan gateway NAT atau gateway internet ke VPC Anda, yang memungkinkan lalu lintas dari VPC Anda mencapai titik akhir publik. Ini membuat VPC Anda berisiko lebih besar karena alamat IP untuk gateway dapat diserang dari Internet publik.

Untuk mengakses database atau layanan

Jika database atau layanan Anda berjalan pada instans Amazon EC2 di VPC, sebaiknya Anda mengonfigurasi fungsi Lambda agar berjalan di VPC yang sama. Kemudian fungsi rotasi

dapat berkomunikasi langsung dengan layanan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses VPC.

Untuk mengizinkan fungsi Lambda mengakses database atau layanan, Anda harus memastikan bahwa grup keamanan yang dilampirkan ke fungsi rotasi Lambda Anda memungkinkan koneksi keluar ke database atau layanan. Anda juga harus memastikan bahwa grup keamanan yang dilampirkan ke database atau layanan Anda mengizinkan koneksi masuk dari fungsi rotasi Lambda.

Untuk rotasi pengguna bergantian di mana rahasia superuser dikelola oleh AWS layanan lain, fungsi rotasi Lambda harus dapat memanggil titik akhir layanan untuk mendapatkan informasi koneksi database. Kami menyarankan Anda mengonfigurasi titik akhir VPC untuk layanan database. Untuk informasi selengkapnya, lihat:

- Amazon RDS API dan titik akhir VPC antarmuka di Panduan Pengguna Amazon RDS.
- Bekerja dengan titik akhir VPC di Panduan Manajemen Pergeseran Merah Amazon.

Langkah 5: (Opsional) Sesuaikan fungsi rotasi

Dalam kasus yang jarang terjadi, Anda mungkin ingin menyesuaikan fungsi rotasi. Misalnya, dengan rotasi pengguna bergantian, Secrets Manager membuat pengguna kloning dengan menyalin parameter konfigurasi runtime dari pengguna pertama. Jika Anda ingin menyertakan lebih banyak atribut, atau mengubah mana yang diberikan kepada pengguna kloning, Anda perlu memperbarui kode dalam `set_secret` fungsi.

Untuk contoh lain, untuk Amazon RDS MySQL, dalam rotasi pengguna bergantian, Secrets Manager membuat pengguna kloning dengan nama tidak lebih dari 16 karakter. Anda dapat memodifikasi fungsi rotasi untuk memungkinkan nama pengguna yang lebih panjang. MySQL versi 5.7 dan yang lebih tinggi mendukung nama pengguna hingga 32 karakter, namun Secrets Manager menambahkan "_clone" (enam karakter) ke akhir nama pengguna, jadi Anda harus menjaga nama pengguna maksimal 26 karakter.

Untuk membuka fungsi rotasi Lambda Anda untuk mengedit

1. Di konsol Secrets Manager, pilih rahasia Anda.
2. Di bagian konfigurasi Rotasi, di bawah fungsi rotasi Lambda, pilih fungsi rotasi Anda.

   Konsol Lambda terbuka.

   - Untuk mengubah kode dalam fungsi, gulir ke bawah ke bagian Sumber kode.

- Untuk MySQL versi 5.7 dan yang lebih tinggi, untuk rotasi pengguna bergantian, untuk mengubah panjang nama pengguna maksimum, di bawah variabel Lingkungan, ubah. `USERNAME_CHARACTER_LIMIT`

Langkah selanjutnya

Lihat the section called “Memecahkan masalah rotasi”.

Siapkan rotasi otomatis untuk AWS Secrets Manager rahasia menggunakan konsol

Rotasi adalah proses memperbarui rahasia secara berkala. Ketika Anda memutar rahasia, Anda memperbarui kredensil di kedua rahasia dan database atau layanan yang menjadi tujuan rahasianya.

Secrets Manager menggunakan fungsi Lambda untuk memutar rahasia. Untuk ringkasan, lihat the section called “Cara kerja rotasi”.

Anda juga dapat menggunakan AWS CLI untuk mengatur rotasi. Untuk informasi selengkapnya, lihat Rotasi otomatis (AWS CLI).

Untuk mengatur rotasi menggunakan konsol, Anda terlebih dahulu mengkonfigurasi rahasia untuk rotasi. Selama langkah itu, Anda juga membuat fungsi rotasi Lambda kosong. Selanjutnya, Anda menetapkan izin untuk fungsi rotasi dan untuk peran eksekusi Lambda. Kemudian Anda menulis kode fungsi rotasi. Langkah terakhir adalah memastikan bahwa fungsi rotasi Lambda dapat mengakses Secrets Manager dan database atau layanan Anda melalui jaringan.

Untuk rahasia database, lihatthe section called “Rotasi otomatis untuk rahasia database (konsol)”.

Untuk mengaktifkan rotasi otomatis, Anda harus memiliki izin untuk membuat peran eksekusi IAM dan melampirkan kebijakan izin padanya. Anda membutuhkan keduanya `iam:CreateRole` dan `iam:AttachRolePolicy` izin.

> **Warning**
>
> Pemberian identitas `iam:CreateRole` dan `iam:AttachRolePolicy` izin memungkinkan identitas untuk memberikan izin apa pun kepada diri mereka sendiri.

Langkah:

Langkah 1: Konfigurasikan rahasia untuk rotasi

Pada langkah ini, Anda mengatur jadwal rotasi untuk rahasia Anda dan membuat fungsi rotasi kosong. Rahasia Anda tidak akan diputar sampai Anda selesai menulis fungsi rotasi. Jika Anda menjadwalkan rotasi sebelum fungsi rotasi ditulis, atau jika gagal karena alasan apa pun, Secrets Manager akan mencoba lagi fungsi rotasi beberapa kali.

Untuk mengkonfigurasi rotasi dan membuat fungsi rotasi kosong

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pada halaman Rahasia, pilih rahasia Anda.
3. Pada halaman Detail rahasia, di bagian konfigurasi Rotasi, pilih Edit rotasi. Dalam kotak dialog Edit konfigurasi rotasi, lakukan hal berikut:

    a. Nyalakan rotasi otomatis.

    b. Di bawah Jadwal rotasi, masukkan jadwal Anda di zona waktu UTC baik di pembuat ekspresi Jadwal atau sebagai ekspresi Jadwal. Secrets Manager menyimpan jadwal Anda sebagai `cron()` ekspresi `rate()` atau. Jendela rotasi secara otomatis dimulai pada tengah malam kecuali Anda menentukan waktu Mulai. Anda dapat memutar rahasia sesering setiap empat jam. Untuk informasi selengkapnya, lihat Ekspresi jadwal.

    c. (Opsional) Untuk durasi Jendela, pilih panjang jendela di mana Anda ingin Secrets Manager memutar rahasia Anda, **3h** misalnya untuk jendela tiga jam. Jendela tidak boleh meluas ke jendela rotasi berikutnya. Jika Anda tidak menentukan Durasi jendela, untuk jadwal rotasi dalam jam, jendela akan ditutup secara otomatis setelah satu jam. Untuk jadwal rotasi dalam beberapa hari, jendela secara otomatis ditutup pada akhir hari.

    d. (Opsional) Pilih Putar segera ketika rahasia disimpan untuk memutar rahasia Anda ketika Anda menyimpan perubahan Anda. Jika Anda menghapus kotak centang, maka rotasi pertama akan dimulai pada jadwal yang Anda tetapkan.

e. Di bawah fungsi Rotasi, pilih Buat fungsi. Konsol Lambda terbuka di jendela baru.

- Di konsol Lambda, pada halaman Create function, lakukan salah satu hal berikut:
  - Jika Anda melihat Browse repositori aplikasi tanpa server, pilihlah.

    A. Di bawah Aplikasi publik, di kotak pencarian, masukkan SecretsManagerRotationTemplate.

    B. Pilih Tampilkan aplikasi yang membuat peran IAM kustom atau kebijakan sumber daya.

    C. Pilih SecretsManagerRotationTemplateubin.

    D. Pada halaman Tinjau, konfigurasikan, dan terapkan, di ubin Pengaturan aplikasi, isi bidang yang diperlukan, lalu pilih Deploy. Untuk daftar titik akhir, lihatthe section called “Titik akhir Secrets Manager”.
  - Jika Anda tidak melihat Browse repositori aplikasi tanpa server, Anda Wilayah AWS mungkin tidak mendukung. AWS Serverless Application Repository Pilih Tulis dari awal.

    A. Untuk nama Fungsi, masukkan nama untuk fungsi rotasi Anda.

    B. Untuk Runtime, pilih Python 3.9.

    C. Ketika fungsi Lambda baru terbuka, gulir ke bawah untuk memilih Konfigurasi, dan kemudian di sebelah kiri pilih Izin.

    D. Gulir ke bawah ke kebijakan berbasis Sumber Daya dan pilih Tambahkan izin untuk memberikan izin bagi Secrets Manager untuk menjalankan fungsi tersebut. Untuk melampirkan kebijakan sumber daya ke fungsi Lambda, lihat Menggunakan kebijakan berbasis sumber daya untuk Lambda.

    Kebijakan berikut menunjukkan cara mengizinkan Secrets Manager menjalankan fungsi Lambda.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "secretsmanager.amazonaws.com"
```

```
            },
        "Action": "lambda:InvokeFunction",
        "Resource": "LambdaRotationFunctionARN"
    }
    ]
}
```

f. Beralih kembali ke konsol Secrets Manager untuk melampirkan fungsi rotasi baru ke rahasia Anda.

g. Untuk fungsi rotasi Lambda, pilih tombol refresh. Kemudian dalam daftar fungsi, pilih fungsi baru Anda.

h. Pilih Simpan.

Langkah 2: Tetapkan izin untuk fungsi rotasi

Fungsi rotasi Lambda memerlukan izin untuk mengakses rahasia di Secrets Manager, dan memerlukan izin untuk mengakses database atau layanan Anda. Pada langkah ini, Anda memberikan izin ini ke peran eksekusi Lambda. Jika rahasia dienkripsi dengan kunci KMS selain Kunci yang dikelola AWS`aws/secretsmanager`, maka Anda perlu memberikan izin peran eksekusi Lambda untuk menggunakan kunci tersebut. Anda dapat menggunakan konteks enkripsi secretArn untuk membatasi penggunaan fungsi dekripsi, sehingga peran fungsi rotasi hanya memiliki akses untuk mendekripsi rahasia yang bertanggung jawab untuk berputar. Untuk contoh kebijakan, lihatIzin untuk rotasi.

Untuk petunjuknya, lihat Peran eksekusi Lambda di Panduan AWS LambdaPengembang.

Langkah 3: (Opsional) Tetapkan kondisi izin tambahan pada fungsi rotasi

Dalam kebijakan sumber daya untuk fungsi rotasi Anda, sebaiknya sertakan kunci konteks `aws:SourceAccount`untuk membantu mencegah Lambda digunakan sebagai wakil yang bingung. Untuk beberapa layanan AWS, untuk menghindari skenario wakil yang membingungkan, AWS merekomendasikan agar Anda menggunakan baik kunci kondisi global `aws:SourceArn` dan `aws:SourceAccount`. Namun, jika Anda menyertakan kondisi `aws:SourceArn` dalam kebijakan fungsi rotasi Anda, fungsi rotasi hanya dapat digunakan untuk memutar rahasia yang ditentukan oleh ARN tersebut. Kami menyarankan Anda hanya menyertakan kunci konteks `aws:SourceAccount` sehingga Anda dapat menggunakan fungsi rotasi untuk beberapa rahasia.

Untuk memperbarui kebijakan sumber daya fungsi rotasi

1. Di konsol Secrets Manager, pilih rahasia Anda, dan kemudian pada halaman detail, di bawah konfigurasi Rotasi, pilih fungsi rotasi Lambda. Konsol Lambda terbuka.
2. Ikuti petunjuk di Menggunakan kebijakan berbasis sumber daya untuk Lambda untuk menambahkan kondisi. `aws:sourceAccount`

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

Langkah 4: Siapkan akses jaringan untuk fungsi rotasi

Untuk dapat memutar rahasia, fungsi rotasi Lambda harus dapat mengakses rahasia. Jika rahasia Anda berisi kredensil, maka fungsi Lambda juga harus dapat mengakses sumber kredensil tersebut, seperti database atau layanan.

Untuk mengakses rahasia

Fungsi rotasi Lambda Anda harus dapat mengakses titik akhir Secrets Manager. Jika fungsi Lambda Anda dapat mengakses internet, maka Anda dapat menggunakan titik akhir publik. Untuk menemukan titik akhir, lihatthe section called “Titik akhir Secrets Manager”.

Jika fungsi Lambda Anda berjalan di VPC yang tidak memiliki akses internet, kami sarankan Anda mengonfigurasi titik akhir pribadi layanan Secrets Manager dalam VPC Anda. VPC Anda kemudian dapat mencegat permintaan yang ditujukan ke titik akhir regional publik dan mengarahkannya ke titik akhir pribadi. Untuk informasi selengkapnya, lihat Titik akhir VPC.

Atau, Anda dapat mengaktifkan fungsi Lambda Anda untuk mengakses titik akhir publik Secrets Manager dengan menambahkan gateway NAT atau gateway internet ke VPC Anda, yang memungkinkan lalu lintas dari VPC Anda mencapai titik akhir publik. Ini membuat VPC Anda berisiko lebih besar karena alamat IP untuk gateway dapat diserang dari Internet publik.

(Opsional) Untuk mengakses database atau layanan

Untuk rahasia seperti kunci API, tidak ada database sumber atau layanan yang perlu Anda perbarui bersama dengan rahasianya.

Jika database atau layanan Anda berjalan pada instans Amazon EC2 di VPC, sebaiknya Anda mengonfigurasi fungsi Lambda agar berjalan di VPC yang sama. Kemudian fungsi rotasi dapat berkomunikasi langsung dengan layanan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses VPC.

Untuk mengizinkan fungsi Lambda mengakses database atau layanan, Anda harus memastikan bahwa grup keamanan yang dilampirkan ke fungsi rotasi Lambda Anda memungkinkan koneksi keluar ke database atau layanan. Anda juga harus memastikan bahwa grup keamanan yang dilampirkan ke database atau layanan Anda mengizinkan koneksi masuk dari fungsi rotasi Lambda.

Langkah 5: Tulis kode fungsi rotasi

Fungsi rotasi yang Anda buat di Langkah 1 adalah titik awal untuk fungsi Anda. Anda menulis kode untuk kasus penggunaan spesifik Anda. Untuk fungsi yang dapat memutar ElastiCache rahasia Amazon, Anda dapat menyalin kode dari template yang sesuai yang disediakan oleh Secrets Manager.

Saat Anda menulis fungsi Anda, berhati-hatilah untuk menyertakan pernyataan debugging atau logging. Pernyataan ini dapat menyebabkan informasi dalam fungsi Anda ditulis ke Amazon CloudWatch, jadi Anda perlu memastikan log tidak menyertakan informasi sensitif apa pun yang dikumpulkan selama pengembangan.

Untuk keamanan, Secrets Manager hanya mengizinkan fungsi rotasi Lambda untuk memutar rahasia secara langsung. Fungsi rotasi tidak dapat memanggil fungsi Lambda kedua untuk memutar rahasia.

Untuk contoh pernyataan log, lihat kode the section called “Templat fungsi rotasi” sumber.

Jika Anda menggunakan binari dan pustaka eksternal, misalnya untuk terhubung ke sumber daya, Anda perlu mengelola tambalan dan menyimpannya. up-to-date

Untuk saran debugging, lihat Menguji dan men-debug aplikasi tanpa server.

Untuk membuka fungsi rotasi Lambda Anda untuk mengedit

1. Di konsol Secrets Manager, pilih rahasia Anda.
2. Di bagian konfigurasi Rotasi, di bawah fungsi rotasi Lambda, pilih fungsi rotasi Anda.

   Konsol Lambda terbuka.

- Untuk mengubah kode dalam fungsi, gulir ke bawah ke bagian Sumber kode.
- Untuk MySQL versi 5.7 dan yang lebih tinggi, untuk rotasi pengguna bergantian, untuk mengubah panjang nama pengguna maksimum, di bawah variabel Lingkungan, ubah. `USERNAME_CHARACTER_LIMIT`

Jika fungsi Anda belum memilikinya, salin kode dari file SecretsManagerRotationTemplate.

Ada empat langkah untuk memutar rahasia, yang sesuai dengan empat metode berikut dari fungsi rotasi Lambda.

Metode

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

Di`create_secret`, pertama-tama Anda memeriksa apakah ada rahasia dengan menelepon `get_secret_value`dengan `ClientRequestToken` passed-in. Jika tidak ada rahasia, Anda membuat rahasia baru dengan `create_secret`dan token sebagai`VersionId`. Kemudian Anda dapat menghasilkan nilai rahasia baru dengan `get_random_password`. Anda harus memastikan nilai rahasia baru hanya mencakup karakter yang valid untuk database atau layanan. Kecualikan karakter dengan menggunakan `ExcludeCharacters` parameter. Panggil `put_secret_value`untuk menyimpannya dengan label `AWSPENDING` pementasan. Menyimpan nilai rahasia baru `AWSPENDING` membantu memastikan idempotensi. Jika rotasi gagal karena alasan apa pun, Anda dapat merujuk ke nilai rahasia itu dalam panggilan berikutnya. Lihat Bagaimana cara membuat fungsi Lambda saya idempoten.

Saat Anda menguji fungsi Anda, gunakan AWS CLI untuk melihat tahapan versi: panggil `describe-secret`dan lihat`VersionIdsToStages`.

`set_secret`

Di`set_secret`, Anda mengubah kredensi dalam database atau layanan agar sesuai dengan nilai rahasia baru dalam `AWSPENDING` versi rahasia.

Jika Anda meneruskan pernyataan ke layanan yang menafsirkan pernyataan, seperti database, gunakan parameterisasi kueri Untuk informasi selengkapnya, lihat Lembar Cheat Parameterisasi Kueri di situs web OWASP.

Fungsi rotasi adalah wakil istimewa yang memiliki otorisasi untuk mengakses dan memodifikasi kredensil pelanggan baik dalam rahasia Secrets Manager dan sumber daya target. Untuk mencegah potensi serangan wakil yang membingungkan, Anda perlu memastikan bahwa penyerang tidak dapat menggunakan fungsi tersebut untuk mengakses sumber daya lain. Sebelum Anda memperbarui kredensialnya:

- Periksa apakah kredensi dalam `AWSCURRENT` versi rahasia valid. Jika `AWSCURRENT` kredensialnya tidak valid, tinggalkan upaya rotasi.
- Periksa apakah nilai `AWSCURRENT` dan `AWSPENDING` rahasia adalah untuk sumber daya yang sama. Untuk nama pengguna dan kata sandi, periksa apakah `AWSPENDING` nama pengguna `AWSCURRENT` dan nama pengguna sama.
- Periksa apakah sumber daya layanan tujuan sama. Untuk database, periksa apakah nama `AWSCURRENT` dan `AWSPENDING` host sama.

`test_secret`

Di`test_secret`, Anda menguji `AWSPENDING` versi rahasia dengan menggunakannya untuk mengakses database atau layanan.

`finish_secret`

Di`finish_secret`, Anda gunakan `update_secret_version_stage`untuk memindahkan label pementasan `AWSCURRENT` dari versi rahasia sebelumnya ke versi rahasia baru. Secrets Manager secara otomatis menambahkan label `AWSPREVIOUS` pementasan ke versi sebelumnya, sehingga Anda mempertahankan versi rahasia terakhir yang diketahui.

Langkah selanjutnya

Lihat the section called “Memecahkan masalah rotasi”.

Mengatur rotasi otomatis untuk AWS Secrets Manager rahasia menggunakan AWS CLI

Rotasi adalah proses memperbarui rahasia secara berkala. Ketika Anda memutar rahasia, Anda memperbarui kredensil di kedua rahasia dan database atau layanan yang menjadi tujuan rahasianya.

Secrets Manager menggunakan fungsi Lambda untuk memutar rahasia. Untuk ringkasan, lihat the section called “Cara kerja rotasi”.

Anda juga dapat menggunakan konsol untuk mengatur rotasi. Untuk informasi selengkapnya, lihat Rotasi otomatis (konsol).

Untuk mengatur rotasi menggunakanAWS CLI, jika Anda memutar rahasia Amazon RDS, Amazon Redshift, atau Amazon DocumentDB, Anda harus terlebih dahulu memilih file. the section called “Strategi rotasi” Jika Anda memilih strategi pengguna bergantian, Anda harus menyimpan rahasia terpisah dengan kredensil untuk superuser database. Selanjutnya, Anda menulis kode fungsi rotasi. Secrets Manager menyediakan template tempat Anda dapat mendasarkan fungsi Anda. Kemudian Anda membuat fungsi Lambda dengan kode Anda dan mengatur izin untuk fungsi Lambda dan peran eksekusi Lambda. Langkah selanjutnya adalah memastikan bahwa fungsi rotasi Lambda dapat mengakses Secrets Manager dan database atau layanan Anda melalui jaringan. Akhirnya, Anda mengkonfigurasi rahasia untuk rotasi.

Untuk mengaktifkan rotasi otomatis, Anda harus memiliki izin untuk membuat peran eksekusi IAM dan melampirkan kebijakan izin padanya. Anda membutuhkan keduanya `iam:CreateRole` dan `iam:AttachRolePolicy` izin.

> **Warning**
>
> Pemberian identitas `iam:CreateRole` dan `iam:AttachRolePolicy` izin memungkinkan identitas untuk memberikan izin apa pun kepada diri mereka sendiri.

Langkah:

- (Opsional) Langkah 1: Buat rahasia superuser
- Langkah 2: Tulis kode fungsi rotasi
- Langkah 3: Buat fungsi Lambda dan peran eksekusi
- Langkah 4: Siapkan akses jaringan

- Langkah 5: Konfigurasikan rahasia untuk rotasi
- Langkah selanjutnya

(Opsional) Langkah 1: Buat rahasia superuser

Untuk Amazon RDS, Amazon Redshift, dan Amazon DocumentDB, Secrets Manager menawarkan dua strategi rotasi:

Strategi rotasi pengguna tunggal

Strategi ini memperbarui kredensi untuk satu pengguna dalam satu rahasia. Untuk instans Amazon RDS Db2, karena pengguna tidak dapat mengubah kata sandi mereka sendiri, Anda harus memberikan kredensi admin dalam rahasia terpisah. Ini adalah strategi rotasi paling sederhana, dan cocok untuk sebagian besar kasus penggunaan. Secara khusus, kami menyarankan Anda menggunakan strategi ini untuk kredensil untuk satu kali (ad hoc) atau pengguna interaktif.

Ketika rahasia berputar, koneksi database terbuka tidak terputus. Sementara rotasi sedang terjadi, ada periode waktu singkat antara ketika kata sandi dalam database berubah dan ketika rahasia diperbarui. Selama waktu ini, ada risiko rendah database menolak panggilan yang menggunakan kredenal yang diputar. Anda dapat mengurangi risiko ini dengan strategi coba lagi yang tepat. Setelah rotasi, koneksi baru menggunakan kredensi baru.

Strategi rotasi pengguna bergantian

Strategi ini memperbarui kredensi untuk dua pengguna dalam satu rahasia. Anda membuat pengguna pertama, dan selama rotasi pertama, fungsi rotasi mengkloningnya untuk membuat pengguna kedua. Setiap kali rahasia berputar, fungsi rotasi mengganti kata sandi pengguna mana yang diperbarui. Karena sebagian besar pengguna tidak memiliki izin untuk mengkloning diri mereka sendiri, Anda harus memberikan kredensialnya untuk rahasia lain. `superuser` Sebaiknya gunakan strategi rotasi pengguna tunggal ketika pengguna kloning di database Anda tidak memiliki izin yang sama dengan pengguna asli, dan untuk kredensil untuk pengguna satu kali (ad hoc) atau interaktif.

Strategi ini sesuai untuk database dengan model izin di mana satu peran memiliki tabel database dan peran kedua memiliki izin untuk mengakses tabel database. Ini juga sesuai untuk aplikasi yang membutuhkan ketersediaan tinggi. Jika aplikasi mengambil rahasia selama rotasi, aplikasi masih mendapatkan set kredensil yang valid. Setelah rotasi, keduanya `user` dan `user_clone` kredensialnya valid. Bahkan ada lebih sedikit kemungkinan aplikasi mendapatkan penolakan

selama jenis rotasi ini daripada rotasi pengguna tunggal. Jika database di-host di server farm di mana perubahan kata sandi membutuhkan waktu untuk menyebar ke semua server, ada risiko database menolak panggilan yang menggunakan kredensi baru. Anda dapat mengurangi risiko ini dengan strategi coba lagi yang tepat.

Secrets Manager membuat pengguna kloning dengan izin yang sama dengan pengguna asli. Jika Anda mengubah izin pengguna asli setelah klon dibuat, Anda juga harus mengubah izin pengguna kloning.

> **Important**
>
> Jika Anda memilih strategi pengguna bergantian, Anda harus Buat rahasia database dan menyimpan kredenal superuser database di dalamnya. Anda memerlukan rahasia dengan kredensil superuser karena rotasi mengkloning pengguna pertama, dan sebagian besar pengguna tidak memiliki izin itu.

Langkah 2: Tulis kode fungsi rotasi

Untuk memutar rahasia, Anda memerlukan fungsi rotasi. Fungsi rotasi adalah fungsi Lambda yang dipanggil Secrets Manager untuk memutar rahasia Anda.

Untuk fungsi yang dapat memutar Amazon RDS, Amazon Aurora, Amazon Redshift, Amazon DocumentDB, ElastiCache atau rahasia Amazon, Anda dapat menyalin kode dari template yang sesuai yang disediakan oleh Secrets Manager.

Untuk semua jenis rahasia lainnya, gunakan template rotasi generik sebagai titik awal untuk menulis fungsi rotasi Anda sendiri.

Simpan fungsi rotasi Anda dalam file ZIP *`my-function.zip`* bersama dengan dependensi yang diperlukan.

Saat Anda menulis fungsi Anda, berhati-hatilah untuk menyertakan pernyataan debugging atau logging. Pernyataan ini dapat menyebabkan informasi dalam fungsi Anda ditulis ke Amazon CloudWatch, jadi Anda perlu memastikan log tidak menyertakan informasi sensitif apa pun yang dikumpulkan selama pengembangan.

Untuk keamanan, Secrets Manager hanya mengizinkan fungsi rotasi Lambda untuk memutar rahasia secara langsung. Fungsi rotasi tidak dapat memanggil fungsi Lambda kedua untuk memutar rahasia.

Untuk contoh pernyataan log, lihat kode the section called “Templat fungsi rotasi” sumber.

Jika Anda menggunakan binari dan pustaka eksternal, misalnya untuk terhubung ke sumber daya, Anda perlu mengelola tambalan dan menyimpannya. up-to-date

Untuk saran debugging, lihat Menguji dan men-debug aplikasi tanpa server.

Untuk membuka fungsi rotasi Lambda Anda untuk mengedit

1. Di konsol Secrets Manager, pilih rahasia Anda.
2. Di bagian konfigurasi Rotasi, di bawah fungsi rotasi Lambda, pilih fungsi rotasi Anda.

   Konsol Lambda terbuka.

   - Untuk mengubah kode dalam fungsi, gulir ke bawah ke bagian Sumber kode.
   - Untuk MySQL versi 5.7 dan yang lebih tinggi, untuk rotasi pengguna bergantian, untuk mengubah panjang nama pengguna maksimum, di bawah variabel Lingkungan, ubah. `USERNAME_CHARACTER_LIMIT`

Jika fungsi Anda belum memilikinya, salin kode dari file SecretsManagerRotationTemplate.

Ada empat langkah untuk memutar rahasia, yang sesuai dengan empat metode berikut dari fungsi rotasi Lambda.

Metode

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

Di`create_secret`, pertama-tama Anda memeriksa apakah ada rahasia dengan menelepon `get_secret_value`dengan `ClientRequestToken` passed-in. Jika tidak ada rahasia, Anda membuat rahasia baru dengan `create_secret`dan token sebagai`VersionId`. Kemudian Anda dapat menghasilkan nilai rahasia baru dengan `get_random_password`. Anda harus memastikan nilai rahasia baru hanya mencakup karakter yang valid untuk database atau layanan. Kecualikan karakter dengan menggunakan `ExcludeCharacters` parameter. Panggil

put_secret_valueuntuk menyimpannya dengan label AWSPENDING pementasan. Menyimpan nilai rahasia baru AWSPENDING membantu memastikan idempotensi. Jika rotasi gagal karena alasan apa pun, Anda dapat merujuk ke nilai rahasia itu dalam panggilan berikutnya. Lihat Bagaimana cara membuat fungsi Lambda saya idempoten.

Saat Anda menguji fungsi Anda, gunakan AWS CLI untuk melihat tahapan versi: panggil describe-secretdan lihatVersionIdsToStages.

set_secret

Diset_secret, Anda mengubah kredensi dalam database atau layanan agar sesuai dengan nilai rahasia baru dalam AWSPENDING versi rahasia.

Jika Anda meneruskan pernyataan ke layanan yang menafsirkan pernyataan, seperti database, gunakan parameterisasi kueri Untuk informasi selengkapnya, lihat Lembar Cheat Parameterisasi Kueri di situs web OWASP.

Fungsi rotasi adalah wakil istimewa yang memiliki otorisasi untuk mengakses dan memodifikasi kredensil pelanggan baik dalam rahasia Secrets Manager dan sumber daya target. Untuk mencegah potensi serangan wakil yang membingungkan, Anda perlu memastikan bahwa penyerang tidak dapat menggunakan fungsi tersebut untuk mengakses sumber daya lain. Sebelum Anda memperbarui kredensialnya:

- Periksa apakah kredensi dalam AWSCURRENT versi rahasia valid. Jika AWSCURRENT kredensialnya tidak valid, tinggalkan upaya rotasi.
- Periksa apakah nilai AWSCURRENT dan AWSPENDING rahasia adalah untuk sumber daya yang sama. Untuk nama pengguna dan kata sandi, periksa apakah AWSPENDING nama pengguna AWSCURRENT dan nama pengguna sama.
- Periksa apakah sumber daya layanan tujuan sama. Untuk database, periksa apakah nama AWSCURRENT dan AWSPENDING host sama.

test_secret

Ditest_secret, Anda menguji AWSPENDING versi rahasia dengan menggunakannya untuk mengakses database atau layanan.

`finish_secret`

Di`finish_secret`, Anda gunakan `update_secret_version_stage`untuk memindahkan label pementasan `AWSCURRENT` dari versi rahasia sebelumnya ke versi rahasia baru. Secrets Manager secara otomatis menambahkan label `AWSPREVIOUS` pementasan ke versi sebelumnya, sehingga Anda mempertahankan versi rahasia terakhir yang diketahui.

Langkah 3: Buat fungsi Lambda dan peran eksekusi

Peran eksekusi Lambda adalah peran yang diasumsikan Lambda saat fungsi dipanggil.

Untuk membuat fungsi rotasi Lambda dan peran eksekusi

1. Buat kebijakan kepercayaan untuk peran eksekusi Lambda dan simpan sebagai file JSON. Sebagai contoh, lihat Izin untuk rotasi. Kebijakan harus:
   - Izinkan peran untuk memanggil operasi Secrets Manager pada rahasia.
   - Izinkan peran untuk menggunakan kunci KMS jika rahasia dienkripsi dengan kunci selain. `aws/secretsmanager`
   - Izinkan peran untuk memanggil layanan yang menjadi rahasia itu.
2. Buat peran eksekusi Lambda dan terapkan kebijakan kepercayaan dengan menelepon. `iam create-role`

   ```
   aws iam create-role \
       --role-name rotation-lambda-role \
       --assume-role-policy-document file://trust-policy.json
   ```
3. (Opsional) Untuk rahasia yang berisi kredensi Amazon RDS atau Aurora, jika Anda menggunakan strategi pengguna bergantian dan rahasia pengguna super dikelola oleh Amazon RDS, maka Anda harus mengizinkan fungsi rotasi untuk memanggil API hanya-baca di Amazon RDS sehingga bisa mendapatkan informasi koneksi untuk database. Untuk melakukannya, lampirkan AmazonRDS kebijakan AWS terkelola ReadOnlyAccess ke peran eksekusi fungsi Lambda dengan memanggil. `iam attach-role-policy`

   ```
   aws iam attach-role-policy \
       --policy-arn arn:aws:iam::aws:policy/AmazonRDSReadOnlyAccess \
       --role-name rotation-lambda-role
   ```
4. Buat fungsi Lambda dari file ZIP dengan menelepon. `lambda create-function`

