aws

Guida per l'utente

AWS Secrets Manager

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Secrets Manager: Guida per l'utente

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cos'è Secrets Manager? 1
Inizia a usare Secrets Manager 1
Conformità agli standard 2
Prezzi 2
AWS servizi che utilizzano AWS Secrets Manager segreti 3
Accesso Secrets Manager 7
Console Secrets Manager 7
Strumenti a riga di comando 7
AWS SDK 8
API query HTTPS 8
Endpoint di Secrets Manager 9
Concetti 14
Secret 14
Versione 15
Rotazione 16
Strategia di rotazione 17
Utente singolo 17
Utenti alternati 17
Tutorial 20
Revisore Amazon CodeGuru 20
Sostituzione dei segreti codificati 20
Fase 1: creazione del segreto 21
Fase 2: aggiornamento del codice 23
Fase 3: aggiornamento del segreto 24
Fasi successive 24
Sostituzione delle credenziali del database codificato 25
Fase 1: creazione del segreto 26
Fase 2: aggiornamento del codice 27
Fase 3: rotazione del segreto 28
Passaggi successivi 29
Rotazione a utenti alternati 29
Autorizzazioni 30
Prerequisiti 31
Fase 1: creazione di un utente del database Amazon RDS 34

Fase 2: creazione di un segreto per le credenziali dell'utente ... 37
Fase 3: eseguire il test del segreto ruotato ... 38
Fase 4: eliminazione delle risorse ... 39
Fasi successive ... 39
Rotazione a utente singolo ... 40
Autorizzazioni ... 40
Prerequisiti ... 41
Fase 1: creazione di un utente del database Amazon RDS ... 41
Fase 2: creazione di un segreto per le credenziali utente del database ... 42
Fase 3: esecuzione del test della password ruotata ... 43
Fase 4: eliminazione delle risorse ... 44
Fasi successive ... 44
Autenticazione e controllo degli accessi ... 45
Autorizzazioni di amministrazione di Secrets Manager ... 45
Autorizzazioni per accedere ai segreti ... 46
Autorizzazioni per le funzioni di rotazione Lambda ... 46
Autorizzazioni per le chiavi di crittografia ... 46
Allegare un policy di autorizzazione a un'identità ... 46
Allegare una policy di autorizzazione a un segreto ... 47
AWS CLI ... 48
AWS SDK ... 49
AWS politiche gestite ... 50
SecretsManagerReadWrite ... 50
Aggiornamenti alle policy ... 52
Determinazione di chi ha le autorizzazioni per i segreti ... 53
Accesso multi-account ... 55
Autorizzazioni per la rotazione ... 57
Policy per un ruolo di esecuzione della funzione di rotazione Lambda ... 57
Istruzione della policy per una chiave gestita dal cliente ... 58
Istruzione della policy per la strategia a utenti alternati ... 59
Esempi di policy di autorizzazione ... 61
Esempio: Autorizzazione per recuperare valori segreti individuali ... 62
Autorizzazione per il recupero di un gruppo di valori segreti in un batch ... 64
Esempio: il carattere jolly ... 65
Esempio: Autorizzazione per creare segreti ... 66
Esempio: autorizzazioni e VPC ... 67

Esempio: Controllare l'accesso ai segreti utilizzando i tag ... 69
Esempio: Limitare l'accesso alle identità con tag che corrispondono ai tag dei segreti ... 70
Esempio: Principale del servizio ... 71
Riferimento per le autorizzazioni ... 72
Operazioni di Secrets Manager ... 72
Risorse di Secrets Manager ... 94
Chiavi di condizione ... 94
Condizione `BlockPublicPolicy` ... 97
Condizioni indirizzo IP ... 98
Condizioni dell'endpoint VPC ... 98
Creazione e gestione di segreti ... 99
Creazione di un segreto del database ... 99
AWS CLI ... 101
AWS SDK ... 102
Struttura JSON di un segreto ... 102
Struttura del segreto di Amazon RDS Db2 ... 103
Struttura del segreto di MariaDB di Amazon RDS ... 103
Struttura del segreto di Amazon RDS e Amazon Aurora MySQL ... 104
Struttura del segreto di Oracle di Amazon RDS ... 105
Struttura del segreto di Amazon RDS e Amazon Aurora PostgreSQL ... 105
Struttura del segreto di Microsoft SQLServer di Amazon RDS ... 106
Struttura del segreto di Amazon DocumentDB ... 106
Struttura del segreto di Amazon Redshift ... 107
Struttura segreta di Amazon Redshift Serverless ... 108
Struttura ElastiCache segreta di Amazon ... 108
Creazione di un segreto ... 109
AWS CLI ... 110
AWS SDK ... 111
Aggiorna un valore segreto ... 111
AWS CLI ... 112
AWS SDK ... 112
Modifica la chiave di crittografia per un segreto ... 113
AWS CLI ... 114
Modificare un segreto ... 115
AWS CLI ... 116
AWS SDK ... 116