```
aws lambda create-function \
    --function-name my-rotation-function \
    --runtime python3.9 \
    --zip-file fileb://my-function.zip \
    --handler my-handler \
    --role arn:aws:iam::123456789012:role/service-role/rotation-lambda-role
```

5. Tetapkan kebijakan sumber daya pada fungsi Lambda untuk mengizinkan Secrets Manager memanggilnya dengan menelepon. `lambda add-permission` Perintah contoh termasuk `source-account` untuk membantu mencegah Lambda digunakan sebagai wakil yang bingung.

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

Langkah 4: Siapkan akses jaringan

Untuk dapat memutar rahasia, fungsi rotasi Lambda harus dapat mengakses rahasia dan database atau layanan.

Untuk mengakses rahasia

Fungsi rotasi Lambda Anda harus dapat mengakses titik akhir Secrets Manager. Jika fungsi Lambda Anda dapat mengakses internet, maka Anda dapat menggunakan titik akhir publik. Untuk menemukan titik akhir, lihatthe section called “Titik akhir Secrets Manager”.

Jika fungsi Lambda Anda berjalan di VPC yang tidak memiliki akses internet, kami sarankan Anda mengonfigurasi titik akhir pribadi layanan Secrets Manager dalam VPC Anda. VPC Anda kemudian dapat mencegat permintaan yang ditujukan ke titik akhir regional publik dan mengarahkannya ke titik akhir pribadi. Untuk informasi selengkapnya, lihat Titik akhir VPC.

Atau, Anda dapat mengaktifkan fungsi Lambda Anda untuk mengakses titik akhir publik Secrets Manager dengan menambahkan gateway NAT atau gateway internet ke VPC Anda, yang memungkinkan lalu lintas dari VPC Anda mencapai titik akhir publik. Ini membuat VPC Anda berisiko lebih besar karena alamat IP untuk gateway dapat diserang dari Internet publik.

Untuk mengakses database atau layanan

Jika database atau layanan Anda berjalan pada instans Amazon EC2 di VPC, sebaiknya Anda mengonfigurasi fungsi Lambda agar berjalan di VPC yang sama. Kemudian fungsi rotasi dapat berkomunikasi langsung dengan layanan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses VPC.

Untuk mengizinkan fungsi Lambda mengakses database atau layanan, Anda harus memastikan bahwa grup keamanan yang dilampirkan ke fungsi rotasi Lambda Anda memungkinkan koneksi keluar ke database atau layanan. Anda juga harus memastikan bahwa grup keamanan yang dilampirkan ke database atau layanan Anda mengizinkan koneksi masuk dari fungsi rotasi Lambda.

Untuk rotasi pengguna bergantian di mana rahasia superuser dikelola oleh AWS layanan lain, fungsi rotasi Lambda harus dapat memanggil titik akhir layanan untuk mendapatkan informasi koneksi database. Kami menyarankan Anda mengonfigurasi titik akhir VPC untuk layanan database. Lihat informasi yang lebih lengkap di:

- Amazon RDS API dan titik akhir VPC antarmuka di Panduan Pengguna Amazon RDS.
- Bekerja dengan titik akhir VPC di Panduan Manajemen Pergeseran Merah Amazon.

Langkah 5: Konfigurasikan rahasia untuk rotasi

Untuk mengaktifkan rotasi otomatis untuk rahasia Anda, hubungi `rotate-secret`. Anda dapat mengatur jadwal rotasi dengan ekspresi `cron()` atau `rate()` jadwal, dan Anda dapat mengatur durasi jendela rotasi. Anda dapat memutar rahasia sesering setiap empat jam. Untuk informasi selengkapnya, lihat Ekspresi jadwal.

```
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\"}"
```

Langkah selanjutnya

Lihat the section called “Memecahkan masalah rotasi”.

Putar AWS Secrets Manager rahasia segera

Anda hanya dapat memutar rahasia yang telah dikonfigurasi rotasi. Untuk menentukan apakah rahasia telah dikonfigurasi untuk rotasi, di konsol, lihat rahasia dan gulir ke bawah ke bagian konfigurasi Rotasi. Jika status Rotasi Diaktifkan, maka rahasianya dikonfigurasi untuk rotasi. Atau diAWS CLI, panggilan `describe-secret`. Jika respons memiliki `RotationLambdaARN` dan`RotationRules`, maka rahasianya dikonfigurasi untuk rotasi. Jika tidak, Anda dapat mengatur rotasi otomatis:

- Rotasi otomatis untuk rahasia database (konsol)
- Rotasi otomatis (konsol)
- Rotasi otomatis (AWS CLI)

Untuk segera memutar rahasia (konsol)

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih rahasiamu.
3. Pada halaman detail rahasia, di bawah konfigurasi Rotasi, pilih Rotate secret segera.
4. Dalam kotak dialog Rotate secret, pilih Rotate.

AWS CLI

Example Putar rahasia segera

`rotate-secret`Contoh berikut memulai rotasi langsung. Output menunjukkan VersionId versi rahasia baru yang dibuat oleh rotasi. Rahasianya harus sudah memiliki rotasi yang dikonfigurasi.

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

AWS Secrets Manager templat fungsi rotasi

Secrets Manager menyediakan template fungsi rotasi untuk:

- Amazon RDS dan Amazon Aurora
- Amazon DocumentDB (dengan kompatibilitas MongoDB)

- Amazon Redshift
- Amazon ElastiCache
- Jenis rahasia lainnya

Untuk menggunakan template, lihat:

- Putar kredenal Amazon RDS, Amazon Aurora Amazon Redshift, dan Amazon DocumentDB
- Jenis kredensil lainnya (instruksi konsol)
- Jenis kredensil lainnya (instruksi)AWS CLI

Template mendukung Python 3.9.

Untuk menulis fungsi rotasi Anda sendiri, lihat Menulis fungsi rotasi.

Amazon RDS dan Amazon Aurora

Topik

- Amazon RDS Db2 pengguna tunggal
- Amazon RDS Db2 bergantian pengguna
- Amazon RDS MariaDB pengguna tunggal
- Amazon RDS MariaDB pengguna bergantian
- Amazon RDS dan Amazon Aurora MySQL pengguna tunggal
- Amazon RDS dan Amazon Aurora MySQL bergantian pengguna
- Amazon RDS Oracle pengguna tunggal
- Amazon RDS Oracle bergantian pengguna
- Amazon RDS dan Amazon Aurora PostgreSQL pengguna tunggal
- Amazon RDS dan Amazon Aurora PostgreSQL pengguna bergantian
- Amazon RDS Microsoft SQLServer pengguna tunggal
- Amazon RDS Microsoft SQLServer bergantian pengguna

Amazon RDS Db2 pengguna tunggal

- Nama templat: SecretsManager RDSdB2 RotationSingleUser

- Strategi rotasi:Strategi rotasi: pengguna tunggal.
- **`SecretString`**struktur:the section called “Struktur rahasia Amazon RDS Db2”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ SecretsManager RDSdB2 /lambda_function.py RotationSingleUser
- Ketergantungan: python-ibmdb

Amazon RDS Db2 bergantian pengguna

- Nama templat: SecretsManager RDSdB2 RotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**struktur:the section called “Struktur rahasia Amazon RDS Db2”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ SecretsManager RDSdB2 /lambda_function.py RotationMultiUser
- Ketergantungan: python-ibmdb

Amazon RDS MariaDB pengguna tunggal

- Nama template: SecretsManager RDSmariaDB RotationSingleUser
- Strategi rotasi:Strategi rotasi: pengguna tunggal.
- **`SecretString`**struktur:the section called “Struktur rahasia Amazon RDS MariaDB”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ RDSmariaDB /lambda_function.py SecretsManager RotationSingleUser
- Ketergantungan: PyMy SQL 1.0.2

Amazon RDS MariaDB pengguna bergantian

- Nama template: SecretsManager RDSmariaDB RotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**struktur:the section called “Struktur rahasia Amazon RDS MariaDB”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ RDSmariaDB /lambda_function.py SecretsManager RotationMultiUser
- Ketergantungan: PyMy SQL 1.0.2

Amazon RDS dan Amazon Aurora MySQL pengguna tunggal

- Nama template: SecretsManager RdsMySQL RotationSingleUser
- Strategi rotasi:the section called “Pengguna tunggal”.
- **`SecretString`**Struktur yang diharapkan:the section called “Amazon RDS dan Amazon Aurora MySQL struktur rahasia”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ SecretsManager RDSmySQL /lambda_function.py RotationSingleUser
- Ketergantungan: PyMy SQL 1.0.2

Amazon RDS dan Amazon Aurora MySQL bergantian pengguna

- Nama template: SecretsManager RdsMySQL RotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**Struktur yang diharapkan:the section called “Amazon RDS dan Amazon Aurora MySQL struktur rahasia”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ SecretsManager RDSmySQL /lambda_function.py RotationMultiUser
- Ketergantungan: PyMy SQL 1.0.2

Amazon RDS Oracle pengguna tunggal

- Nama template: SecretsManager RDS OracleRotationSingleUser
- Strategi rotasi:the section called “Pengguna tunggal”.
- **`SecretString`**Struktur yang diharapkan:the section called “Struktur rahasia Amazon RDS Oracle”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDS /lambda_function.py OracleRotationSingleUser
- Ketergantungan: python-oracledb 2.0.1

Amazon RDS Oracle bergantian pengguna

- Nama template: SecretsManager RDS OracleRotationMultiUser

- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**Struktur yang diharapkan:the section called “Struktur rahasia Amazon RDS Oracle”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager - lambdas/tree/master/ RDS /lambda_function.py OracleRotationMultiUser
- Ketergantungan: python-oracledb 2.0.1

Amazon RDS dan Amazon Aurora PostgreSQL pengguna tunggal

- Nama template: SecretsManager RDSPostgreSQL RotationSingleUser
- Strategi rotasi:Strategi rotasi: pengguna tunggal.
- **`SecretString`**Struktur yang diharapkan:the section called “Amazon RDS dan Amazon Aurora PostgreSQL struktur rahasia”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ SecretsManager rdsPostgreSQL /lambda_function.py RotationSingleUser
- Ketergantungan: PyGre SQL 5.0.7

Amazon RDS dan Amazon Aurora PostgreSQL pengguna bergantian

- Nama template: SecretsManager RDSPostgreSQL RotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**Struktur yang diharapkan:the section called “Amazon RDS dan Amazon Aurora PostgreSQL struktur rahasia”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ SecretsManager rdsPostgreSQL /lambda_function.py RotationMultiUser
- Ketergantungan: PyGre SQL 5.0.7

Amazon RDS Microsoft SQLServer pengguna tunggal

- Nama template: SecretsManager RDSSQL ServerRotationSingleUser
- Strategi rotasi:the section called “Pengguna tunggal”.
- **`SecretString`**Struktur yang diharapkan:the section called “Amazon RDS Microsoft SQLServer struktur rahasia”.

- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ SecretsManager RDSSQL /lambda_function.py ServerRotationSingleUser
- Ketergantungan: Pymssql 2.2.2

Amazon RDS Microsoft SQLServer bergantian pengguna

- Nama template: SecretsManager RDSSQL ServerRotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**Struktur yang diharapkan:the section called “Amazon RDS Microsoft SQLServer struktur rahasia”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ SecretsManager RDSSQL /lambda_function.py ServerRotationMultiUser
- Ketergantungan: Pymssql 2.2.2

Amazon DocumentDB (dengan kompatibilitas MongoDB)

Amazon DocumentDB pengguna tunggal

- Nama template: SecretsManagerMongo DB RotationSingleUser
- Strategi rotasi:the section called “Pengguna tunggal”.
- **`SecretString`**Struktur yang diharapkan:the section called “Struktur rahasia Amazon DocumentDB”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerMongo -lambdas/pohon/master/ DB /lambda_function.py RotationSingleUser
- Ketergantungan: Pymongo 3.2

Amazon DocumentDB pengguna bergantian

- Nama template: SecretsManagerMongo DB RotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**Struktur yang diharapkan:the section called “Struktur rahasia Amazon DocumentDB”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerMongo -lambdas/pohon/master/ DB /lambda_function.py RotationMultiUser

- Ketergantungan: Pymongo 3.2

Amazon Redshift

Amazon Redshift pengguna tunggal

- Nama template: SecretsManagerRedshiftRotationSingleUser
- Strategi rotasi:the section called “Pengguna tunggal”.
- **`SecretString`**Struktur yang diharapkan: the section called “Struktur rahasia Amazon Redshift” atauthe section called “Amazon Redshift Struktur rahasia tanpa server”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ /lambda_function.py SecretsManagerRedshiftRotationSingleUser
- Ketergantungan: PyGre SQL 5.0.7

Amazon Redshift bergantian pengguna

- Nama template: SecretsManagerRedshiftRotationMultiUser
- Strategi rotasi:the section called “Pengguna bergantian”.
- **`SecretString`**Struktur yang diharapkan: the section called “Struktur rahasia Amazon Redshift” atauthe section called “Amazon Redshift Struktur rahasia tanpa server”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ /lambda_function.py SecretsManagerRedshiftRotationMultiUser
- Ketergantungan: PyGre SQL 5.0.7

Amazon ElastiCache

Untuk menggunakan templat ini, lihat Memutar kata sandi secara otomatis untuk pengguna di Panduan ElastiCache Pengguna Amazon.

- Nama template: SecretsManagerElasticacheUserRotation
- **`SecretString`**Struktur yang diharapkan:the section called “Struktur ElastiCache rahasia Amazon”.
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ /lambda_function.py SecretsManagerElasticacheUserRotation

Jenis rahasia lainnya

Secrets Manager menyediakan template ini sebagai titik awal bagi Anda untuk membuat fungsi rotasi untuk semua jenis rahasia.

- Nama template: SecretsManagerRotationTemplate
- Kode sumber: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ /lambda_function.py SecretsManagerRotationTemplate

Saat Anda menulis fungsi Anda, berhati-hatilah untuk menyertakan pernyataan debugging atau logging. Pernyataan ini dapat menyebabkan informasi dalam fungsi Anda ditulis ke Amazon CloudWatch, jadi Anda perlu memastikan log tidak menyertakan informasi sensitif apa pun yang dikumpulkan selama pengembangan.

Untuk keamanan, Secrets Manager hanya mengizinkan fungsi rotasi Lambda untuk memutar rahasia secara langsung. Fungsi rotasi tidak dapat memanggil fungsi Lambda kedua untuk memutar rahasia.

Untuk contoh pernyataan log, lihat kode the section called “Templat fungsi rotasi” sumber.

Jika Anda menggunakan binari dan pustaka eksternal, misalnya untuk terhubung ke sumber daya, Anda perlu mengelola tambalan dan menyimpannya. up-to-date

Untuk saran debugging, lihat Menguji dan men-debug aplikasi tanpa server.

Ada empat langkah untuk memutar rahasia, yang sesuai dengan empat metode berikut dari fungsi rotasi Lambda.

Metode

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

Di`create_secret`, pertama-tama Anda memeriksa apakah ada rahasia dengan menelepon `get_secret_value`dengan `ClientRequestToken` passed-in. Jika tidak ada rahasia, Anda membuat rahasia baru dengan `create_secret`dan token sebagai`VersionId`. Kemudian

Anda dapat menghasilkan nilai rahasia baru dengan `get_random_password`. Anda harus memastikan nilai rahasia baru hanya mencakup karakter yang valid untuk database atau layanan. Kecualikan karakter dengan menggunakan `ExcludeCharacters` parameter. Panggil `put_secret_value`untuk menyimpannya dengan label `AWSPENDING` pementasan. Menyimpan nilai rahasia baru `AWSPENDING` membantu memastikan idempotensi. Jika rotasi gagal karena alasan apa pun, Anda dapat merujuk ke nilai rahasia itu dalam panggilan berikutnya. Lihat Bagaimana cara membuat fungsi Lambda saya idempoten.

Saat Anda menguji fungsi Anda, gunakan AWS CLI untuk melihat tahapan versi: panggil `describe-secret`dan lihat`VersionIdsToStages`.

`set_secret`

Di`set_secret`, Anda mengubah kredensi dalam database atau layanan agar sesuai dengan nilai rahasia baru dalam `AWSPENDING` versi rahasia.

Jika Anda meneruskan pernyataan ke layanan yang menafsirkan pernyataan, seperti database, gunakan parameterisasi kueri Untuk informasi selengkapnya, lihat Lembar Cheat Parameterisasi Kueri di situs web OWASP.

Fungsi rotasi adalah wakil istimewa yang memiliki otorisasi untuk mengakses dan memodifikasi kredensil pelanggan baik dalam rahasia Secrets Manager dan sumber daya target. Untuk mencegah potensi serangan wakil yang membingungkan, Anda perlu memastikan bahwa penyerang tidak dapat menggunakan fungsi tersebut untuk mengakses sumber daya lain. Sebelum Anda memperbarui kredensialnya:

- Periksa apakah kredensi dalam `AWSCURRENT` versi rahasia valid. Jika `AWSCURRENT` kredensialnya tidak valid, tinggalkan upaya rotasi.
- Periksa apakah nilai `AWSCURRENT` dan `AWSPENDING` rahasia adalah untuk sumber daya yang sama. Untuk nama pengguna dan kata sandi, periksa apakah `AWSPENDING` nama pengguna `AWSCURRENT` dan nama pengguna sama.
- Periksa apakah sumber daya layanan tujuan sama. Untuk database, periksa apakah nama `AWSCURRENT` dan `AWSPENDING` host sama.

`test_secret`

Di`test_secret`, Anda menguji `AWSPENDING` versi rahasia dengan menggunakannya untuk mengakses database atau layanan.

`finish_secret`

Di`finish_secret`, Anda gunakan `update_secret_version_stage`untuk memindahkan label pementasan `AWSCURRENT` dari versi rahasia sebelumnya ke versi rahasia baru. Secrets Manager secara otomatis menambahkan label `AWSPREVIOUS` pementasan ke versi sebelumnya, sehingga Anda mempertahankan versi rahasia terakhir yang diketahui.

Jadwalkan ekspresi dalam rotasi Secrets Manager

Saat Anda mengaktifkan rotasi otomatis, Anda dapat menggunakan ekspresi cron () atau rate () untuk mengatur jadwal untuk memutar rahasia Anda. Dengan ekspresi laju, Anda dapat membuat jadwal rotasi yang berulang pada interval jam atau hari. Dengan ekspresi cron, Anda dapat membuat jadwal rotasi yang lebih detail daripada interval rotasi. Jadwal rotasi Secrets Manager menggunakan zona waktu UTC. Anda dapat memutar rahasia sesering setiap empat jam. Secrets Manager memutar rahasia Anda kapan saja selama jendela rotasi.

Untuk mengaktifkan rotasi, lihat:

- the section called “Rotasi otomatis untuk rahasia database (konsol)”
- the section called “Rotasi otomatis (konsol)”
- the section called “Rotasi otomatis (AWS CLI)”

Ekspresi rate

Ekspresi tingkat Secrets Manager memiliki format berikut, di mana *`Nilai`* adalah bilangan bulat positif dan *`Unit`* dapat berupa`hour`,, `hours`day, atau`days`:

```
rate(Value Unit)
```

Anda dapat memutar rahasia sesering setiap empat jam. Contoh:

- `rate(4 hours)`berarti rahasia diputar setiap empat jam.
- `rate(1 day)`berarti rahasianya diputar setiap hari.
- `rate(10 days)`berarti rahasianya diputar setiap 10 hari.

Untuk tingkat dalam jam, jendela rotasi default dimulai pada tengah malam dan ditutup setelah satu jam. Anda dapat mengatur durasi Jendela untuk mengubah jendela rotasi. Jendela rotasi

tidak boleh meluas ke jendela rotasi berikutnya. Salah satu cara untuk memeriksa ini adalah untuk mengkonfirmasi bahwa jendela rotasi kurang dari atau sama dengan jumlah jam antara rotasi.

Untuk tingkat dalam beberapa hari, jendela rotasi default dimulai pada tengah malam dan ditutup pada akhir hari. Anda dapat mengatur durasi Jendela untuk mengubah jendela rotasi. Jendela rotasi tidak boleh diperpanjang ke hari UTC berikutnya. Salah satu cara untuk memeriksa ini adalah dengan mengonfirmasi bahwa jam mulai ditambah durasi jendela kurang dari atau sama dengan 24 jam.

Ekspresi Cron

Ekspresi cron memiliki format berikut:

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

Ekspresi cron yang mencakup penambahan jam akan disetel ulang setiap hari. Misalnya, `cron(0 4/12 * * ? *)` berarti 4:00 AM, 4:00 PM, dan kemudian hari berikutnya 4:00 AM, 4:00 PM. Jadwal rotasi Secrets Manager menggunakan zona waktu UTC.

Untuk jadwal dalam jam, jendela rotasi default ditutup setelah satu jam. Anda dapat mengatur durasi Jendela untuk mengubah jendela rotasi. Jendela rotasi tidak boleh masuk ke jendela rotasi berikutnya. Anda dapat memutar rahasia sesering setiap empat jam.

Contoh jadwal	Ekspresi
Setiap delapan jam dimulai pada tengah malam.	`cron(0 /8 * * ? *)`
Setiap delapan jam mulai pukul 8:00 pagi.	`cron(0 8/8 * * ? *)`
Setiap sepuluh jam, mulai pukul 2:00 pagi. Jendela rotasi akan dimulai pada 2:00, 12:00, dan 22:00, dan kemudian hari berikutnya pada 2:00, 12:00, dan 22:00.	`cron(0 2/10 * * ? *)`
Setiap hari pukul 10:00 pagi.	`cron(0 10 * * ? *)`
Setiap hari Sabtu pukul 18.00.	`cron(0 18 ? * SAT *)`
Hari pertama setiap bulan pukul 8:00 pagi.	`cron(0 8 1 * ? *)`

Contoh jadwal	Ekspresi
Setiap tiga bulan pada hari Minggu pertama pukul 1:00 pagi.	`cron(0 1 ? 1/3 SUN#1 *)`
Hari terakhir setiap bulan pukul 17:00.	`cron(0 17 L * ? *)`
Senin sampai Jumat pukul 8:00 pagi.	`cron(0 8 ? * MON-FRI *)`
Hari pertama dan ke-15 setiap bulan pukul 16:00.	`cron(0 16 1,15 * ? *)`
Minggu pertama setiap bulan pada tengah malam.	`cron(0 0 ? * SUN#1 *)`

Persyaratan ekspresi cron di Secrets Manager

Secrets Manager memiliki beberapa batasan pada apa yang dapat Anda gunakan untuk ekspresi cron. Ekspresi cron untuk Secrets Manager harus memiliki 0 di bidang menit karena jendela rotasi Secrets Manager dimulai pada jam. Itu harus memiliki * di bidang tahun, karena Secrets Manager tidak mendukung jadwal rotasi yang terpisah lebih dari satu tahun. Tabel berikut menunjukkan opsi yang dapat Anda gunakan.

Bidang	Nilai	Wildcard
Menit	Harus 0	Tidak ada
Jam	0–23	Gunakan/(garis miring ke depan) untuk menentukan kenaikan. Misalnya `2/10` berarti setiap 10 jam dimulai pukul 2:00 pagi. Anda dapat memutar rahasia sesering setiap empat jam.
D ay-of-month	1–31	Gunakan, (koma) untuk memasukkan nilai tambahan. Misalnya `1,15` berarti hari

Bidang	Nilai	Wildcard
		pertama dan ke-15 setiap bulan. Gunakan - (tanda hubung) untuk menentukan rentang. Misalnya `1-15` berarti hari 1 sampai 15 dalam sebulan. Gunakan* (tanda bintang) untuk menyertakan semua nilai di bidang. Misalnya * berarti setiap hari dalam sebulan. Wildcard ? (tanda tanya) menentukan satu atau yang lain. Anda tidak dapat menentukan kolom `Day-of-month` dan `Day-of-week` dalam ekspresi cron yang sama. Jika Anda menentuka n sebuah nilai di salah satu kolom, maka Anda harus menggunakan ? (tanda tanya) di kolom yang lain. Gunakan/(garis miring ke depan) untuk menentuka n kenaikan. Misalnya, `1/2` berarti setiap dua hari dimulai pada hari 1, dengan kata lain, hari 1, 3, 5, dan seterusnya. Gunakan L untuk menentukan hari terakhir bulan itu.

Bidang	Nilai	Wildcard
		Gunakan ***DAY*** L untuk menentukan hari bernama terakhir dalam sebulan. Misalnya SUNL berarti hari Minggu terakhir setiap bulan.
Bulan	1—12 atau JAN—DEC	Gunakan, (koma) untuk memasukkan nilai tambahan. Misalnya, JAN,APR,JUL,OCT berarti Januari, April, Juli, dan Oktober. Gunakan - (tanda hubung) untuk menentukan rentang. Misalnya 1-3 berarti bulan 1 sampai 3 tahun. Gunakan* (tanda bintang) untuk menyertakan semua nilai di bidang. Misalnya * berarti setiap bulan. Gunakan/(garis miring ke depan) untuk menentukan kenaikan. Misalnya, 1/3 berarti setiap bulan ketiga, dimulai pada bulan 1, dengan kata lain bulan 1, 4, 7, dan 10.

Bidang	Nilai	Wildcard
D ay-of-week	1—7 atau SUN—SAT	Gunakan # untuk menentuka n hari dalam seminggu dalam sebulan. Misalnya, `TUE#3` berarti Selasa ketiga setiap bulan. Gunakan, (koma) untuk memasukkan nilai tambahan. Misalnya `1,4` berarti hari pertama dan keempat dalam seminggu. Gunakan - (tanda hubung) untuk menentukan rentang. Misalnya `1-4` berarti hari 1 sampai 4 dalam seminggu. Gunakan* (tanda bintang) untuk menyertakan semua nilai di bidang. Misalnya * berarti setiap hari dalam seminggu. Wildcard ? (tanda tanya) menentukan satu atau yang lain. Anda tidak dapat menentukan kolom `Day-of-month` dan `Day-of-week` dalam ekspresi cron yang sama. Jika Anda menentuka n sebuah nilai di salah satu kolom, maka Anda harus menggunakan ? (tanda tanya) di kolom yang lain.

Bidang	Nilai	Wildcard
		Gunakan/(garis miring ke depan) untuk menentuka n kenaikan. Misalnya, 1/2 berarti setiap hari kedua dalam seminggu, dimulai pada hari pertama, jadi hari 1, 3, 5, dan 7. Gunakan L untuk menentukan hari terakhir dalam seminggu.
Tahun	Harus *	Tidak ada

Memecahkan masalah rotasi AWS Secrets Manager

Untuk banyak layanan, Secrets Manager menggunakan fungsi Lambda untuk memutar rahasia. Untuk informasi selengkapnya, lihat the section called “Cara kerja rotasi”. Fungsi rotasi Lambda berinteraksi dengan database atau layanan rahasianya serta Secrets Manager. Ketika rotasi tidak bekerja seperti yang Anda harapkan, Anda harus terlebih dahulu memeriksa CloudWatch log.

> **Note**
>
> Beberapa layanan dapat mengelola rahasia untuk Anda, termasuk mengelola rotasi otomatis. Untuk informasi selengkapnya, lihat the section called “Rotasi terkelola”.

Untuk melihat CloudWatch log untuk fungsi Lambda Anda

1. Buka konsol Secrets Manager di https://console.aws.amazon.com/secretsmanager/.
2. Pilih rahasia Anda, dan kemudian pada halaman detail, di bawah konfigurasi Rotasi, pilih fungsi rotasi Lambda. Konsol Lambda terbuka.
3. Pada tab Monitor, pilih Log, lalu pilih Lihat log masuk CloudWatch.

   CloudWatch Konsol membuka dan menampilkan log untuk fungsi Anda.

Untuk menafsirkan log

Tidak ada aktivitas setelah “Menemukan kredensyal dalam variabel lingkungan”

Jika tidak ada aktivitas setelah “Ditemukan kredensial dalam variabel lingkungan”, dan durasi tugas panjang, misalnya batas waktu Lambda default 30000ms, maka fungsi Lambda mungkin habis waktu saat mencoba mencapai titik akhir Secrets Manager.

Fungsi rotasi Lambda Anda harus dapat mengakses titik akhir Secrets Manager. Jika fungsi Lambda Anda dapat mengakses internet, maka Anda dapat menggunakan titik akhir publik. Untuk menemukan titik akhir, lihatthe section called “Titik akhir Secrets Manager”.

Jika fungsi Lambda Anda berjalan di VPC yang tidak memiliki akses internet, kami sarankan Anda mengonfigurasi titik akhir pribadi layanan Secrets Manager dalam VPC Anda. VPC Anda kemudian dapat mencegat permintaan yang ditujukan ke titik akhir regional publik dan mengarahkannya ke titik akhir pribadi. Untuk informasi selengkapnya, lihat Titik akhir VPC.

Atau, Anda dapat mengaktifkan fungsi Lambda Anda untuk mengakses titik akhir publik Secrets Manager dengan menambahkan gateway NAT atau gateway internet ke VPC Anda, yang memungkinkan lalu lintas dari VPC Anda mencapai titik akhir publik. Ini membuat VPC Anda berisiko lebih besar karena alamat IP untuk gateway dapat diserang dari Internet publik.

Tidak ada aktivitas setelah “createSecret”

Berikut ini adalah masalah yang dapat menyebabkan rotasi berhenti setelah createSecret:

ACL Jaringan VPC tidak mengizinkan lalu lintas HTTPS masuk dan keluar.

Untuk informasi selengkapnya, lihat Mengontrol lalu lintas ke subnet menggunakan ACL Jaringan di Panduan Pengguna Amazon VPC.

Konfigurasi batas waktu fungsi Lambda terlalu pendek untuk melakukan tugas.

Untuk informasi selengkapnya, lihat Mengonfigurasi opsi fungsi Lambda di Panduan AWS Lambda Pengembang.

Titik akhir VPC Secrets Manager tidak mengizinkan CIDR VPC masuk ke grup keamanan yang ditetapkan.

Untuk informasi selengkapnya, lihat Mengontrol lalu lintas ke sumber daya menggunakan grup keamanan di Panduan Pengguna Amazon VPC.

Kebijakan titik akhir VPC Secrets Manager tidak mengizinkan Lambda menggunakan titik akhir VPC.

Untuk informasi selengkapnya, lihat Titik akhir VPC.

Rahasianya menggunakan rotasi pengguna bergantian, rahasia superuser dikelola oleh Amazon RDS, dan fungsi Lambda tidak dapat mengakses RDS API.

Untuk rotasi pengguna bergantian di mana rahasia superuser dikelola oleh AWS layanan lain, fungsi rotasi Lambda harus dapat memanggil titik akhir layanan untuk mendapatkan informasi koneksi database. Kami menyarankan Anda mengonfigurasi titik akhir VPC untuk layanan database. Lihat informasi yang lebih lengkap di:

- Amazon RDS API dan titik akhir VPC antarmuka di Panduan Pengguna Amazon RDS.
- Bekerja dengan titik akhir VPC di Panduan Manajemen Pergeseran Merah Amazon.

Kesalahan: “Akses ke KMS tidak diizinkan”

Jika Anda lihat`ClientError: An error occurred (AccessDeniedException) when calling the GetSecretValue operation: Access to KMS is not allowed`, fungsi rotasi tidak memiliki izin untuk mendekripsi rahasia menggunakan kunci KMS yang digunakan untuk mengenkripsi rahasia. Mungkin ada kondisi dalam kebijakan izin yang membatasi konteks enkripsi ke rahasia tertentu. Untuk informasi tentang izin yang diperlukan, lihatthe section called “Pernyataan kebijakan untuk kunci yang dikelola pelanggan”.

Kesalahan: “Kunci hilang dari JSON rahasia”

Fungsi rotasi Lambda membutuhkan nilai rahasia berada dalam struktur JSON tertentu. Jika Anda melihat kesalahan ini, maka JSON mungkin kehilangan kunci yang coba diakses oleh fungsi rotasi. Untuk informasi tentang struktur JSON untuk setiap jenis rahasia, lihatthe section called “Struktur JSON dari sebuah rahasia”.

Kesalahan: “setSecret: Tidak dapat masuk ke database”

Berikut ini adalah masalah yang dapat menyebabkan kesalahan ini:

Fungsi rotasi tidak dapat mengakses database.

- Jika durasi tugas panjang, misalnya lebih dari 5000 ms, maka fungsi rotasi Lambda mungkin tidak dapat mengakses database melalui jaringan.

- Jika database atau layanan Anda berjalan pada instans Amazon EC2 di VPC, sebaiknya Anda mengonfigurasi fungsi Lambda agar berjalan di VPC yang sama. Kemudian fungsi rotasi dapat berkomunikasi langsung dengan layanan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses VPC.

- Untuk mengizinkan fungsi Lambda mengakses database atau layanan, Anda harus memastikan bahwa grup keamanan yang dilampirkan ke fungsi rotasi Lambda Anda memungkinkan koneksi keluar ke database atau layanan. Anda juga harus memastikan bahwa grup keamanan yang dilampirkan ke database atau layanan Anda mengizinkan koneksi masuk dari fungsi rotasi Lambda.

Kredensi dalam rahasia tidak benar.

- Jika durasi tugas pendek, maka fungsi rotasi Lambda mungkin tidak dapat mengautentikasi dengan kredensyal dalam rahasia. Periksa kredensyal dengan masuk secara manual dengan informasi dalam `AWSCURRENT` dan `AWSPREVIOUS` versi rahasia menggunakan perintah. AWS CLI `get-secret-value`

Database digunakan **`scram-sha-256`** untuk mengenkripsi kata sandi.

- Jika database Anda adalah Aurora PostgreSQL versi 13 atau yang lebih baru dan digunakan `scram-sha-256` untuk mengenkripsi kata sandi, tetapi fungsi rotasi menggunakan `libpq` versi 9 atau lebih lama yang tidak mendukung`scram-sha-256`, maka fungsi rotasi tidak dapat terhubung ke database.

Untuk menentukan pengguna database mana yang menggunakan **`scram-sha-256`** enkripsi

- Lihat Memeriksa pengguna dengan kata sandi non-Scram di blog Otentikasi SCRAM di RDS untuk PostgreSQL 13.

Untuk menentukan versi fungsi rotasi **`libpq`** Anda yang digunakan

1. Di komputer berbasis Linux, di konsol Lambda, navigasikan ke fungsi rotasi Anda dan unduh bundel penerapan. Buka kompres file zip ke direktori kerja.
2. Pada baris perintah, di direktori kerja, jalankan:

   ```
   readelf -a libpq.so.5 | grep RUNPATH
   ```

3. Jika Anda melihat string *PostgreSQL-9.4.x*, atau versi utama kurang dari 10, maka fungsi rotasi tidak mendukung`scram-sha-256`.
   - Output untuk fungsi rotasi yang tidak mendukung`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]
     ```

   - Output untuk fungsi rotasi yang mendukung`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**
>
> Jika Anda mengatur rotasi rahasia otomatis sebelum 30 Desember 2021, fungsi rotasi Anda menggabungkan versi lama `libpq` yang tidak mendukung`scram-sha-256`. Untuk mendukung`scram-sha-256`, Anda perlu membuat ulang fungsi rotasi Anda.

Database membutuhkan akses SSL/TLS.

Jika database Anda memerlukan koneksi SSL/TLS, tetapi fungsi rotasi menggunakan koneksi yang tidak terenkripsi, maka fungsi rotasi tidak dapat terhubung ke database. Fungsi rotasi untuk Amazon RDS (kecuali Oracle dan Db2) dan Amazon DocumentDB secara otomatis menggunakan Secure Socket Layer (SSL) atau Transport Layer Security (TLS) untuk terhubung ke database Anda, jika tersedia. Jika tidak, mereka menggunakan koneksi yang tidak terenkripsi.

> **Note**
>
> Jika Anda mengatur rotasi rahasia otomatis sebelum 20 Desember 2021, fungsi rotasi Anda mungkin didasarkan pada templat lama yang tidak mendukung SSL/TLS. Untuk mendukung koneksi yang menggunakan SSL/TLS, Anda perlu membuat ulang fungsi rotasi Anda.

Untuk menentukan kapan fungsi rotasi Anda dibuat

1. Di konsol Secrets Manager https://console.aws.amazon.com/secretsmanager/, buka rahasia Anda. Di bagian konfigurasi Rotasi, di bawah fungsi rotasi Lambda, Anda melihat fungsi Lambda ARN, misalnya,. `arn:aws:lambda:aws-region:123456789012:function:SecretsManagerMyRotationFunction` Salin nama fungsi dari akhir ARN, dalam contoh ini. `SecretsManagerMyRotationFunction`
2. Di AWS Lambda konsol https://console.aws.amazon.com/lambda/, di bawah Fungsi, tempel nama fungsi Lambda Anda di kotak pencarian, pilih Enter, lalu pilih fungsi Lambda.
3. Di halaman detail fungsi, pada tab Konfigurasi, di bawah Tag, salin nilai di sebelah kunci aws:cloudformation:stack-name.
4. Di AWS CloudFormation konsol https://console.aws.amazon.com/cloudformation, di bawah Tumpukan, tempel nilai kunci di kotak pencarian, lalu pilih Enter.
5. Daftar tumpukan menyaring sehingga hanya tumpukan yang membuat fungsi rotasi Lambda yang muncul. Di kolom Tanggal dibuat, lihat tanggal tumpukan dibuat. Ini adalah tanggal fungsi rotasi Lambda dibuat.

Kesalahan: “Tidak dapat mengimpor modul 'lambda_function'”

Anda mungkin menerima kesalahan ini jika Anda menjalankan fungsi Lambda sebelumnya yang secara otomatis ditingkatkan dari Python 3.7 ke versi Python yang lebih baru. Untuk mengatasi

kesalahan, Anda dapat mengubah versi fungsi Lambda kembali ke Python 3.7, dan kemudian. the section called “Tingkatkan fungsi rotasi yang ada dari Python 3.7 ke 3.9” Untuk informasi selengkapnya, lihat Mengapa rotasi fungsi Secrets Manager Lambda saya gagal dengan kesalahan “modul pg tidak ditemukan”? di AWS re:post.

Tingkatkan fungsi rotasi yang ada dari Python 3.7 ke 3.9

Beberapa fungsi rotasi yang dibuat sebelum November 2022 menggunakan Python 3.7. AWS SDK untuk Python berhenti mendukung Python 3.7 pada Desember 2023. Untuk informasi selengkapnya, lihat Pembaruan kebijakan dukungan Python untuk AWS SDK dan Alat. Untuk beralih ke fungsi rotasi baru yang menggunakan Python 3.9, Anda dapat menambahkan properti runtime ke fungsi rotasi yang ada atau membuat ulang fungsi rotasi.

Untuk menemukan fungsi rotasi Lambda mana yang menggunakan Python 3.7

1. Masuk ke AWS Management Console dan buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Dalam daftar Fungsi, filter untuk**`SecretsManager`**.
3. Dalam daftar fungsi yang difilter, di bawah Runtime, cari Python 3.7.

Untuk meningkatkan ke Python 3.9:

- Opsi 1: Buat ulang fungsi rotasi menggunakan AWS CloudFormation
- Opsi 2: Perbarui runtime untuk fungsi rotasi yang ada menggunakan AWS CloudFormation
- Opsi 3: Untuk AWS CDK pengguna, tingkatkan perpustakaan CDK

Opsi 1: Buat ulang fungsi rotasi menggunakan AWS CloudFormation

Saat Anda menggunakan konsol Secrets Manager untuk mengaktifkan rotasi, Secrets Manager menggunakan AWS CloudFormation untuk membuat sumber daya yang diperlukan, termasuk fungsi rotasi Lambda. Jika Anda menggunakan konsol untuk mengaktifkan rotasi, atau Anda membuat fungsi rotasi menggunakan AWS CloudFormation tumpukan, Anda dapat menggunakan AWS CloudFormation tumpukan yang sama untuk membuat ulang fungsi rotasi dengan nama baru. Fungsi baru menggunakan versi Python yang lebih baru.

Untuk menemukan AWS CloudFormation tumpukan yang menciptakan fungsi rotasi

- Pada halaman detail fungsi Lambda, pada tab Konfigurasi, pilih Tag. Lihat ARN di sebelah aws:cloudformation:stack-id.

  Nama tumpukan disematkan di ARN, seperti yang ditunjukkan pada contoh berikut.

  - ARN: `arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`
  - Nama tumpukan: **`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**

Untuk membuat ulang fungsi rotasi ()AWS CloudFormation

1. Di AWS CloudFormation, cari tumpukan berdasarkan nama, lalu pilih Perbarui.

   Jika kotak dialog muncul merekomendasikan Anda memperbarui tumpukan root, pilih Buka tumpukan root, lalu pilih Perbarui.
2. Pada halaman Update stack, pilih Edit template di desainer, lalu pilih View in Designer.
3. Dalam desainer, dalam kode template, di`SecretRotationScheduleHostedRotationLambda`, ganti nilai untuk `"functionName": "SecretsManagerTestRotationRDS"` dengan nama fungsi baru, misalnya di JSON, **`"functionName": "SecretsManagerTestRotationRDSupdated"`**
4. Lanjutkan melalui alur kerja AWS CloudFormation tumpukan dan kemudian pilih Kirim.

Opsi 2: Perbarui runtime untuk fungsi rotasi yang ada menggunakan AWS CloudFormation

Saat Anda menggunakan konsol Secrets Manager untuk mengaktifkan rotasi, Secrets Manager menggunakan AWS CloudFormation untuk membuat sumber daya yang diperlukan, termasuk fungsi rotasi Lambda. Jika Anda menggunakan konsol untuk mengaktifkan rotasi, atau Anda membuat fungsi rotasi menggunakan AWS CloudFormation tumpukan, Anda dapat menggunakan AWS CloudFormation tumpukan yang sama untuk memperbarui runtime untuk fungsi rotasi.

Untuk menemukan AWS CloudFormation tumpukan yang menciptakan fungsi rotasi

- Pada halaman detail fungsi Lambda, pada tab Konfigurasi, pilih Tag. Lihat ARN di sebelah aws:cloudformation:stack-id.

  Nama tumpukan disematkan di ARN, seperti yang ditunjukkan pada contoh berikut.

  - ARN: `arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`
  - Nama tumpukan: **`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**

Untuk memperbarui runtime untuk fungsi rotasi ()AWS CloudFormation

1. Di AWS CloudFormation, cari tumpukan berdasarkan nama, lalu pilih Perbarui.

   Jika kotak dialog muncul merekomendasikan Anda memperbarui tumpukan root, pilih Buka tumpukan root, lalu pilih Perbarui.
2. Pada halaman Update stack, pilih Edit template di desainer, lalu pilih View in Designer.
3. Di desainer, di template JSON, untuk, di bawah`SecretRotationScheduleHostedRotationLambda`, di bawah `PropertiesParameters`, tambahkan **`"runtime": "python3.9"`**
4. Lanjutkan melalui alur kerja AWS CloudFormation tumpukan dan kemudian pilih Kirim.

Opsi 3: Untuk AWS CDK pengguna, tingkatkan perpustakaan CDK

Jika Anda menggunakan versi AWS CDK sebelumnya v2.94.0 untuk mengatur rotasi rahasia Anda, Anda dapat memperbarui fungsi Lambda dengan memutakhirkan ke v2.94.0 atau yang lebih baru. Untuk informasi selengkapnya, lihat Panduan Pengembang AWS Cloud Development Kit (AWS CDK) v2.

AWS Secrets Managerrahasia yang dikelola oleh AWS layanan lain

Banyak AWS layanan menyimpan dan menggunakan rahasia diAWS Secrets Manager. Dalam beberapa kasus, rahasia ini adalah rahasia yang dikelola, yang berarti bahwa layanan yang membuatnya membantu mengelolanya. Misalnya, beberapa rahasia terkelola menyertakan rotasi terkelola, jadi Anda tidak perlu mengonfigurasi rotasi sendiri. Layanan pengelolaan mungkin juga membatasi Anda untuk memperbarui rahasia atau menghapusnya tanpa periode pemulihan, yang membantu mencegah pemadaman karena layanan pengelolaan bergantung pada rahasianya.

Rahasia terkelola menggunakan konvensi penamaan yang menyertakan ID layanan pengelolaan untuk membantu mengidentifikasinya.

```
Secret name: ServiceID!MySecret
Secret ARN : arn:aws:us-east-1:ServiceID!MySecret-a1b2c3
```

ID untuk layanan yang mengelola rahasia

- `appflow` – the section called “Amazon AppFlow”
- `databrew` – the section called “AWS Glue DataBrew”
- `datasync` – the section called “AWS DataSync”
- `directconnect` – the section called “AWS Direct Connect”
- `ecs-sc` – the section called “Amazon Elastic Container Service”
- `events` – the section called “Amazon EventBridge”
- `marketplace-deployment` – the section called “AWS Marketplace”
- `opsworks-cm` – the section called “AWS OpsWorks for Chef Automate”
- `rds` – the section called “Amazon RDS dan Aurora”
- `redshift` – the section called “Amazon Redshift”
- `sqlworkbench` – the section called “Editor kueri Amazon Redshift v2”

Untuk menemukan rahasia yang dikelola oleh AWS layanan lain, lihat Menemukan rahasia yang dikelola.

Untuk daftar lengkap layanan yang menggunakan rahasia, lihatthe section called “AWS Layanan yang Menggunakan AWS Secrets Manager Rahasia”.

Amazon AppFlow

Di Amazon AppFlow, saat Anda mengonfigurasi aplikasi SaaS sebagai sumber atau tujuan, Anda membuat koneksi. Ini termasuk informasi yang diperlukan untuk menghubungkan ke aplikasi SaaS, seperti token otentikasi, nama pengguna, dan kata sandi. Amazon AppFlow menyimpan data koneksi Anda dalam rahasia yang dikelola Secrets Manager dengan awalan`appflow`. Biaya penyimpanan rahasia sudah termasuk dengan biaya untuk Amazon AppFlow. Untuk informasi selengkapnya, lihat Perlindungan data AppFlow di Amazon di Panduan AppFlow Pengguna Amazon.

AWS Glue DataBrew

AWS Glue DataBrewmenyediakan `DETERMINISTIC_DECRYPT`, `DETERMINISTIC_ENCRYPT`, dan langkah-langkah `CRYPTOGRAPHIC_HASH`resep untuk melakukan transformasi pada informasi yang dapat diidentifikasi secara pribadi (PII) dalam kumpulan data, yang menggunakan kunci enkripsi yang disimpan dalam rahasia Secrets Manager. Jika Anda menggunakan rahasia DataBrew default untuk menyimpan kunci enkripsi, DataBrew buat rahasia terkelola dengan awalan`databrew`. Biaya penyimpanan rahasia sudah termasuk dengan biaya untuk menggunakan DataBrew.

AWS DataSync

Untuk mengumpulkan informasi tentang sistem penyimpanan lokal, AWS DataSync Discovery menggunakan kredensyal untuk antarmuka manajemen sistem penyimpanan. DataSync menyimpan kredensyal tersebut dalam rahasia yang dikelola Secrets Manager dengan awalan. `datasync` Anda dikenakan biaya untuk rahasia itu. Untuk informasi selengkapnya, lihat Menambahkan sistem penyimpanan lokal ke DataSync Discovery di Panduan AWS DataSync Pengguna.

AWS Direct Connect

AWS Direct Connectmenyimpan nama kunci asosiasi konektivitas dan key pair asosiasi konektivitas (pasangan CKN/CAK) dalam rahasia terkelola dengan awalan. `directconnect` Biaya rahasia sudah termasuk dengan biaya untukAWS Direct Connect. Untuk memperbarui rahasia, Anda harus menggunakan AWS Direct Connect bukan Secrets Manager. Untuk informasi selengkapnya, lihat Mengaitkan CKN/CAK MacSec dengan LAG di Panduan Pengguna. AWS Direct Connect

Amazon Elastic Container Service

Saat Anda menggunakan Amazon ECS Service Connect, Amazon ECS menggunakan rahasia Secrets Manager untuk menyimpan sertifikat AWS Private Certificate Authority TLS. Biaya penyimpanan rahasia sudah termasuk dengan biaya untuk Amazon ECS. Untuk memperbarui rahasia, Anda harus menggunakan Amazon ECS daripada Secrets Manager. Untuk informasi selengkapnya, lihat TLS dengan Service Connect di Panduan Pengembang Layanan Amazon Elastic Container.

Amazon EventBridge

Saat Anda membuat tujuan Amazon EventBridge API, EventBridge menyimpan koneksi untuk itu dalam rahasia yang dikelola Secrets Manager dengan awalan`events`. Biaya penyimpanan rahasia sudah termasuk dengan biaya untuk menggunakan tujuan API. Untuk memperbarui rahasia, Anda harus menggunakan EventBridge bukan Secrets Manager. Untuk informasi selengkapnya, lihat tujuan API di Panduan EventBridge Pengguna Amazon.

AWS Marketplace

Ketika Anda menggunakan AWS Marketplace Quick Launch, AWS Marketplace mendistribusikan perangkat lunak Anda bersama dengan kunci lisensi. AWS Marketplacemenyimpan kunci lisensi di akun Anda sebagai rahasia yang dikelola Secrets Manager. Biaya penyimpanan rahasia sudah termasuk dengan biaya untukAWS Marketplace. Untuk memperbarui rahasia, Anda harus menggunakan AWS Marketplace bukan Secrets Manager. Untuk informasi selengkapnya, lihat Mengkonfigurasi Peluncuran Cepat di Panduan AWS Marketplace Penjual.

AWS OpsWorks for Chef Automate

Saat Anda membuat server baruAWS OpsWorks CM, OpsWorks CM menyimpan informasi untuk server dalam rahasia yang dikelola Secrets Manager dengan awalan`opsworks-cm`. Biaya rahasia sudah termasuk dalam biaya untukAWS OpsWorks. Untuk informasi selengkapnya, lihat Integrasi dengan AWS Secrets Manager di Panduan AWS OpsWorks Pengguna.

Amazon RDS dan Aurora

Untuk mengelola kredensyal pengguna master untuk Amazon Relational Database Service (Amazon RDS), termasuk Aurora, Amazon RDS dapat membuat rahasia terkelola untuk Anda. Anda dikenakan

biaya untuk rahasia itu. Amazon RDS juga mengelola rotasi untuk kredensyal ini. Untuk informasi selengkapnya, lihat Manajemen kata sandi dengan Amazon RDS dan AWS Secrets Manager di Panduan Pengguna Amazon RDS dan manajemen Kata Sandi dengan Amazon Aurora AWS Secrets Manager dan di Panduan Pengguna Amazon Aurora.

Untuk kredensyal Amazon RDS lainnya, lihat. the section called “Buat rahasia database”

Amazon Redshift

Untuk mengelola kredensyal admin untuk Amazon Redshift, Amazon Redshift dapat membuat rahasia terkelola untuk Anda. Anda dikenakan biaya untuk rahasia itu. Amazon Redshift juga mengelola rotasi untuk kredensyal ini. Untuk informasi selengkapnya, lihat Mengelola kata sandi admin Amazon Redshift menggunakan AWS Secrets Manager dalam Panduan Manajemen Amazon Redshift.

Untuk kredensyal Amazon Redshift lainnya, lihat. the section called “Buat rahasia database” Untuk menggunakan rahasia kredensyal saat Anda memanggil Data API, lihat Menggunakan Amazon Redshift Data API. Untuk menggunakan rahasia saat Anda menggunakan editor kueri Amazon Redshift untuk menyambung ke database, lihat Menanyakan database menggunakan editor kueri di Panduan Manajemen Amazon Redshift dan. the section called “Editor kueri Amazon Redshift v2”

Editor kueri Amazon Redshift v2

Saat Anda menggunakan editor kueri Amazon Redshift v2 untuk menyambung ke database, Amazon Redshift dapat menyimpan kredensyal Anda dalam rahasia yang dikelola Secrets Manager dengan awalan. `sqlworkbench` Biaya penyimpanan rahasia sudah termasuk dengan biaya untuk menggunakan Amazon Redshift. Untuk memperbarui rahasia, Anda harus menggunakan Amazon Redshift daripada Secrets Manager. Untuk informasi selengkapnya, lihat Bekerja dengan editor kueri v2 di Panduan Manajemen Amazon Redshift.

Menggunakan titik akhir AWS Secrets Manager VPC

Kami menyarankan Anda menjalankan infrastruktur sebanyak mungkin di jaringan pribadi yang tidak dapat diakses dari internet publik. Anda dapat membuat koneksi pribadi antara VPC dan Secrets Manager Anda dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses Secrets Manager API secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan Secrets Manager API. Lalu lintas antara VPC dan Secrets Manager Anda tidak meninggalkan jaringan. AWS Untuk informasi selengkapnya, lihat Antarmuka VPC endpoint (AWS PrivateLink) dalam Panduan Pengguna Amazon VPC.

Ketika Secrets Manager memutar rahasia dengan menggunakan fungsi rotasi Lambda, misalnya rahasia yang berisi kredensi database, fungsi Lambda membuat permintaan ke database dan Secrets Manager. Saat Anda mengaktifkan rotasi otomatis menggunakan konsol, Secrets Manager membuat fungsi Lambda di VPC yang sama dengan database Anda. Kami menyarankan Anda membuat titik akhir Secrets Manager di VPC yang sama sehingga permintaan dari fungsi rotasi Lambda ke Secrets Manager tidak meninggalkan jaringan Amazon.

Jika Anda mengaktifkan DNS pribadi untuk titik akhir, Anda dapat membuat permintaan API ke Secrets Manager menggunakan nama DNS default untuk Wilayah, misalnya,. `secretsmanager.us-east-1.amazonaws.com` Untuk informasi selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Anda dapat memastikan bahwa permintaan ke Secrets Manager berasal dari akses VPC dengan menyertakan kondisi dalam kebijakan izin Anda. Untuk informasi selengkapnya, lihat the section called “Contoh: Izin dan VPC”.

Anda dapat menggunakan AWS CloudTrail log untuk mengaudit penggunaan rahasia Anda melalui titik akhir VPC.

Untuk membuat titik akhir VPC untuk Secrets Manager

1. Lihat Membuat titik akhir antarmuka di Panduan Pengguna Amazon VPC. Gunakan nama layanan: com.amazonaws. *`wilayah .secretsmanager`*
2. Untuk mengontrol akses ke titik akhir, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir.

Subnet bersama

Anda tidak dapat membuat, mendeskripsikan, memodifikasi, atau menghapus titik akhir VPC di subnet yang dibagikan dengan Anda. Namun, Anda dapat menggunakan titik akhir VPC di subnet yang dibagikan dengan Anda. Untuk informasi tentang berbagi VPC, lihat Membagikan VPC Anda dengan akun lain di Panduan Pengguna Amazon Virtual Private Cloud.

Buat AWS Secrets Manager rahasia di AWS CloudFormation

Anda dapat membuat rahasia dalam CloudFormation tumpukan dengan menggunakan `AWS::SecretsManager::Secret` sumber daya dalam CloudFormation template, seperti yang ditunjukkan padaBuat rahasia.

Untuk membuat rahasia admin untuk Amazon RDS atau Aurora, kami sarankan Anda `ManageMasterUserPassword` menggunakannya. `AWS::RDS::DBCluster` Kemudian Amazon RDS menciptakan rahasia dan mengelola rotasi untuk Anda. Untuk informasi selengkapnya, lihat Rotasi terkelola.

Untuk kredensi Amazon Redshift dan Amazon DocumentDB, pertama-tama buat rahasia dengan kata sandi yang dihasilkan oleh Secrets Manager, dan kemudian gunakan referensi dinamis untuk mengambil nama pengguna dan kata sandi dari rahasia untuk digunakan sebagai kredensi untuk database baru. Selanjutnya, gunakan `AWS::SecretsManager::SecretTargetAttachment` sumber daya untuk menambahkan detail tentang database ke rahasia yang dibutuhkan Secrets Manager untuk memutar rahasia. Akhirnya, untuk mengaktifkan rotasi otomatis, gunakan `AWS::SecretsManager::RotationSchedule` sumber daya dan sediakan fungsi rotasi dan jadwal. Lihat contoh berikut:

- Buat rahasia dengan kredensi Amazon Redshift
- Buat rahasia dengan kredensi Amazon DocumentDB

Untuk melampirkan kebijakan sumber daya ke rahasia Anda, gunakan `AWS::SecretsManager::ResourcePolicy` sumber daya.

Untuk informasi tentang membuat sumber dayaAWS CloudFormation, lihat Mempelajari dasar-dasar templat di Panduan AWS CloudFormation Pengguna. Anda juga dapat menggunakanAWS Cloud Development Kit (AWS CDK). Untuk informasi selengkapnya, lihat AWS Secrets ManagerMembangun Perpustakaan.

Buat AWS Secrets Manager rahasia dengan AWS CloudFormation

Contoh ini menciptakan rahasia bernama**`CloudFormationCreatedSecret-a1b2c3d4e5f6`**. Nilai rahasianya adalah JSON berikut, dengan kata sandi 32 karakter yang dihasilkan saat rahasia dibuat.

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

Contoh ini menggunakan CloudFormation sumber daya berikut:

- AWS::SecretsManager::Secret

Untuk informasi tentang membuat sumber dayaAWS CloudFormation, lihat Mempelajari dasar-dasar templat di Panduan AWS CloudFormation Pengguna.

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
```

Buat AWS Secrets Manager rahasia dengan rotasi otomatis dan instans Amazon RDS MySQL DB dengan AWS CloudFormation

Untuk membuat rahasia admin untuk Amazon RDS atau Aurora, kami sarankan Anda `ManageMasterUserPassword` menggunakan, seperti yang ditunjukkan pada contoh Create a Secrets Manager rahasia untuk kata sandi utama di. `AWS::RDS::DBCluster` Kemudian Amazon RDS menciptakan rahasia dan mengelola rotasi untuk Anda. Untuk informasi selengkapnya, lihat Rotasi terkelola.

Buat AWS Secrets Manager rahasia dan cluster Amazon Redshift dengan AWS CloudFormation

Untuk membuat rahasia admin untuk Amazon Redshift, kami sarankan Anda menggunakan contoh di `AWS::Redshift::Cluster`dan. `AWS::RedshiftServerless::Namespace`

Buat AWS Secrets Manager rahasia dan instance Amazon DocumentDB dengan AWS CloudFormation

Contoh ini membuat rahasia dan instance Amazon DocumentDB menggunakan kredensi dalam rahasia sebagai pengguna dan kata sandi. Rahasianya memiliki kebijakan berbasis sumber daya terlampir yang mendefinisikan siapa yang dapat mengakses rahasia tersebut. Template juga membuat fungsi rotasi Lambda dari Templat fungsi rotasi dan mengonfigurasi rahasia untuk memutar secara otomatis antara pukul 08:00 dan 10:00 UTC pada hari pertama setiap bulan. Sebagai praktik keamanan terbaik, instancenya ada di Amazon VPC.

Contoh ini menggunakan CloudFormation sumber daya berikut untuk Secrets Manager:

- `AWS::SecretsManager::Secret`
- `AWS::SecretsManager::SecretTargetAttachment`
- `AWS::SecretsManager::RotationSchedule`

Untuk informasi tentang membuat sumber dayaAWS CloudFormation, lihat Mempelajari dasar-dasar templat di Panduan AWS CloudFormation Pengguna.

JSON

```
{
   "AWSTemplateFormatVersion":"2010-09-09",
   "Transform":"AWS::SecretsManager-2020-07-23",
   "Resources":{
      "TestVPC":{
         "Type":"AWS::EC2::VPC",
         "Properties":{
            "CidrBlock":"10.0.0.0/16",
            "EnableDnsHostnames":true,
            "EnableDnsSupport":true
         }
      },
      "TestSubnet01":{
         "Type":"AWS::EC2::Subnet",
         "Properties":{
            "CidrBlock":"10.0.96.0/19",
            "AvailabilityZone":{
               "Fn::Select":[
                  "0",
                  {
                     "Fn::GetAZs":{
                        "Ref":"AWS::Region"
                     }
                  }
               ]
            },
            "VpcId":{
               "Ref":"TestVPC"
            }
         }
      },
      "TestSubnet02":{
         "Type":"AWS::EC2::Subnet",
         "Properties":{
            "CidrBlock":"10.0.128.0/19",
            "AvailabilityZone":{
               "Fn::Select":[
                  "1",
                  {
                     "Fn::GetAZs":{
                        "Ref":"AWS::Region"
```

```
                    }
                 }
              ]
           },
           "VpcId":{
              "Ref":"TestVPC"
           }
        }
     },
     "SecretsManagerVPCEndpoint":{
        "Type":"AWS::EC2::VPCEndpoint",
        "Properties":{
           "SubnetIds":[
              {
                 "Ref":"TestSubnet01"
              },
              {
                 "Ref":"TestSubnet02"
              }
           ],
           "SecurityGroupIds":[
              {
                 "Fn::GetAtt":[
                    "TestVPC",
                    "DefaultSecurityGroup"
                 ]
              }
           ],
           "VpcEndpointType":"Interface",
           "ServiceName":{
              "Fn::Sub":"com.amazonaws.${AWS::Region}.secretsmanager"
           },
           "PrivateDnsEnabled":true,
           "VpcId":{
              "Ref":"TestVPC"
           }
        }
     },
     "MyDocDBClusterRotationSecret":{
        "Type":"AWS::SecretsManager::Secret",
        "Properties":{
           "GenerateSecretString":{
              "SecretStringTemplate":"{\"username\": \"someadmin\",\"ssl\": true}",
              "GenerateStringKey":"password",
```

```
               "PasswordLength":16,
               "ExcludeCharacters":"\"@/\\"
            },
            "Tags":[
               {
                  "Key":"AppName",
                  "Value":"MyApp"
               }
            ]
         }
      },
      "MyDocDBCluster":{
         "Type":"AWS::DocDB::DBCluster",
         "Properties":{
            "DBSubnetGroupName":{
               "Ref":"MyDBSubnetGroup"
            },
            "MasterUsername":{
               "Fn::Sub":"{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::username}}"
            },
            "MasterUserPassword":{
               "Fn::Sub":"{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::password}}"
            },
            "VpcSecurityGroupIds":[
               {
                  "Fn::GetAtt":[
                     "TestVPC",
                     "DefaultSecurityGroup"
                  ]
               }
            ]
         }
      },
      "DocDBInstance":{
         "Type":"AWS::DocDB::DBInstance",
         "Properties":{
            "DBClusterIdentifier":{
               "Ref":"MyDocDBCluster"
            },
            "DBInstanceClass":"db.r5.large"
         }
      },
```

```
    "MyDBSubnetGroup":{
       "Type":"AWS::DocDB::DBSubnetGroup",
       "Properties":{
          "DBSubnetGroupDescription":"",
          "SubnetIds":[
             {
                "Ref":"TestSubnet01"
             },
             {
                "Ref":"TestSubnet02"
             }
          ]
       }
    },
    "SecretDocDBClusterAttachment":{
       "Type":"AWS::SecretsManager::SecretTargetAttachment",
       "Properties":{
          "SecretId":{
             "Ref":"MyDocDBClusterRotationSecret"
          },
          "TargetId":{
             "Ref":"MyDocDBCluster"
          },
          "TargetType":"AWS::DocDB::DBCluster"
       }
    },
    "MySecretRotationSchedule":{
       "Type":"AWS::SecretsManager::RotationSchedule",
       "DependsOn":"SecretDocDBClusterAttachment",
       "Properties":{
          "SecretId":{
             "Ref":"MyDocDBClusterRotationSecret"
          },
          "HostedRotationLambda":{
             "RotationType":"MongoDBSingleUser",
             "RotationLambdaName":"MongoDBSingleUser",
             "VpcSecurityGroupIds":{
                "Fn::GetAtt":[
                   "TestVPC",
                   "DefaultSecurityGroup"
                ]
             },
             "VpcSubnetIds":{
                "Fn::Join":[
```

```
                    ",",
                    [
                       {
                          "Ref":"TestSubnet01"
                       },
                       {
                          "Ref":"TestSubnet02"
                       }
                    ]
                 ]
              }
           },
           "RotationRules":{
             "Duration": "2h",
             "ScheduleExpression": "cron(0 8 1 * ? *)"
           }
        }
     }
  }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::SecretsManager-2020-07-23
Resources:
  TestVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
  TestSubnet01:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.96.0/19
      AvailabilityZone:
        Fn::Select:
        - '0'
        - Fn::GetAZs:
            Ref: AWS::Region
      VpcId:
        Ref: TestVPC
```

```
  TestSubnet02:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.128.0/19
      AvailabilityZone:
        Fn::Select:
        - '1'
        - Fn::GetAZs:
            Ref: AWS::Region
      VpcId:
        Ref: TestVPC
  SecretsManagerVPCEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      SubnetIds:
      - Ref: TestSubnet01
      - Ref: TestSubnet02
      SecurityGroupIds:
      - Fn::GetAtt:
        - TestVPC
        - DefaultSecurityGroup
      VpcEndpointType: Interface
      ServiceName:
        Fn::Sub: com.amazonaws.${AWS::Region}.secretsmanager
      PrivateDnsEnabled: true
      VpcId:
        Ref: TestVPC
  MyDocDBClusterRotationSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      GenerateSecretString:
        SecretStringTemplate: '{\"username\": \"someadmin\",\"ssl\": true}'
        GenerateStringKey: password
        PasswordLength: 16
        ExcludeCharacters: "\"@/\\"
      Tags:
      - Key: AppName
        Value: MyApp
  MyDocDBCluster:
    Type: AWS::DocDB::DBCluster
    Properties:
      DBSubnetGroupName:
        Ref: MyDBSubnetGroup
      MasterUsername:
```

```
        Fn::Sub: "{{resolve:secretsmanager:${MyDocDBClusterRotationSecret}::username}}"
      MasterUserPassword:
        Fn::Sub: "{{resolve:secretsmanager:${MyDocDBClusterRotationSecret}::password}}"
      VpcSecurityGroupIds:
      - Fn::GetAtt:
        - TestVPC
        - DefaultSecurityGroup
  DocDBInstance:
    Type: AWS::DocDB::DBInstance
    Properties:
      DBClusterIdentifier:
        Ref: MyDocDBCluster
      DBInstanceClass: db.r5.large
  MyDBSubnetGroup:
    Type: AWS::DocDB::DBSubnetGroup
    Properties:
      DBSubnetGroupDescription: ''
      SubnetIds:
      - Ref: TestSubnet01
      - Ref: TestSubnet02
  SecretDocDBClusterAttachment:
    Type: AWS::SecretsManager::SecretTargetAttachment
    Properties:
      SecretId:
        Ref: MyDocDBClusterRotationSecret
      TargetId:
        Ref: MyDocDBCluster
      TargetType: AWS::DocDB::DBCluster
  MySecretRotationSchedule:
    Type: AWS::SecretsManager::RotationSchedule
    DependsOn: SecretDocDBClusterAttachment
    Properties:
      SecretId:
        Ref: MyDocDBClusterRotationSecret
      HostedRotationLambda:
        RotationType: MongoDBSingleUser
        RotationLambdaName: MongoDBSingleUser
        VpcSecurityGroupIds:
          Fn::GetAtt:
          - TestVPC
          - DefaultSecurityGroup
        VpcSubnetIds:
          Fn::Join:
          - ","
```

```
        - - Ref: TestSubnet01
          - Ref: TestSubnet02
    RotationRules:
      Duration: 2h
      ScheduleExpression: 'cron(0 8 1 * ? *)'
```

Bagaimana Secrets Manager menggunakan AWS CloudFormation

Saat Anda menggunakan konsol untuk mengaktifkan rotasi, Secrets Manager menggunakan AWS CloudFormation untuk membuat sumber daya untuk rotasi. Jika Anda membuat fungsi rotasi baru selama proses itu, AWS CloudFormation buat `AWS::Serverless::Function`berdasarkan yang sesuaiTemplat fungsi rotasi. Kemudian AWS CloudFormation atur `RotationSchedule`, yang mengatur fungsi rotasi dan aturan rotasi untuk rahasia. Anda dapat melihat AWS CloudFormation tumpukan dengan memilih View stack di banner setelah Anda mengaktifkan rotasi otomatis.

Untuk informasi tentang mengaktifkan rotasi otomatis, lihatPutar rahasia.

Buat AWS Secrets Manager rahasia di AWS Cloud Development Kit (AWS CDK)

Untuk membuat, mengelola, dan mengambil rahasia di aplikasi CDK, Anda dapat menggunakan AWS Secrets ManagerConstruct Library, yang berisi,, `ResourcePolicy`, `RotationScheduleSecretSecretRotation`, dan konstruksi. `SecretTargetAttachment`

Sebagai contoh, lihat:

- Buat rahasia
- Impor rahasia
- Ambil rahasia
- Berikan izin untuk menggunakan rahasia
- Putar rahasia
- Putar rahasia database
- Replikasi rahasia ke Wilayah lain

Untuk informasi selengkapnya tentang CDK, lihat Panduan Pengembang AWS Cloud Development Kit (AWS CDK) v2.

Memantau AWS Secrets Manager rahasia

AWSmenyediakan alat pemantauan untuk menonton rahasia Secrets Manager, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu. Anda dapat menggunakan log jika Anda perlu menyelidiki penggunaan atau perubahan yang tidak terduga, dan kemudian Anda dapat memutar kembali perubahan yang tidak diinginkan. Anda juga dapat mengatur pemeriksaan otomatis untuk penggunaan rahasia yang tidak pantas dan segala upaya untuk menghapus rahasia.

Topik

Log AWS Secrets Manager peristiwa dengan AWS CloudTrail

AWS CloudTrailmerekam semua panggilan API untuk Secrets Manager sebagai peristiwa, termasuk panggilan dari konsol Secrets Manager, serta beberapa peristiwa lain untuk rotasi dan penghapusan versi rahasia. Untuk daftar catatan Secrets Manager entri log, lihatCloudTrail entri.

Anda dapat menggunakan CloudTrail konsol untuk melihat 90 hari terakhir dari peristiwa yang direkam. Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk peristiwa untuk Secrets Manager, buat jejak sehingga CloudTrail mengirimkan file log ke bucket Amazon S3. Lihat Membuat jejak untuk AWS akun Anda. Anda juga dapat mengonfigurasi CloudTrail untuk menerima file CloudTrail log dari beberapa Akun AWS dan Wilayah AWS.

Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data yang dikumpulkan dalam CloudTrail log. Lihat integrasi AWS layanan dengan CloudTrail log. Anda juga bisa mendapatkan notifikasi saat CloudTrail menerbitkan file log baru ke bucket Amazon S3 Anda. Lihat Mengonfigurasi notifikasi Amazon SNS untuk. CloudTrail

Untuk mengambil peristiwa Secrets Manager dari CloudTrail log (konsol)

1. Buka CloudTrail konsol di https://console.aws.amazon.com/cloudtrail/.

2. Pastikan konsol menunjuk ke wilayah tempat kejadian Anda terjadi. Konsol hanya menampilkan peristiwa yang terjadi di wilayah yang dipilih. Pilih wilayah dari daftar drop-down di sudut kanan atas konsol.
3. Di panel navigasi sebelah kiri, pilih Riwayat acara.
4. Pilih kriteria Filter dan/atau rentang waktu untuk membantu Anda menemukan acara yang Anda cari. Misalnya, untuk melihat semua peristiwa Secrets Manager, untuk Pilih atribut, pilih Sumber acara. Kemudian, untuk Masukkan sumber acara, pilih**`secretsmanager.amazonaws.com`**.
5. Untuk melihat detail tambahan, pilih panah perluas di sebelah acara. Untuk melihat semua informasi yang tersedia, pilih Lihat acara.

AWS CLI

Example Ambil peristiwa Secrets Manager dari log CloudTrail

`lookup-events`Contoh berikut mencari peristiwa Secrets Manager.

```
aws cloudtrail lookup-events \
    --region us-east-1 \
    --lookup-attributes
 AttributeKey=EventSource,AttributeValue=secretsmanager.amazonaws.com
```

AWS CloudTrailentri untuk Secrets Manager

AWS Secrets Managermenulis entri ke AWS CloudTrail log Anda untuk semua operasi Secrets Manager dan untuk acara lain yang terkait dengan rotasi dan penghapusan. Untuk informasi tentang mengambil tindakan pada peristiwa ini, lihatAcara Match Secrets Manager dengan EventBridge.

Jenis entri log

- Entri log untuk operasi Secrets Manager
- Entri log untuk penghapusan
- Entri log untuk replikasi
- Entri log untuk rotasi

Entri log untuk operasi Secrets Manager

Peristiwa yang dihasilkan oleh panggilan ke operasi Secrets Manager memiliki`"detail-type": ["AWS API Call via CloudTrail"]`.

> **Note**
>
> Sebelum Februari 2024, beberapa operasi Secrets Manager melaporkan peristiwa yang berisi “aRn” bukan “arn” untuk ARN rahasia. Untuk informasi lebih lanjut, lihat AWSre:Post.

Berikut ini adalah CloudTrail entri yang dihasilkan saat Anda atau layanan memanggil Secrets Manager beroperasi melalui API, SDK, atau CLI.

BatchGetSecretValue

Dihasilkan oleh BatchGetSecretValueoperasi. Untuk informasi tentang mengambil rahasia, lihatAmbil rahasia.

CancelRotateSecret

Dihasilkan oleh CancelRotateSecretoperasi. Untuk informasi tentang rotasi, lihatPutar rahasia.

CreateSecret

Dihasilkan oleh CreateSecretoperasi. Untuk informasi tentang membuat rahasia, lihatBuat dan kelola rahasia.

DeleteResourcePolicy

Dihasilkan oleh DeleteResourcePolicyoperasi. Untuk informasi tentang izin, lihatKontrol autentikasi dan akses.

DeleteSecret

Dihasilkan oleh DeleteSecretoperasi. Untuk informasi tentang menghapus rahasia, lihatthe section called “Hapus rahasia”.

DescribeSecret

Dihasilkan oleh DescribeSecretoperasi.

GetRandomPassword

Dihasilkan oleh GetRandomPasswordoperasi.

GetResourcePolicy

Dihasilkan oleh GetResourcePolicyoperasi. Untuk informasi tentang izin, lihatKontrol autentikasi dan akses.

GetSecretValue

Dihasilkan oleh GetSecretValuedan BatchGetSecretValueoperasi. Untuk informasi tentang mengambil rahasia, lihatAmbil rahasia.

ListSecrets

Dihasilkan oleh ListSecretsoperasi. Untuk informasi tentang daftar rahasia, lihatthe section called “Temukan rahasia”.

ListSecretVersionIds

Dihasilkan oleh ListSecretVersionIdsoperasi.

PutResourcePolicy

Dihasilkan oleh PutResourcePolicyoperasi. Untuk informasi tentang izin, lihatKontrol autentikasi dan akses.

PutSecretValue

Dihasilkan oleh PutSecretValueoperasi. Untuk informasi tentang memperbarui rahasia, lihatthe section called “Merubah rahasia”.

RemoveRegionsFromReplication

Dihasilkan oleh RemoveRegionsFromReplicationoperasi. Untuk informasi tentang mereplikasi rahasia, lihatthe section called “Replikasi rahasia ke Wilayah lain”.

ReplicateSecretToRegions

Dihasilkan oleh ReplicateSecretToRegionsoperasi. Untuk informasi tentang mereplikasi rahasia, lihatthe section called “Replikasi rahasia ke Wilayah lain”.

RestoreSecret

Dihasilkan oleh RestoreSecretoperasi. Untuk informasi tentang memulihkan rahasia yang dihapus, lihatthe section called “Kembalikan rahasia”.

RotateSecret

Dihasilkan oleh RotateSecretoperasi. Untuk informasi tentang rotasi, lihatPutar rahasia.

StopReplicationToReplica

Dihasilkan oleh StopReplicationToReplicaoperasi. Untuk informasi tentang mereplikasi rahasia, lihatthe section called “Replikasi rahasia ke Wilayah lain”.

TagResource

Dihasilkan oleh TagResourceoperasi. Untuk informasi tentang menandai rahasia, lihatthe section called “Rahasia tag”.

UntagResource

Dihasilkan oleh UntagResourceoperasi. Untuk informasi tentang membuka tanda rahasia, lihat. the section called “Rahasia tag”

UpdateSecret

Dihasilkan oleh UpdateSecretoperasi. Untuk informasi tentang memperbarui rahasia, lihatthe section called “Merubah rahasia”.

UpdateSecretVersionStage

Dihasilkan oleh UpdateSecretVersionStageoperasi. Untuk informasi tentang tahapan versi, lihatthe section called “Versi”.

ValidateResourcePolicy

Dihasilkan oleh ValidateResourcePolicyoperasi. Untuk informasi tentang izin, lihatKontrol autentikasi dan akses.

Entri log untuk penghapusan

Selain acara untuk operasi Secrets Manager, Secrets Manager menghasilkan peristiwa berikut yang terkait dengan penghapusan. Peristiwa ini memiliki`"detail-type": ["AWS Service Event via CloudTrail"]`.

CancelSecretVersionDelete

Dihasilkan oleh layanan Secrets Manager. Jika Anda memanggil `DeleteSecret` rahasia yang memiliki versi, dan kemudian meneleponRestoreSecret, Secrets Manager mencatat peristiwa ini untuk setiap versi rahasia yang dipulihkan. Untuk informasi tentang memulihkan rahasia yang dihapus, lihatthe section called “Kembalikan rahasia”.

EndSecretVersionDelete

Dihasilkan oleh layanan Secrets Manager ketika versi rahasia dihapus. Untuk informasi selengkapnya, lihat the section called “Hapus rahasia”.

StartSecretVersionDelete

Dihasilkan oleh layanan Secrets Manager saat Secrets Manager memulai penghapusan untuk versi rahasia. Untuk informasi tentang menghapus rahasia, lihatthe section called “Hapus rahasia”.

SecretVersionDeletion

Dihasilkan oleh layanan Secrets Manager saat Secrets Manager menghapus versi rahasia yang tidak digunakan lagi. Untuk informasi selengkapnya, lihat Versi rahasia.

Entri log untuk replikasi

Selain acara untuk operasi Secrets Manager, Secrets Manager menghasilkan peristiwa berikut yang terkait dengan replikasi. Peristiwa ini memiliki`"detail-type": ["AWS Service Event via CloudTrail"]`.

ReplicationFailed

Dihasilkan oleh layanan Secrets Manager saat replikasi gagal. Untuk informasi tentang mereplikasi rahasia, lihatthe section called “Replikasi rahasia ke Wilayah lain”.

ReplicationStarted

Dihasilkan oleh layanan Secrets Manager saat Secrets Manager mulai mereplikasi rahasia. Untuk informasi tentang mereplikasi rahasia, lihatthe section called “Replikasi rahasia ke Wilayah lain”.

ReplicationSucceeded

Dihasilkan oleh layanan Secrets Manager ketika sebuah rahasia berhasil direplikasi. Untuk informasi tentang mereplikasi rahasia, lihatthe section called “Replikasi rahasia ke Wilayah lain”.

Entri log untuk rotasi

Selain acara untuk operasi Secrets Manager, Secrets Manager menghasilkan peristiwa berikut yang terkait dengan rotasi. Peristiwa ini memiliki`"detail-type": ["AWS Service Event via CloudTrail"]`.

RotationStarted

Dihasilkan oleh layanan Secrets Manager saat Secrets Manager mulai memutar rahasia. Untuk informasi tentang rotasi, lihatPutar rahasia.

RotationAbandoned

Dihasilkan oleh layanan Secrets Manager ketika Secrets Manager meninggalkan upaya rotasi dan menghapus `AWSPENDING` label dari versi rahasia yang ada. Secrets Manager meninggalkan rotasi saat Anda membuat versi baru rahasia selama rotasi. Untuk informasi tentang rotasi, lihatPutar rahasia.

RotationFailed

Dihasilkan oleh layanan Secrets Manager saat rotasi gagal. Untuk informasi tentang rotasi, lihatthe section called “Memecahkan masalah rotasi”.

RotationSucceeded

Dihasilkan oleh layanan Secrets Manager ketika sebuah rahasia berhasil diputar. Untuk informasi tentang rotasi, lihatPutar rahasia.

TestRotationStarted

Dihasilkan oleh layanan Secrets Manager saat Secrets Manager mulai menguji rotasi untuk rahasia yang tidak dijadwalkan untuk rotasi langsung. Untuk informasi tentang rotasi, lihatPutar rahasia.

TestRotationSucceeded

Dihasilkan oleh layanan Secrets Manager ketika Secrets Manager berhasil menguji rotasi untuk rahasia yang tidak dijadwalkan untuk rotasi langsung. Untuk informasi tentang rotasi, lihatPutar rahasia.

TestRotationFailed

Dihasilkan oleh layanan Secrets Manager ketika Secrets Manager menguji rotasi untuk rahasia yang tidak dijadwalkan untuk rotasi langsung dan rotasi gagal. Untuk informasi tentang rotasi, lihatthe section called “Memecahkan masalah rotasi”.

Cocokkan AWS Secrets Manager acara dengan Amazon EventBridge

Di Amazon EventBridge, Anda dapat mencocokkan peristiwa Secrets Manager dari entri CloudTrail log. Anda dapat mengonfigurasi EventBridge aturan yang mencari peristiwa ini dan kemudian mengirim peristiwa baru yang dihasilkan ke target untuk mengambil tindakan. Untuk daftar CloudTrail

entri yang dicatat oleh Secrets Manager, lihatCloudTrail entri. Untuk petunjuk penyiapan EventBridge, lihat Memulai EventBridge di Panduan EventBridge Pengguna.

Cocokkan semua perubahan dengan rahasia tertentu

Contoh berikut menunjukkan pola EventBridge peristiwa yang cocok dengan entri log untuk perubahan rahasia.

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

Cocokkan acara saat nilai rahasia berputar

Contoh berikut menunjukkan pola EventBridge peristiwa yang cocok dengan entri CloudTrail log untuk perubahan nilai rahasia yang terjadi dari pembaruan manual atau rotasi otomatis. Karena beberapa peristiwa ini berasal dari operasi Secrets Manager dan beberapa dihasilkan oleh layanan Secrets Manager, Anda harus menyertakan `detail-type` untuk keduanya.

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

Monitor AWS Secrets Manager dengan Amazon CloudWatch

Anda dapat memantau AWS Secrets Manager menggunakan Amazon CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

Untuk Secrets Manager, Anda dapat menggunakannya CloudWatch untuk memberi tahu Anda saat tingkat permintaan untuk API atau jumlah rahasia di akun Anda mencapai ambang tertentu. Anda juga dapat menggunakan CloudWatch untuk memantau perkiraan biaya Secrets Manager. Untuk informasi selengkapnya, lihat Membuat alarm penagihan untuk memantau perkiraan AWS tagihan Anda.

Topik

- Metrik dan dimensi Secrets Manager
- Membuat alarm untuk memantau metrik Secrets Manager
- Burung kenari Amazon CloudWatch Synthetics

Metrik dan dimensi Secrets Manager

Namespace `AWS/SecretsManager` mencakup metrik berikut.

Metrik	Deskripsi
`ResourceCount`	Jumlah rahasia di akun Anda, termasuk rahasia yang ditandai untuk dihapus. Metrik diterbitkan setiap jam. Unit: Hitungan

Dimensi untuk metrik Secrets Manager.

Dimensi	Deskripsi
`Service`	Nama dari layanan AWS yang berisi sumber daya. Untuk Secrets Manager, nilai untuk dimensi ini adalah`Secrets Manager`.
`Type`	Tipe entitas yang dilaporkan. Untuk Secrets Manager, nilai untuk dimensi ini adalah`Resource`.
`Resource`	Tipe sumber daya yang sedang berjalan. Untuk Secrets Manager, nilai untuk dimensi ini adalah`SecretCount` .
`Class`	Tidak ada.

Permintaan Secrets Manager API yang dapat Anda pantau menggunakan CloudWatch metrik termasuk`GetSecretValue`,`DescribeSecret`,`ListSecrets`, dan lainnya. Untuk menemukan metrik, di CloudWatch konsol, pilih Semua metrik, lalu di kotak pencarian, masukkan istilah pencarian Anda, misalnya. **`secrets`**

Membuat alarm untuk memantau metrik Secrets Manager

Anda dapat membuat CloudWatch alarm yang mengirimkan pesan Amazon SNS saat nilai metrik berubah dan menyebabkan alarm berubah status. Alarm mengawasi metrik selama periode waktu yang Anda tentukan, dan melakukan tindakan berdasarkan nilai metrik relatif terhadap ambang batas tertentu selama beberapa periode waktu. Alarm memanggil tindakan untuk perubahan status berkelanjutan saja. CloudWatch alarm tidak memanggil tindakan hanya karena mereka berada dalam keadaan tertentu; negara harus telah berubah dan dipertahankan untuk sejumlah periode tertentu.

Untuk informasi selengkapnya, lihat Menggunakan CloudWatch alarm Amazon dan Membuat CloudWatch alarm berdasarkan deteksi anomali.

Burung kenari Amazon CloudWatch Synthetics

Canary Amazon CloudWatch Synthetics adalah skrip yang dapat dikonfigurasi yang berjalan sesuai jadwal untuk memantau titik akhir dan API Anda. Canary mengikuti rute yang sama dan melakukan tindakan yang sama sebagai pelanggan, yang memungkinkan bagi Anda untuk terus memverifikasi pengalaman pelanggan bahkan ketika Anda tidak memiliki lalu lintas pelanggan pada aplikasi Anda.

Untuk contoh cara mengintegrasikan Secrets Manager, lihat Mengintegrasikan kenari Anda dengan layanan lain AWS.

Memantau AWS Secrets Manager rahasia yang dijadwalkan untuk dihapus dengan menggunakan Amazon CloudWatch

Anda dapat menggunakan kombinasiAWS CloudTrail, Amazon CloudWatch Logs, dan Amazon Simple Notification Service (Amazon SNS) untuk membuat alarm yang memberi tahu Anda tentang upaya apa pun untuk mengakses penghapusan rahasia yang tertunda. Jika Anda menerima pemberitahuan dari alarm, Anda mungkin ingin membatalkan penghapusan rahasia untuk memberi diri Anda lebih banyak waktu untuk menentukan apakah Anda benar-benar ingin menghapusnya. Investigasi Anda mungkin mengakibatkan rahasia dipulihkan karena Anda masih membutuhkan rahasianya. Atau, Anda mungkin perlu memperbarui pengguna dengan rincian rahasia baru untuk digunakan.

Prosedur berikut menjelaskan cara menerima pemberitahuan ketika permintaan untuk `GetSecretValue` operasi yang menghasilkan pesan kesalahan tertentu yang ditulis ke file CloudTrail log Anda. Operasi API lainnya dapat dilakukan secara rahasia tanpa memicu alarm. CloudWatch Alarm ini mendeteksi penggunaan yang mungkin menunjukkan seseorang atau aplikasi menggunakan kredensi yang sudah ketinggalan zaman.

Sebelum memulai prosedur ini, Anda harus mengaktifkan CloudTrail akun Wilayah AWS dan tempat Anda ingin memantau permintaan AWS Secrets Manager API. Untuk instruksi, buka Membuat jejak untuk pertama kalinya di Panduan AWS CloudTrail Pengguna.

Langkah 1: Konfigurasikan pengiriman file CloudTrail log ke CloudWatch log

Anda harus mengonfigurasi pengiriman file CloudTrail log Anda ke CloudWatch Log. Anda melakukan ini agar CloudWatch Log dapat memonitornya untuk permintaan Secrets Manager API untuk mengambil penghapusan rahasia yang tertunda.

Untuk mengonfigurasi pengiriman file CloudTrail log ke CloudWatch Log

1. Buka CloudTrail konsol di https://console.aws.amazon.com/cloudtrail/.
2. Di bilah navigasi atas, pilih AWS Wilayah untuk memantau rahasia.
3. Di panel navigasi kiri, pilih Jalur, lalu pilih nama jejak yang akan dikonfigurasi. CloudWatch
4. Pada halaman Konfigurasi Jalur, gulir ke bawah ke bagian CloudWatch Log, lalu pilih ikon edit (✏).

5. Untuk grup log baru atau yang sudah ada, ketikkan nama untuk grup log, seperti**`CloudTrail/MyCloudWatchLogGroup`**.
6. Untuk peran IAM, Anda dapat menggunakan peran default bernama CloudTrail_ CloudWatchLogs _Role. Peran ini memiliki kebijakan peran default dengan izin yang diperlukan untuk mengirimkan CloudTrail peristiwa ke grup log.
7. Pilih Lanjutkan untuk menyimpan konfigurasi Anda.
8. Pada saat AWS CloudTrailakan mengirimkan CloudTrail peristiwa yang terkait dengan aktivitas API di akun Anda ke halaman grup CloudWatch log Log Anda, pilih Izinkan.

Langkah 2: Buat CloudWatch alarm

Untuk menerima pemberitahuan saat operasi Secrets Manager `GetSecretValue` API meminta untuk mengakses penghapusan rahasia yang tertunda, Anda harus membuat CloudWatch alarm dan mengonfigurasi notifikasi.

Untuk membuat CloudWatch alarm

1. Masuk ke CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di bilah navigasi atas, pilih AWS Wilayah tempat Anda ingin memantau rahasia.
3. Di panel navigasi bagian kiri, pilih Log.
4. Dalam daftar Grup Log, pilih kotak centang di samping grup log yang Anda buat dalam prosedur sebelumnya, seperti CloudTrail/MyCloudWatchLogGroup. Kemudian pilih Buat Filter Metrik.
5. Untuk Pola Filter, ketik atau tempel yang berikut ini:

   ```
   { $.eventName = "GetSecretValue" && $.errorMessage = "*secret because it was marked
    for deletion*" }
   ```

   Pilih Tetapkan Metrik.
6. Pada halaman Buat Metrik Filter dan Tetapkan Metrik, lakukan hal berikut:

   a. Untuk Namespace Metrik, ketik **`CloudTrailLogMetrics`**.

   b. Untuk Nama Metrik, ketik **`AttemptsToAccessDeletedSecrets`**.

   c. Pilih Tampilkan pengaturan metrik lanjutan, lalu jika perlu untuk Nilai Metrik, ketik**`1`**.

   d. Pilih Buat Filter.
7. Dalam kotak filter, pilih Buat Alarm.

8. Di jendela Buat Alarm, lakukan hal berikut:

   a. Untuk Nama, ketik **`AttemptsToAccessDeletedSecretsAlarm`**.

   b. Kapanpun:, for is:, pilih **>=**, lalu ketik**1**.

   c. Di samping Kirim pemberitahuan ke:, lakukan salah satu hal berikut:

      - Untuk membuat dan menggunakan topik Amazon SNS baru, pilih Daftar baru, lalu ketik nama topik baru. Untuk Daftar email:, ketik setidaknya satu alamat email. Anda dapat mengetik beberapa alamat email dengan memisahkannya dengan koma.
      - Untuk menggunakan topik Amazon SNS yang sudah ada, pilih nama topik yang akan digunakan. Jika daftar tidak ada, pilih Pilih daftar.

   d. Pilih Buat Alarm.

Langkah 3: Uji CloudWatch alarm

Untuk menguji alarm Anda, buat rahasia dan kemudian jadwalkan untuk dihapus. Kemudian, cobalah untuk mengambil nilai rahasia. Anda segera menerima email di alamat yang Anda konfigurasikan di alarm. Ini mengingatkan Anda untuk penggunaan rahasia yang dijadwalkan untuk dihapus.

Validasi kepatuhan untuk AWS Secrets Manager

Tanggung jawab kepatuhan Anda saat menggunakan Secrets Manager ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWSmenyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah untuk deployment lingkungan dasar yang fokus pada keamanan dan kepatuhan di AWS.
- Merancang Laporan Resmi Keamanan dan Kepatuhan HIPAA – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang patuh-HIPAA.
- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Configmenilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan. Untuk informasi selengkapnya, lihat the section called “Rahasia audit untuk kepatuhan”.
- AWS Security Hubmemberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk informasi tentang penggunaan Security Hub guna mengevaluasi sumber daya Secrets Manager, lihat AWS Secrets Managerkontrol di Panduan AWS Security Hub Pengguna.
- IAM Access Analyzer menganalisis kebijakan, termasuk pernyataan kondisi dalam kebijakan, yang memungkinkan entitas eksternal mengakses rahasia. Untuk informasi selengkapnya, lihat Mempratinjau akses dengan Access Analyzer.
- AWS Systems Managermenyediakan runbook standar untuk Secrets Manager. Untuk informasi selengkapnya, lihat referensi runbook Automation Systems Manager untuk Secrets Manager.

AWS Secrets Managertelah menjalani audit untuk standar berikut dan dapat menjadi bagian dari solusi Anda ketika Anda perlu mendapatkan sertifikasi kepatuhan.

AWStelah memperluas program kepatuhan Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan (HIPAA) untuk dimasukkan AWS Secrets Manager sebagai layanan yang memenuhi syarat HIPAA. Jika Anda memiliki Business Associate Agreement (BAA) yang dieksekusiAWS, Anda dapat menggunakan Secrets Manager

	untuk membantu membangun aplikasi yang sesuai dengan HIPAA Anda. AWSmenawarkan whitepaper yang berfokus pada HIPAA untuk pelanggan yang tertarik untuk mempelajari lebih lanjut tentang bagaimana mereka dapat memanfaatkan pemrosesan dan AWS penyimpanan informasi kesehatan. Untuk informasi lebih lanjut, lihat Kepatuhan HIPAA.
	AWS Secrets Managermemiliki Pengesahan Kepatuhan untuk Standar Keamanan Data (DSS) Industri Kartu Pembayaran (PCI) versi 3.2 di Penyedia Layanan Level 1. Pelanggan yang menggunakan AWS produk dan layanan untuk menyimpan, memproses, atau mengirimkan data pemegang kartu dapat menggunakannya AWS Secrets Manager saat mereka mengelola sertifikasi kepatuhan PCI DSS mereka sendiri. Untuk informasi lebih lanjut tentang PCI DSS, termasuk cara meminta salinan AWS PCI Compliance Package, lihat PCI DSS Level 1.
	AWS Secrets Managertelah berhasil menyelesaikan sertifikasi kepatuhan untuk ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018, dan ISO 9001. Untuk informasi selengkapnya, lihat ISO 27001, ISO 27017, ISO 27018, ISO 9001.
	Laporan System and Organization Control (SOC) adalah laporan pemeriksaan pihak ketiga independen yang menunjukkan bagaimana Secrets Manager mencapai kontrol dan tujuan kepatuhan utama. Tujuan dari laporan ini adalah untuk membantu Anda dan auditor Anda memahami AWS kontrol yang ditetapkan untuk mendukung operasi dan kepatuhan. Untuk informasi selengkapnya, lihat Kepatuhan SOC.
	Federal Risk and Authorization Management Program (FedRAMP) adalah program pemerintah yang menyediakan pendekatan standar untuk penilaian keamanan, otorisasi, dan pemantauan berkelanjutan untuk produk dan layanan cloud. Program FedRAMP juga menyediakan otorisasi sementara untuk layanan dan wilayah untuk Timur/Barat GovCloud dan untuk mengkonsumsi data pemerintah atau yang diatur. Untuk informasi lain, lihat Kepatuhan FedRAMP .
	The Department of Defense (DoD) Cloud Computing Security Requirements Guide (SRG) menyediakan penilaian standar dan proses otorisasi untuk penyedia layanan cloud (CSP) untuk mendapatkan otorisasi sementara DoD, sehingga mereka dapat melayani pelanggan DoD. Untuk informasi selengkapnya, lihat DoD SRG Resources

	Program Penilai Terdaftar Keamanan Informasi (IRAP) memungkinkan pelanggan pemerintah Australia untuk memvalidasi bahwa kontrol yang sesuai telah ada dan menentukan model tanggung jawab yang sesuai untuk memenuhi persyaratan Manual Keamanan Informasi (ISM) pemerintah Australia yang diproduksi oleh Australian Cyber Security Centre (ACSC). Untuk informasi lebih lanjut, lihat Sumber Daya IRAP
	Amazon Web Services (AWS) mencapai pengesahan Laporan Audit Penyedia Layanan Outsourced (OSPAR). AWSKeselarasan dengan Pedoman Asosiasi Bank di Singapura (ABS) tentang Tujuan Pengendalian dan Prosedur untuk Penyedia Layanan Outsourced (Pedoman ABS) menunjukkan AWS komitmen kepada pelanggan untuk memenuhi harapan tinggi bagi penyedia layanan cloud yang ditetapkan oleh industri jasa keuangan di Singapura. Untuk informasi selengkapnya, lihat Sumber Daya OSPAR

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact.

AWS Secrets ManagerRahasia audit untuk kepatuhan dengan menggunakan AWS Config

Anda dapat menggunakannya AWS Config untuk mengevaluasi rahasia Anda dan menilai seberapa baik mereka mematuhi praktik internal, pedoman industri, dan peraturan Anda. Anda menentukan persyaratan keamanan dan kepatuhan internal Anda untuk rahasia menggunakan AWS Config aturan. Kemudian AWS Config dapat mengidentifikasi rahasia yang tidak sesuai dengan aturan Anda. Anda juga dapat melacak perubahan metadata rahasia, konfigurasi rotasi, kunci KMS yang digunakan untuk enkripsi rahasia, fungsi rotasi Lambda, dan tag yang terkait dengan rahasia.

Anda dapat menerima pemberitahuan dari Amazon SNS tentang konfigurasi rahasia Anda. Misalnya, Anda dapat menerima notifikasi Amazon SNS untuk daftar rahasia yang tidak dikonfigurasi untuk rotasi yang memungkinkan Anda mendorong praktik terbaik keamanan untuk memutar rahasia.

Jika Anda memiliki rahasia di beberapa Akun AWS dan Wilayah AWS di organisasi Anda, Anda dapat menggabungkan konfigurasi dan data kepatuhan tersebut.

Untuk menambahkan aturan baru untuk rahasia Anda

- Ikuti petunjuk tentang Bekerja dengan aturan AWS Config terkelola, dan pilih salah satu aturan berikut:
  - `secretsmanager-rotation-enabled-check`— Memeriksa apakah rotasi dikonfigurasi untuk rahasia yang disimpan di Secrets Manager.
  - `secretsmanager-scheduled-rotation-success-check`— Memeriksa apakah rotasi sukses terakhir berada dalam frekuensi rotasi yang dikonfigurasi. Frekuensi minimum untuk cek adalah setiap hari.
  - `secretsmanager-secret-periodic-rotation`— Memeriksa apakah rahasia diputar dalam jumlah hari yang ditentukan.
  - `secretsmanager-secret-unused`— Memeriksa apakah rahasia diakses dalam jumlah hari yang ditentukan.
  - `secretsmanager-using-cmk`— Memeriksa apakah rahasia dienkripsi menggunakan Kunci yang dikelola AWS `aws/secretsmanager` atau kunci yang dikelola pelanggan yang Anda buat. AWS KMS

Setelah Anda menyimpan aturan, AWS Config evaluasi rahasia Anda setiap kali metadata rahasia berubah. Anda dapat mengonfigurasi AWS Config untuk memberi tahu Anda tentang perubahan. Untuk informasi selengkapnya, lihat Pemberitahuan yang AWS Config mengirim ke topik Amazon SNS.

Agregat rahasia dari Anda Akun AWS dan Wilayah AWS

Anda dapat mengonfigurasi Agregator Data Multi-Wilayah AWS Config Multi-Akun untuk meninjau konfigurasi rahasia Anda di semua akun dan wilayah di organisasi Anda, lalu meninjau konfigurasi rahasia Anda dan membandingkannya dengan praktik terbaik manajemen rahasia.

Anda harus mengaktifkan AWS Config dan aturan AWS Config terkelola khusus untuk rahasia di semua akun dan wilayah sebelum Anda membuat agregator. Untuk informasi selengkapnya, lihat Gunakan CloudFormation StackSets untuk menyediakan sumber daya di beberapa Akun AWS dan Wilayah.

Untuk informasi selengkapnya tentang AWS Config Aggregator, lihat Agregasi Data Multi-Wilayah Multi-Akun dan Menyiapkan Agregator Menggunakan Konsol di Panduan Pengembang. AWS Config

Keamanan di AWS Secrets Manager

Keamanan di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Anda dan AWS berbagi tanggung jawab untuk keamanan. Model tanggung jawab bersama menggambarkan ini sebagai keamanan cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWSjuga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi keefektifan keamanan kami sebagai bagian dari program kepatuhan AWS. Untuk mempelajari program kepatuhan yang berlaku di AWS Secrets Manager, lihat Cakupan Layanan Menurut Program Kepatuhan AWS.
- Keamanan di cloud — AWS Layanan Anda menentukan tanggung jawab Anda. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Untuk sumber daya lainnya, lihat Pilar Keamanan - Kerangka AWS Well-Architected.

Topik

Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager

Saat Anda menggunakan AWS Command Line Interface (AWS CLI) untuk menjalankan AWS operasi, Anda memasukkan perintah tersebut di shell perintah. Misalnya, Anda dapat menggunakan prompt perintah Windows atau Windows PowerShell, atau shell Bash atau Z, antara lain. Banyak dari

shell perintah ini mencakup fungsionalitas yang dirancang untuk meningkatkan produktivitas. Tetapi fungsi ini dapat digunakan untuk mengkompromikan rahasia Anda. Misalnya, di sebagian besar shell, Anda dapat menggunakan tombol panah atas untuk melihat perintah yang terakhir dimasukkan. Fitur riwayat perintah dapat dimanfaatkan oleh siapa saja yang mengakses sesi tanpa jaminan Anda. Selain itu, utilitas lain yang bekerja di latar belakang mungkin memiliki akses ke parameter perintah Anda, dengan tujuan yang dimaksudkan untuk membantu Anda melakukan tugas dengan lebih efisien. Untuk mengurangi risiko tersebut, pastikan Anda mengambil langkah-langkah berikut:

- Selalu kunci komputer Anda ketika Anda berjalan menjauh dari konsol Anda.
- Copot pemasangan atau nonaktifkan utilitas konsol yang tidak perlu atau tidak lagi digunakan.
- Pastikan shell atau program akses jarak jauh, jika Anda menggunakan salah satu atau yang lain, jangan mencatat perintah yang diketik.
- Gunakan teknik untuk meneruskan parameter yang tidak ditangkap oleh riwayat perintah shell. Contoh berikut menunjukkan bagaimana Anda dapat mengetik teks rahasia ke dalam file teks, dan kemudian meneruskan file ke AWS Secrets Manager perintah dan segera menghancurkan file. Ini berarti riwayat shell yang khas tidak menangkap teks rahasia.

  Contoh berikut menunjukkan perintah Linux yang khas tetapi shell Anda mungkin memerlukan perintah yang sedikit berbeda:

```
$ 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.
```

Setelah Anda menjalankan perintah ini, Anda harus dapat menggunakan panah atas dan bawah untuk menggulir melalui riwayat perintah dan melihat bahwa teks rahasia tidak ditampilkan pada baris apa pun.

> **Important**
>
> Secara default, Anda tidak dapat melakukan teknik yang setara di Windows kecuali Anda terlebih dahulu mengurangi ukuran buffer riwayat perintah menjadi 1.

Untuk mengkonfigurasi Windows Command Prompt untuk hanya memiliki 1 buffer riwayat perintah dari 1 perintah

1. Buka prompt perintah Administrator (Jalankan sebagai administrator).
2. Pilih ikon di kiri atas, lalu pilih Properties.
3. Pada tab Options, atur Buffer Size dan Number of Buffer keduanya ke**1**, lalu pilih OK.
4. Setiap kali Anda harus mengetik perintah yang tidak Anda inginkan dalam riwayat, segera ikuti dengan satu perintah lain, seperti:

   ```
   echo.
   ```

   Ini memastikan Anda menyiram perintah sensitif.

Untuk shell Windows Command Prompt, Anda dapat mengunduh alat SysInternalsSDelete, dan kemudian menggunakan perintah yang mirip dengan yang berikut ini:

```
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.
```

Perlindungan data di AWS Secrets Manager

Model tanggung jawab bersama AWS diterapkan untuk perlindungan data AWS Secrets Manager. Sebagaimana dijelaskan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Konten ini meliputi konfigurasi keamanan dan tugas-tugas pengelolaan untuk berbagai layanan Layanan AWS yang Anda gunakan. Untuk informasi lebih lanjut tentang privasi data, lihat FAQ tentang Privasi Data. Untuk informasi tentang perlindungan data di Eropa, lihat postingan blog Model Tanggung Jawab Bersama AWS dan GDPR di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya Anda melindungi kredensial Akun AWS dan menyiapkan akun pengguna individu dengan AWS Identity and Access Management (IAM). Dengan cara seperti itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut ini:

- Gunakan otentikasi multi-faktor (MFA) dengan setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Secrets Manager mendukung TLS 1.2 dan 1.3 di semua Wilayah. Secrets Manager juga mendukung opsi pertukaran kunci pasca-kuantum hibrida untuk protokol enkripsi jaringan TLS (PQTLS).
- Tanda tangani permintaan terprogram Anda ke Secrets Manager dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda dapat menggunakan AWS Security Token Service(AWS STS) untuk menghasilkan kredensyal keamanan sementara untuk menandatangani permintaan.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail. Lihat the section called “Log dengan AWS CloudTrail ”.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat the section called “Titik akhir Secrets Manager”.
- Jika Anda menggunakan AWS CLI untuk mengakses Secrets Manager,the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

Enkripsi saat tidak aktif

Secrets Manager menggunakan enkripsi via AWS Key Management Service (AWS KMS) untuk melindungi kerahasiaan data saat istirahat. AWS KMSmenyediakan penyimpanan kunci dan layanan enkripsi yang digunakan oleh banyak AWS layanan. Setiap rahasia di Secrets Manager dienkripsi dengan kunci data yang unik. Setiap kunci data dilindungi oleh kunci KMS. Anda dapat memilih untuk menggunakan enkripsi default dengan Secrets Manager Kunci yang dikelola AWS untuk akun, atau Anda dapat membuat kunci terkelola pelanggan Anda sendiriAWS KMS. Menggunakan kunci yang dikelola pelanggan memberi Anda kontrol otorisasi yang lebih terperinci atas aktivitas utama KMS Anda. Untuk informasi selengkapnya, lihat the section called “Enkripsi rahasia dan dekripsi”.

Enkripsi dalam transit

Secrets Manager menyediakan endpoint yang aman dan pribadi untuk mengenkripsi data dalam perjalanan. Endpoint yang aman dan pribadi memungkinkan AWS untuk melindungi integritas permintaan API ke Secrets Manager. AWSmemerlukan panggilan API ditandatangani oleh pemanggil menggunakan sertifikat X.509 dan/atau Secrets Manager Secret Access Key. Persyaratan ini dinyatakan dalam Proses Penandatanganan Versi Tanda Tangan 4 (Sigv4).

Jika Anda menggunakan AWS Command Line Interface (AWS CLI) atau salah satu AWS SDK untuk melakukan panggilanAWS, Anda mengonfigurasi kunci akses yang akan digunakan. Kemudian alat-alat tersebut secara otomatis menggunakan tombol akses untuk menandatangani permintaan untuk Anda. Lihat the section called “Mengurangi risiko menggunakan AWS CLI untuk menyimpan rahasia Anda AWS Secrets Manager”.

Privasi lalu lintas antar jaringan

AWSmenawarkan opsi untuk menjaga privasi saat merutekan lalu lintas melalui rute jaringan yang dikenal dan pribadi.

Lalu lintas antara layanan dan aplikasi dan klien lokal

Anda memiliki dua opsi konektivitas antara jaringan privat dan AWS Secrets Manager:

- Koneksi Site-to-Site VPN AWS. Untuk informasi selengkapnya, lihat Apa itu AWS VPN Site-to-Site?
- Koneksi AWS Direct Connect. Untuk informasi selengkapnya, lihat Apa itu AWS Direct Connect?

Lalu lintas antara sumber daya AWS di Wilayah yang sama

Jika Anda ingin mengamankan lalu lintas antara Secrets Manager dan klien APIAWS, siapkan AWS PrivateLinkuntuk mengakses titik akhir Secrets Manager API secara pribadi.

Pengelolaan kunci enkripsi

Ketika Secrets Manager perlu mengenkripsi versi baru dari data rahasia yang dilindungi, Secrets Manager mengirimkan permintaan AWS KMS untuk menghasilkan kunci data baru dari kunci KMS. Secrets Manager menggunakan kunci data ini untuk enkripsi amplop. Secrets Manager menyimpan kunci data terenkripsi dengan rahasia terenkripsi. Ketika rahasia perlu didekripsi, Secrets Manager meminta AWS KMS untuk mendekripsi kunci data. Secrets Manager kemudian menggunakan kunci data yang didekripsi untuk mendekripsi rahasia terenkripsi. Secrets Manager tidak pernah menyimpan kunci data dalam bentuk yang tidak terenkripsi dan menghapus kunci dari memori sesegera mungkin. Untuk informasi selengkapnya, lihat the section called “Enkripsi rahasia dan dekripsi”.

Enkripsi rahasia dan dekripsi di AWS Secrets Manager

Secrets Manager menggunakan enkripsi amplop dengan AWS KMS kunci dan kunci data untuk melindungi setiap nilai rahasia. Setiap kali nilai rahasia dalam rahasia berubah, Secrets Manager meminta kunci data baru AWS KMS untuk melindunginya. Kunci data dienkripsi di bawah kunci KMS dan disimpan dalam metadata rahasia. Untuk mendekripsi rahasia, Secrets Manager terlebih dahulu mendekripsi kunci data terenkripsi menggunakan kunci KMS. AWS KMS

Secrets Manager tidak menggunakan kunci KMS untuk mengenkripsi nilai rahasia secara langsung. Sebaliknya, ia menggunakan kunci KMS untuk menghasilkan dan mengenkripsi kunci data simetris Advanced Encryption Standard (AES) 256-bit, dan menggunakan kunci data untuk mengenkripsi nilai rahasia. Secrets Manager menggunakan kunci data plaintext untuk mengenkripsi nilai rahasia di luar AWS KMS, dan kemudian menghapusnya dari memori. Ini menyimpan salinan terenkripsi dari kunci data dalam metadata dari rahasia.

Saat Anda membuat rahasia, Anda dapat memilih kunci yang dikelola pelanggan enkripsi simetris di Akun AWS dan Wilayah, atau Anda dapat menggunakan Kunci yang dikelola AWS for Secrets Manager (`aws/secretsmanager`). Jika Anda memilih Kunci yang dikelola AWS `aws/secretsmanager` dan itu belum ada, Secrets Manager membuatnya dan mengaitkannya dengan rahasia. Anda dapat menggunakan kunci KMS yang sama atau kunci KMS yang berbeda untuk setiap rahasia di akun Anda. Anda mungkin ingin menggunakan kunci KMS yang berbeda untuk

mengatur izin khusus pada kunci untuk sekelompok rahasia, atau jika Anda ingin mengaudit operasi tertentu untuk kunci tersebut. Secrets Manager hanya mendukung kunci KMS enkripsi simetris. Jika Anda menggunakan kunci KMS di toko kunci eksternal, operasi kriptografi pada kunci KMS mungkin memakan waktu lebih lama dan kurang dapat diandalkan dan tahan lama karena permintaan harus melakukan perjalanan di luar. AWS

Untuk informasi tentang mengubah kunci enkripsi untuk rahasia, lihatthe section called “Ubah kunci enkripsi untuk rahasia”.

Saat Anda mengubah kunci enkripsi, Secrets Manager mengenkripsi ulang `AWSCURRENTAWSPENDING`, dan `AWSPREVIOUS` versi dengan kunci baru. Untuk menghindari mengunci Anda dari rahasia, Secrets Manager menyimpan semua versi yang ada dienkripsi dengan kunci sebelumnya. Itu berarti Anda dapat mendekripsi`AWSCURRENT`,`AWSPENDING`, dan `AWSPREVIOUS` versi dengan kunci sebelumnya atau kunci baru.

Untuk membuatnya sehingga hanya `AWSCURRENT` dapat didekripsi oleh kunci enkripsi baru, buat versi baru rahasia dengan kunci baru. Kemudian untuk dapat mendekripsi versi `AWSCURRENT` rahasia, Anda harus memiliki izin untuk kunci baru.

Untuk menemukan kunci KMS yang terkait dengan rahasia, lihat rahasia di konsol atau panggil ListSecretsatau DescribeSecret. Ketika rahasia dikaitkan dengan Kunci yang dikelola AWS for Secrets Manager (`aws/secretsmanager`), operasi ini tidak mengembalikan pengenal kunci KMS.

Topik

- Apa yang dienkripsi?
- Proses enkripsi dan dekripsi
- Izin untuk kunci KMS
- Bagaimana Secrets Manager menggunakan kunci KMS Anda
- Kebijakan utama dari Kunci yang dikelola AWS (aws/secretsmanager)
- Konteks enkripsi Secrets Manager
- Memantau interaksi Secrets Manager dengan AWS KMS

Apa yang dienkripsi?

Secrets Manager mengenkripsi nilai rahasia, tetapi tidak mengenkripsi yang berikut:

- Nama dan deskripsi rahasia

- Pengaturan rotasi
- ARN dari kunci KMS yang terkait dengan rahasia
- Setiap AWS tag terlampir

Proses enkripsi dan dekripsi

Untuk mengenkripsi nilai rahasia dalam rahasia, Secrets Manager menggunakan proses berikut.

1. Secrets Manager memanggil AWS KMS GenerateDataKeyoperasi dengan ID kunci KMS untuk rahasia dan permintaan untuk kunci simetris AES 256-bit. AWS KMS mengembalikan kunci data plaintext dan salinan kunci data yang dienkripsi di bawah kunci KMS.
2. Secrets Manager menggunakan kunci data plaintext dan algoritma Advanced Encryption Standard (AES) untuk mengenkripsi nilai rahasia di luar. AWS KMS Ini akan menghapus kunci plaintext dari memori sesegera mungkin setelah menggunakannya.
3. Secrets Manager menyimpan kunci data terenkripsi dalam metadata rahasia sehingga ini tersedia untuk mendekripsi nilai rahasia. Namun, tidak ada API Secrets Manager yang mengembalikan rahasia terenkripsi atau kunci data terenkripsi.

Untuk mendekripsi nilai rahasia terenkripsi:

1. Secrets Manager memanggil operasi AWS KMS Dekripsi dan meneruskan kunci data terenkripsi.
2. AWS KMS menggunakan kunci KMS untuk rahasia untuk mendekripsi kunci data. Ini mengembalikan kunci data plaintext.
3. Secrets Manager menggunakan kunci data plaintext untuk mendekripsi nilai rahasia. Kemudian, ini menghapus kunci data dari memori sesegera mungkin.

Izin untuk kunci KMS

Ketika Secrets Manager menggunakan kunci KMS dalam operasi kriptografi, ia bertindak atas nama pengguna yang mengakses atau memperbarui nilai rahasia. Anda dapat memberikan izin dalam kebijakan IAM atau kebijakan utama. Operasi Secrets Manager berikut memerlukan AWS KMS izin.

- CreateSecret
- GetSecretValue
- PutSecretValue

- UpdateSecret
- ReplicateSecretToRegions

Untuk mengizinkan kunci KMS hanya digunakan untuk permintaan yang berasal dari Secrets Manager, dalam kebijakan izin, Anda dapat menggunakan kunci ViaService kondisi kms: dengan nilainya. `secretsmanager.<Region>.amazonaws.com`

Anda juga dapat menggunakan kunci atau nilai dalam konteks enkripsi sebagai syarat untuk menggunakan kunci KMS untuk operasi kriptografi. Misalnya, Anda dapat menggunakan operator ketentuan string di IAM atau dokumen kebijakan kunci, atau menggunakan batasan hibah dalam hibah. Perbanyakan hibah kunci KMS dapat memakan waktu hingga lima menit. Untuk informasi lebih lanjut, lihat CreateGrant.

Bagaimana Secrets Manager menggunakan kunci KMS Anda

Secrets Manager memanggil AWS KMS operasi berikut dengan kunci KMS Anda.

GenerateDataKey

Secrets Manager memanggil AWS KMS GenerateDataKeyoperasi sebagai tanggapan atas operasi Secrets Manager berikut.

- CreateSecret— Jika rahasia baru menyertakan nilai rahasia, Secrets Manager meminta kunci data baru untuk mengenkripsi itu.
- PutSecretValue— Secrets Manager meminta kunci data baru untuk mengenkripsi nilai rahasia yang ditentukan.
- ReplicateSecretToRegions— Untuk mengenkripsi rahasia yang direplikasi, Secrets Manager meminta kunci data untuk kunci KMS di Region replika.
- UpdateSecret— Jika Anda mengubah nilai rahasia atau kunci KMS, Secrets Manager meminta kunci data baru untuk mengenkripsi nilai rahasia baru.

RotateSecretOperasi tidak memanggil`GenerateDataKey`, karena tidak mengubah nilai rahasia. Namun, jika `RotateSecret` memanggil fungsi rotasi Lambda yang mengubah nilai rahasia, panggilannya ke `PutSecretValue` operasi memicu `GenerateDataKey` permintaan.

Dekripsi

Secrets Manager memanggil operasi Dekripsi sebagai respons untuk operasi Secrets Manager berikut.

- GetSecretValuedan BatchGetSecretValue— Secrets Manager mendekripsi nilai rahasia sebelum mengembalikannya ke penelepon. Untuk mendekripsi nilai rahasia terenkripsi, Secrets Manager memanggil operasi Dekripsi untuk AWS KMS mendekripsi kunci data terenkripsi dalam rahasia. Kemudian, ini menggunakan kunci data plaintext untuk mendekripsi nilai rahasia terenkripsi. Untuk perintah batch, Secrets Manager dapat menggunakan kembali kunci yang didekripsi, sehingga tidak semua panggilan menghasilkan permintaan. `Decrypt`
- PutSecretValuedan UpdateSecret— Sebagian besar `PutSecretValue` dan `UpdateSecret` permintaan tidak memicu `Decrypt` operasi. Namun, ketika permintaan `PutSecretValue` atau `UpdateSecret` berusaha untuk mengubah nilai rahasia dalam versi rahasia yang ada, Secrets Manager mendekripsi nilai rahasia yang ada dan membandingkannya dengan nilai rahasia dalam permintaan untuk mengonfirmasi bahwa mereka adalah sama. Tindakan ini memastikan bahwa operasi Secrets Manager adalah idempoten. Untuk mendekripsi nilai rahasia terenkripsi, Secrets Manager memanggil operasi Dekripsi untuk AWS KMS mendekripsi kunci data terenkripsi dalam rahasia. Kemudian, ini menggunakan kunci data plaintext untuk mendekripsi nilai rahasia terenkripsi.
- ReplicateSecretToRegions— Secrets Manager pertama kali mendekripsi nilai rahasia di Wilayah utama sebelum mengenkripsi ulang nilai rahasia dengan kunci KMS di Region replika.

Enkripsi

Secrets Manager memanggil operasi Enkripsi sebagai respons terhadap operasi Secrets Manager berikut:

- UpdateSecret— Jika Anda mengubah kunci KMS, Secrets Manager mengenkripsi ulang kunci data yang melindungi`AWSCURRENT`,`AWSPREVIOUS`, dan versi `AWSPENDING` rahasia dengan kunci baru.
- ReplicateSecretToRegions— Secrets Manager mengenkripsi ulang kunci data selama replikasi menggunakan kunci KMS di Region replika.

DescribeKey

Secrets Manager memanggil DescribeKeyoperasi untuk menentukan apakah akan mencantumkan kunci KMS saat Anda membuat atau mengedit rahasia di konsol Secrets Manager.

Memvalidasi akses ke kunci KMS

Ketika Anda membuat atau mengubah kunci KMS yang terkait dengan rahasia, Secrets Manager memanggil `GenerateDataKey` dan `Decrypt` operasi dengan kunci KMS yang ditentukan. Panggilan ini mengonfirmasi bahwa penelepon memiliki izin untuk menggunakan kunci KMS

untuk operasi ini. Secrets Manager membuang hasil operasi tersebut; itu tidak menggunakannya dalam operasi kriptografi.

Anda dapat mengidentifikasi panggilan validasi ini karena nilai dari kunci `SecretVersionId` konteks enkripsi dalam permintaan ini adalah `RequestToValidateKeyAccess`.

> **Note**
>
> Di masa lalu, panggilan validasi Secrets Manager tidak termasuk konteks enkripsi. Anda mungkin menemukan panggilan tanpa konteks enkripsi di AWS CloudTrail log lama.

Kebijakan utama dari Kunci yang dikelola AWS (**`aws/secretsmanager`**)

Kebijakan kunci Kunci yang dikelola AWS untuk Secrets Manager (`aws/secretsmanager`) memberi pengguna izin untuk menggunakan kunci KMS untuk operasi tertentu hanya jika Secrets Manager membuat permintaan atas nama pengguna. Kebijakan kunci tidak mengizinkan pengguna untuk menggunakan kunci KMS secara langsung.

Kebijakan utama ini, seperti kebijakan semua Kunci yang dikelola AWS, ditetapkan oleh layanan. Anda tidak dapat mengubah kebijakan kunci, tetapi Anda dapat melihatnya kapan saja. Untuk detailnya, lihat Melihat kebijakan utama.

Pernyataan kebijakan dalam kebijakan kunci memiliki efek sebagai berikut:

- Izinkan pengguna di akun untuk menggunakan kunci KMS untuk operasi kriptografi hanya ketika permintaan berasal dari Secrets Manager atas nama mereka. Kunci kondisi `kms:ViaService` memberlakukan pembatasan ini.
- Memungkinkan AWS akun untuk membuat kebijakan IAM yang memungkinkan pengguna untuk melihat properti kunci KMS dan mencabut hibah.
- Meskipun Secrets Manager tidak menggunakan hibah untuk mendapatkan akses ke kunci KMS, kebijakan ini juga memungkinkan Secrets Manager untuk membuat hibah untuk kunci KMS atas nama pengguna dan memungkinkan akun untuk mencabut hibah yang memungkinkan Secrets Manager untuk menggunakan kunci KMS. Ini adalah elemen standar dokumen kebijakan untuk sebuah Kunci yang dikelola AWS.

Berikut ini adalah kebijakan utama untuk Kunci yang dikelola AWS contoh 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": "*"
    }
  ]
}
```

Konteks enkripsi Secrets Manager

Konteks enkripsi adalah seperangkat pasangan nilai kunci yang berisi data non-rahasia yang berubah-ubah. Ketika Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, secara AWS KMS kriptografis mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda harus meneruskan konteks enkripsi yang sama.

Dalam permintaannya GenerateDataKeydan Dekripsi ke AWS KMS, Secrets Manager menggunakan konteks enkripsi dengan dua pasangan nama-nilai yang mengidentifikasi rahasia dan versinya, seperti yang ditunjukkan pada contoh berikut. Nama-nama tidak bervariasi, tetapi nilai-nilai konteks enkripsi gabungan akan berbeda untuk setiap nilai rahasia.

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
a1b2c3",
    "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

Anda dapat menggunakan konteks enkripsi untuk mengidentifikasi operasi kriptografi ini dalam catatan audit dan log, seperti AWS CloudTraildan Amazon CloudWatch Logs, dan sebagai syarat untuk otorisasi dalam kebijakan dan hibah.

Enkripsi konteks Secrets Manager terdiri dari dua pasangan nama-nilai.

- SecretARN — Pasangan nama-nilai pertama mengidentifikasi rahasia. Kuncinya adalah `SecretARN`. Nilai tersebut adalah Amazon Resource Name (ARN) dari rahasia.

  ```
  "SecretARN": "ARN of an Secrets Manager secret"
  ```

  Sebagai contoh, jika ARN dari rahasia adalah `arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3`, maka konteks enkripsi akan mencakup pasangan berikut.

  ```
  "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3"
  ```

- SecretVersionId— Pasangan nama-nilai kedua mengidentifikasi versi rahasia. Kuncinya adalah `SecretVersionId`. Nilai adalah ID versi.

  ```
  "SecretVersionId": "<version-id>"
  ```

  Sebagai contoh, jika ID versi dari rahasia adalah `EXAMPLE1-90ab-cdef-fedc-ba987SECRET1`, maka konteks enkripsi akan mencakup pasangan berikut.

  ```
  "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
  ```

Saat Anda membuat atau mengubah kunci KMS untuk rahasia, Secrets Manager mengirim GenerateDataKeydan Mendekripsi permintaan AWS KMS untuk memvalidasi bahwa pemanggil memiliki izin untuk menggunakan kunci KMS untuk operasi ini. Ini membuang tanggapan; tidak menggunakannya pada nilai rahasia.

Dalam permintaan validasi ini, nilai dari `SecretARN` adalah ARN sebenarnya dari rahasia, tetapi nilai `SecretVersionId` adalah `RequestToValidateKeyAccess`, seperti yang ditunjukkan dalam konteks enkripsi contoh berikut. Nilai khusus ini membantu Anda untuk mengidentifikasi permintaan validasi di log dan jejak audit.

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
a1b2c3",
    "SecretVersionId": "RequestToValidateKeyAccess"
}
```

> **Note**
>
> Di masa lalu, permintaan validasi Secrets Manager tidak termasuk konteks enkripsi. Anda mungkin menemukan panggilan tanpa konteks enkripsi di AWS CloudTrail log lama.

Memantau interaksi Secrets Manager dengan AWS KMS

Anda dapat menggunakan AWS CloudTrail dan Amazon CloudWatch Logs untuk melacak permintaan yang dikirimkan Secrets Manager atas nama Anda. AWS KMS Untuk informasi tentang pemantauan penggunaan rahasia, lihatMemantau rahasia.

GenerateDataKey

Saat Anda membuat atau mengubah nilai rahasia dalam rahasia, Secrets Manager mengirimkan GenerateDataKeypermintaan AWS KMS yang menentukan kunci KMS untuk rahasia tersebut.

Peristiwa yang mencatat operasi `GenerateDataKey` serupa dengan peristiwa contoh berikut. Permintaan dipanggil oleh `secretsmanager.amazonaws.com`. Parameter termasuk Nama Sumber Daya Amazon (ARN) dari kunci KMS untuk rahasia, penentu kunci yang memerlukan kunci 256-bit, dan konteks enkripsi yang mengidentifikasi rahasia dan versi.

```
{
    "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"
}
```


Dekripsi

Ketika Anda mendapatkan atau mengubah nilai rahasia rahasia, Secrets Manager mengirimkan permintaan Dekripsi AWS KMS untuk mendekripsi kunci data terenkripsi. Untuk perintah batch, Secrets Manager dapat menggunakan kembali kunci yang didekripsi, sehingga tidak semua panggilan menghasilkan permintaan. `Decrypt`

Peristiwa yang mencatat operasi `Decrypt` serupa dengan peristiwa contoh berikut. Pengguna adalah kepala sekolah di AWS akun Anda yang mengakses tabel. Parameter termasuk kunci tabel terenkripsi (sebagai gumpalan ciphertext) dan konteks enkripsi yang mengidentifikasi tabel dan akun. AWS AWS KMS memperoleh ID kunci KMS dari ciphertext.


```
{
    "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"
}
```

Enkripsi

Ketika Anda mengubah kunci KMS yang terkait dengan rahasia, Secrets Manager mengirimkan permintaan Enkripsi AWS KMS untuk mengenkripsi ulang`AWSCURRENT`,`AWSPREVIOUS`, dan versi `AWSPENDING` rahasia dengan kunci baru. Saat Anda mereplikasi rahasia ke Wilayah lain, Secrets Manager juga mengirimkan permintaan Enkripsi ke. AWS KMS

Peristiwa yang mencatat operasi `Encrypt` serupa dengan peristiwa contoh berikut. Pengguna adalah kepala sekolah di AWS akun Anda yang mengakses tabel.

```
{
    "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"
}
```

Keamanan infrastruktur dalam AWS Secrets Manager

Sebagai layanan terkelola, AWS Secrets Manager dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWSDiarsiteksikan dengan Baik Pilar Keamanan.

Akses ke Secrets Manager melalui jaringan adalah melalui API yang AWS dipublikasikan menggunakan TLS. Secrets Manager API dapat dipanggil dari lokasi jaringan mana pun. Namun, Secrets Manager mendukung kebijakan akses berbasis sumber daya, yang dapat mencakup pembatasan berdasarkan alamat IP sumber. Anda juga dapat menggunakan kebijakan sumber daya Secrets Manager untuk mengontrol akses ke rahasia dari titik akhir virtual private cloud (VPC) tertentu, atau VPC tertentu. Secara efektif, ini mengisolasi akses jaringan ke rahasia tertentu hanya dari VPC spesifik dalam AWS jaringan. Untuk informasi selengkapnya, lihat Titik akhir VPC.

Ketahanan di AWS Secrets Manager

AWSmembangun infrastruktur global di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Availability Zones memungkinkan Anda menjadi lebih tersedia, toleran terhadap kesalahan, dan skalabel daripada infrastruktur pusat data tunggal atau ganda tradisional.

Untuk informasi lebih lanjut tentang ketahanan dan pemulihan bencana, lihat Reliability Pillar - AWS Well-Architected Framework.

Untuk informasi selengkapnya tentang Wilayah AWS dan Zona Ketersediaan, lihat Infrastruktur Global AWS.

TLS pasca-kuantum

Secrets Manager mendukung opsi pertukaran kunci pasca-kuantum hibrida untuk protokol enkripsi jaringan Transport Layer Security (TLS). Anda dapat menggunakan opsi TLS ini saat terhubung ke titik akhir API Secrets Manager. Kami menawarkan fitur ini sebelum algoritma pasca-kuantum distandarisasi sehingga Anda dapat mulai menguji efek protokol pertukaran kunci ini pada panggilan Secrets Manager. Fitur pertukaran kunci pasca-kuantum hibrida opsional ini setidaknya seaman enkripsi TLS yang kami gunakan saat ini dan kemungkinan akan memberikan manfaat keamanan tambahan. Namun, fitur-fitur tersebut memengaruhi latensi dan throughput dibandingkan dengan protokol pertukaran kunci klasik yang digunakan saat ini.

Guna melindungi data yang dienkripsi saat ini terhadap potensi serangan masa depan, AWS berpartisipasi dengan komunitas kriptografi dalam pengembangan algoritme tahan-kuantum atau pasca-kuantum. Kami telah menerapkan suite cipher pertukaran kunci pasca-kuantum hibrida di titik akhir Secrets Manager. Cipher suite hibrida ini, yang menggabungkan elemen klasik dan pasca-kuantum, memastikan bahwa koneksi TLS Anda setidaknya sekuat itu dengan cipher suite klasik. Namun, karena karakteristik kinerja dan persyaratan bandwidth suite cipher hybrid berbeda dari mekanisme pertukaran kunci klasik, kami sarankan Anda mengujinya pada panggilan API Anda.

Secrets Manager mendukung PQTLS di semua Wilayah kecuali Wilayah China.

Untuk mengkonfigurasi TLS pasca-kuantum hibrida

1. Tambahkan klien Common Runtime AWS untuk dependensi Maven Anda. Sebaiknya gunakan versi terbaru yang tersedia. Misalnya, pernyataan ini menambahkan versi 2.20.0.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>aws-crt-client</artifactId>
  <version>2.20.0</version>
</dependency>
```

2. Tambahkan AWS SDK for Java 2.x ke project Anda dan inisialisasi. Aktifkan suite sandi pasca-kuantum hibrida pada klien HTTP Anda.

```
SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
            .postQuantumTlsEnabled(true)
            .build();
```

3. Buat klien asinkron Secrets Manager.

```
SecretsManagerAsyncClient SecretsManagerAsync = SecretsManagerAsyncClient.builder()
            .httpClient(awsCrtHttpClient)
            .build();
```

   Sekarang ketika Anda memanggil operasi Secrets Manager API, panggilan Anda ditransmisikan ke titik akhir Secrets Manager menggunakan TLS pasca-kuantum hibrida.

Untuk informasi lebih lanjut tentang penggunaan TLS pasca-kuantum hibrida, lihat:

- AWS SDK for Java 2.xPanduan Pengembang dan posting blog yang AWS SDK for Java 2.xdirilis.
- Memperkenalkans2n-tls, Implementasi dan Penggunaan s2n-tls TLS Open Source Baru.
- Kriptografi Pasca-Kuantum di Institut Nasional untuk Standar dan Teknologi (NIST).
- Metode Enkapsulasi Kunci Pasca-Quantum Hybrid (PQ KEM) untuk Transport Layer Security 1.2 (TLS).

TLS pasca-kuantum untuk Secrets Manager tersedia di semua Wilayah AWS kecuali China.

Pemecahan Masalah AWS Secrets Manager

Gunakan informasi di sini untuk membantu Anda mendiagnosis dan memperbaiki masalah yang mungkin Anda temui saat bekerja dengan Secrets Manager.

Untuk masalah yang terkait dengan rotasi, lihatthe section called “Memecahkan masalah rotasi”.

Topik

Pesan “Akses ditolak” saat mengirim permintaan ke Secrets Manager

Verifikasi bahwa Anda memiliki izin untuk memanggil operasi dan sumber daya yang Anda minta. Administrator harus memberikan izin dengan melampirkan kebijakan IAM ke pengguna IAM Anda, atau grup di mana Anda menjadi anggota. Jika pernyataan kebijakan yang memberikan izin tersebut menyertakan syarat apapun, seperti time-of-day batas alamat IP, maka Anda juga harus memenuhi persyaratan tersebut ketika Anda mengirim permintaan. Untuk informasi tentang melihat atau mengubah kebijakan untuk pengguna, grup, atau peran IAM, lihat Bekerja dengan Kebijakan dalam Panduan Pengguna IAM. Untuk informasi tentang izin yang diperlukan untuk Secrets Manager, lihatKontrol autentikasi dan akses.

Jika Anda menandatangani permintaan API secara manual, tanpa menggunakan AWSSDK, verifikasi bahwa Anda menandatangani permintaan dengan benar.

“Akses ditolak” untuk kredensi keamanan sementara

Verifikasi pengguna atau peran IAM yang Anda gunakan untuk membuat permintaan memiliki izin yang benar. Izin untuk kredensi keamanan sementara berasal dari pengguna atau peran IAM. Ini

berarti izin terbatas pada izin yang diberikan kepada pengguna atau peran IAM. Untuk informasi lebih lanjut tentang bagaiman izin kredensial keamanan sementara ditentukan, lihat Mengontrol Izin untuk Kredensial Keamanan Sementara dalam Panduan Pengguna IAM.

Verifikasi bahwa permintaan Anda ditandatangani dengan benar dan bahwa permintaan tersebut memiliki bentuk yang baik. Untuk detailnya, lihat dokumentasi toolkit untuk SDK yang Anda pilih, atau Menggunakan Kredensial Keamanan Sementara untuk Meminta Akses ke AWS Sumber Daya di Panduan Pengguna IAM.

Verifikasikan bahwa kredensial keamanan sementara Anda belum kedaluwarsa. Untuk informasi lebih lanjut, lihat Meminta Kredensial Keamanan Sementara dalam Panduan Pengguna IAM.

Untuk informasi tentang izin yang diperlukan untuk Secrets Manager, lihatKontrol autentikasi dan akses.

Perubahan yang saya buat tidak selalu langsung terlihat.

Secrets Manager menggunakan model komputasi terdistribusi yang disebut konsistensi akhirnya. Setiap perubahan yang Anda lakukan di Secrets Manager (atau AWS layanan lainnya) membutuhkan waktu agar terlihat dari semua titik akhir yang memungkinkan. Beberapa penundaan dihasilkan dari waktu yang diperlukan untuk mengirim data dari server ke server, dari zona replikasi ke zona replikasi, dan dari wilayah ke wilayah di seluruh dunia. Secrets Manager juga menggunakan caching untuk meningkatkan kinerja, tetapi dalam beberapa kasus ini dapat menambah waktu. Perubahan mungkin tidak terlihat sampai waktu data yang disimpan di-cache sebelumnya habis.

Rancang aplikasi global Anda untuk memperhitungkan potensi penundaan ini. Selain itu, pastikan aplikasi bekerja sesuai harapan, bahkan ketika perubahan yang dilakukan di satu lokasi tidak secara langsung terlihat di lokasi lain.

Untuk informasi selengkapnya tentang bagaimana beberapa AWS layanan lainnya dipengaruhi oleh konsistensi akhir, lihat:

- Mengelola konsistensi data di Panduan Developer Basis Data Amazon Redshift
- Model Konsistensi Data Amazon S3 di Panduan Pengguna Amazon Simple Storage Service
- Memastikan Konsistensi Saat Menggunakan Amazon S3 dan Amazon EMR untuk Alur Kerja ETL di Blog Big Data AWS
- Amazon EC2 Eventual Consistency di Referensi API Amazon EC2

“Tidak dapat menghasilkan kunci data dengan kunci KMS asimetris” saat membuat rahasia

Secrets Manager menggunakan kunci KMS simetris yang terkait dengan rahasia untuk menghasilkan kunci data untuk setiap nilai rahasia. Anda tidak dapat menggunakan tombol KMS asimetris. Verifikasi Anda menggunakan kunci KMS enkripsi simetris alih-alih kunci KMS asimetris. Untuk petunjuk, lihat Mengidentifikasi kunci KMS asimetris.

Operasi AWS CLI atau AWS SDK tidak dapat menemukan rahasia saya dari ARN sebagian

Dalam banyak kasus, Secrets Manager dapat menemukan rahasia Anda dari bagian ARN daripada ARN penuh. Namun, jika nama rahasia Anda berakhir dengan tanda hubung diikuti oleh enam karakter, Secrets Manager mungkin tidak dapat menemukan rahasia hanya dari sebagian ARN. Sebagai gantinya, kami sarankan Anda menggunakan ARN lengkap atau nama rahasianya.

Detail lebih lanjut

Secrets Manager mencakup enam karakter acak di akhir nama rahasia untuk membantu memastikan bahwa ARN rahasia itu unik. Jika rahasia asli dihapus, dan kemudian rahasia baru dibuat dengan nama yang sama, kedua rahasia memiliki ARN yang berbeda karena karakter ini. Pengguna dengan akses ke rahasia lama tidak secara otomatis mendapatkan akses ke rahasia baru karena ARN berbeda.

Secrets Manager membangun ARN untuk rahasia dengan Region, akun, nama rahasia, dan kemudian tanda hubung dan enam karakter lagi, sebagai berikut:

```
arn:aws:secretsmanager:us-east-2:111122223333:secret:SecretName-abcdef
```

Jika nama rahasia Anda diakhiri dengan tanda hubung dan enam karakter, hanya menggunakan sebagian dari ARN dapat muncul ke Secrets Manager seolah-olah Anda menentukan ARN lengkap. Misalnya, Anda mungkin memiliki rahasia bernama `MySecret-abcdef` ARN

```
arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret-abcdef-nutBrk
```

Jika Anda memanggil operasi berikut, yang hanya menggunakan bagian dari ARN rahasia, maka Secrets Manager mungkin tidak menemukan rahasianya.

```
$ aws secretsmanager describe-secret --secret-id arn:aws:secretsmanager:us-
east-2:111122223333:secret:MySecret-abcdef
```

Rahasia ini dikelola oleh AWS layanan, dan Anda harus menggunakan layanan itu untuk memperbaruinya.

Jika Anda menemukan pesan ini saat mencoba memodifikasi rahasia, rahasia hanya dapat diperbarui dengan menggunakan layanan pengelolaan yang tercantum dalam pesan. Untuk informasi selengkapnya, lihat Rahasia yang dikelola oleh layanan lain.

Untuk menentukan siapa yang mengelola rahasia, Anda dapat meninjau nama rahasia. Rahasia yang dikelola oleh layanan lain diawali dengan ID layanan tersebut. Atau, diAWS CLI, panggil deskripsikan-rahasia, dan kemudian tinjau bidangnya. `OwningService`

Kuota AWS Secrets Manager

Secrets Manager read API memiliki kuota TPS yang tinggi, dan API control plane yang jarang disebut memiliki kuota TPS yang lebih rendah. Kami menyarankan Anda menghindari menelepon `PutSecretValue` atau dengan `UpdateSecret` kecepatan berkelanjutan lebih dari sekali setiap 10 menit. Saat Anda `UpdateSecret` menelepon `PutSecretValue` atau memperbarui nilai rahasia, Secrets Manager membuat versi baru dari rahasia tersebut. Secrets Manager menghapus versi yang tidak berlabel ketika ada lebih dari 100, tetapi tidak menghapus versi yang dibuat kurang dari 24 jam yang lalu. Jika Anda memperbarui nilai rahasia lebih dari sekali setiap 10 menit, Anda membuat lebih banyak versi daripada yang dihapus Secrets Manager, dan Anda akan mencapai kuota untuk versi rahasia.

Anda dapat mengoperasikan beberapa wilayah di akun Anda, dan setiap kuota khusus untuk setiap wilayah.

Ketika aplikasi dalam satu Akun AWS menggunakan rahasia yang dimiliki oleh akun yang berbeda, itu dikenal sebagai permintaan lintas akun. Untuk permintaan lintas akun, Secrets Manager membatasi akun identitas yang membuat permintaan, bukan akun yang memiliki rahasia. Misalnya, jika identitas dari akun A menggunakan rahasia di akun B, penggunaan rahasia hanya berlaku untuk kuota di akun A.

Kuota Secrets Manager

Nama	Default	Dapat Disesuaikan	Deskripsi
Tingkat gabungan permintaan DeleteResourcePolicy GetResourcePolicy, PutResourcePolicy,, dan ValidateResourcePolicy API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk DeleteResourcePolicy, GetResourcePolicy, PutResourcePolicy, dan permintaan ValidateResourcePolicy API digabungkan.

Nama	Default	Dapat Disesuaikan	Deskripsi
Tingkat gabungan permintaan DescribeSecret dan GetSecretValue API	Setiap Wilayah yang didukung: 10.000 per detik	Tidak	Transaksi maksimum per detik untuk DescribeSecret dan permintaan GetSecretValue API digabungkan.
Tingkat gabungan permintaan PutSecretValue, RemoveRegionsFromReplication ReplicateSecretToRegion, StopReplicationToReplica, UpdateSecret,, dan UpdateSecretVersionStage API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk PutSecretValue, RemoveRegionsFromReplication,, ReplicateSecretToRegion, StopReplicationToReplica, UpdateSecret, dan permintaan UpdateSecretVersionStage API digabungkan.
Tingkat gabungan permintaan RestoreSecret API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk permintaan RestoreSecret API.
Tingkat gabungan permintaan RotateSecret dan CancelRotateSecret API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk RotateSecret dan permintaan CancelRotateSecret API digabungkan.
Tingkat gabungan permintaan TagResource dan UntagResource API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk TagResource dan permintaan UntagResource API digabungkan.

Nama	Default	Dapat Disesuaikan	Deskripsi
Tingkat permintaan BatchGetSecretValue API	Setiap Wilayah yang didukung: 100 per detik	Tidak	Transaksi maksimum per detik untuk permintaan BatchGetSecretValue API.
Tingkat permintaan CreateSecret API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk permintaan CreateSecret API.
Tingkat permintaan DeleteSecret API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk permintaan DeleteSecret API.
Tingkat permintaan GetRandomPassword API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk permintaan GetRandomPassword API.
Tingkat permintaan ListSecretVersionIds API	Setiap Wilayah yang didukung: 50 per detik	Tidak	Transaksi maksimum per detik untuk permintaan ListSecretVersionIds API.
Tingkat permintaan ListSecrets API	Setiap Wilayah yang didukung: 100 per detik	Tidak	Transaksi maksimum per detik untuk permintaan ListSecrets API.
Panjang kebijakan berbasis sumber daya	Setiap Wilayah yang didukung: 20.480	Tidak	Jumlah maksimum karakter dalam kebijakan izin berbasis sumber daya yang dilampirkan pada rahasia.

Nama	Default	Dapat Disesuaikan	Deskripsi
Ukuran nilai rahasia	Setiap Wilayah yang didukung: 65.536 Bytes	Tidak	Ukuran maksimum nilai rahasia terenkripsi. Jika nilai rahasia adalah string, maka ini adalah jumlah karakter yang diizinkan dalam nilai rahasia.
Rahasia	Setiap Wilayah yang didukung: 500.000	Tidak	Jumlah maksimum rahasia di setiap AWS Wilayah AWS akun ini.
Label pementasan yang dilampirkan di semua versi rahasia	Setiap Wilayah yang didukung: 20	Tidak	Jumlah maksimum label pementasan yang dilampirkan di semua versi rahasia.
Versi per rahasia	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum versi rahasia.

Tambahkan percobaan ulang ke aplikasi Anda

AWSKlien Anda mungkin melihat panggilan ke Secrets Manager gagal karena masalah tak terduga di sisi klien. Atau panggilan mungkin gagal karena pembatasan tarif dari Secrets Manager. Jika Anda melebihi kuota permintaan API, Secrets Manager membatasi permintaan tersebut. Ini menolak permintaan yang valid dan mengembalikan throttling kesalahan. Untuk kedua jenis kegagalan, kami sarankan Anda mencoba lagi panggilan setelah masa tunggu singkat. Ini disebut strategi backoff dan coba lagi.

Jika mengalami kesalahan berikut, Anda mungkin ingin menambahkan percobaan ulang ke kode aplikasi Anda:

Kesalahan sementara dan pengecualian

- `RequestTimeout`
- `RequestTimeoutException`
- `PriorRequestNotComplete`
- `ConnectionError`
- `HTTPClientError`

Pelambatan sisi layanan dan kesalahan dan pengecualian batas

- `Throttling`
- `ThrottlingException`
- `ThrottledException`
- `RequestThrottledException`
- `TooManyRequestsException`
- `ProvisionedThroughputExceededException`
- `TransactionInProgressException`
- `RequestLimitExceeded`
- `BandwidthLimitExceeded`
- `LimitExceededException`
- `RequestThrottled`
- `SlowDown`

Untuk informasi lebih lanjut, serta kode contoh, pada percobaan ulang, backoff eksponensial, dan jitter, lihat sumber daya berikut:

- Backoff dan Jitter Eksponensial
- Batas waktu, percobaan ulang, dan backoff dengan jitter
- Eror pengulangan dan mundur eksponensial dalam AWS.

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis terakhir AWS Secrets Manager. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Secrets Manager berubah menjadi kebijakan AWS terkelola	Kebijakan `SecretsManagerReadWrite` terkelola sekarang termasuk `redshift-serverless` izin. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola AWS Secrets Manager	Maret 12, 2024

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting dalam setiap rilis Panduan AWS Secrets Manager Pengguna sebelum Februari 2024.

Perubahan	Deskripsi	Tanggal
Ketersediaan umum	Ini adalah rilis publik perdana Secrets Manager.	Apr 4, 2018

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.