Scopri i segreti 117
AWS CLI 118
AWS SDK 119
Eliminare un segreto 119
AWS CLI 121
AWS SDK 121
Ripristino di un segreto 122
AWS CLI 122
AWS SDK 123
Replica un segreto ad altre Regioni 123
AWS CLI 125
SDK AWS 125
Risoluzione dei problemi 125
Promozione di un segreto di replica a segreto autonomo 126
AWS CLI 127
AWS SDK 127
Tag segreti 127
AWS CLI 128
AWS SDK 129
Recuperare segreti 130
In codice 130
All'interno di altri sistemi e servizi AWS 131
AWS CLI 131
Console AWS 132
Recupera segreti in un batch 132
Autorizzazioni per il recupero di segreti in un batch 132
AWS CLI 133
Connettersi a un database SQL 133
Stabilire una connessione a un database 135
Impostazione di una connessione specificando l'endpoint e la porta 138
Utilizzare il pooling di connessioni c3p0 per stabilire una connessione 141
Utilizzo del pooling di connessioni c3p0 per stabilire una connessione specificando l'endpoint e la porta 142
Applicazioni Java 143
SecretCache 145
SecretCacheConfiguration 147

SecretCacheHook 150
Applicazioni Python 150
SecretCache 152
SecretCacheConfig 153
SecretCacheHook 154
@InjectSecretString 155
@InjectKeywordedSecretString 156
Applicazioni .NET 156
SecretsManagerCache 159
SecretCacheConfiguration 162
ISecretCacheHook 163
Applicazioni Go 163
type Cache 165
type CacheConfig 166
type CacheHook 167
AWS Batch 167
AWS CloudFormation 168
Amazon Elastic Container Service 169
Amazon EKS 170
Installazione di ASCP 170
Configurazione del controllo degli accessi 171
Identificazione dei segreti montare 172
Risoluzione dei problemi 175
Tutorial 175
`SecretProviderClass` 177
GitHub lavori 181
Prerequisiti 181
Utilizzo 182
Denominazione delle variabili di ambiente 183
Esempi 184
AWS IoT Greengrass 186
AWS Lambda 186
Variabili di ambiente 189
Parameter Store 191
Rotazione dei segreti 192
Come funziona la rotazione 192

Rotazione gestita 195
Rotazione automatica per i segreti del database (console) 197
Fase 1: Scelta di una strategia di rotazione e (facoltativamente) creazione di un segreto del superutente 198
Fase 2: Configurazione della rotazione e creazione di una funzione di rotazione 199
Passaggio 3: (facoltativo) impostazione di condizioni di autorizzazione aggiuntive sulla funzione di rotazione 201
Fase 4: Configurazione dell'accesso alla rete per la funzione di rotazione 202
Fase 5: (facoltativo) personalizzazione della funzione di rotazione 203
Fasi successive 204
Rotazione automatica (console) 204
Fase 1: Configurare il segreto per la rotazione 205
Fase 2: Impostare le autorizzazioni per la funzione di rotazione 207
Fase 3: (Facoltativa) Impostazione di una condizione di autorizzazione aggiuntiva sulla funzione di rotazione 208
Fase 4: Configurazione dell'accesso alla rete per la funzione di rotazione 208
Fase 5: Scrittura del codice della funzione di rotazione 209
Passaggi successivi 212
Rotazione automatica (AWS CLI) 212
(Facoltativa) Fase 1: Creazione di un segreto del superutente 213
Fase 2: Scrittura del codice della funzione di rotazione 214
Fase 3: Creare la funzione Lambda e il ruolo di esecuzione 217
Fase 4: Configurazione dell'accesso alla rete 218
Fase 5: Configurare il segreto per la rotazione 220
Passaggi successivi 220
Rotazione immediata di un segreto 220
AWS CLI 221
Modelli di funzione di rotazione 221
Amazon RDS e Amazon Aurora 221
Amazon DocumentDB 226
Amazon Redshift 226
Amazon ElastiCache 227
Other type of secret (Altro tipo di segreto) 227
Espressioni di pianificazione 229
Espressioni rate 230
Espressioni cron 231

Risoluzione dei problemi della rotazione ... 236
Nessuna attività dopo "Found credentials in environment variables" (Trovate credenziali nelle variabili di ambiente) ... 237
Nessuna attività dopo "createSecret" ... 237
Errore: "Access to KMS is not allowed" ... 238
Errore: "Key is missing from secret JSON" (Chiave non presente nella struttura JSON del segreto) ... 239
Errore: "setSecret: Unable to log into database" (setSecret: impossibile accedere al database) ... 239
Errore: "Unable to import module 'lambda_function'" ... 242
Aggiornare una funzione di rotazione esistente da Python 3.7 a 3.9 ... 242
Segreti gestiti da altri servizi ... 246
Amazon AppFlow ... 247
AWS Glue DataBrew ... 247
AWS DataSync ... 247
AWS Direct Connect ... 247
Amazon Elastic Container Service ... 248
Amazon EventBridge ... 248
Marketplace AWS ... 248
AWS OpsWorks for Chef Automate ... 248
Amazon RDS e Aurora ... 248
Amazon Redshift ... 249
Editor di query v2 di Amazon Redshift ... 249
Endpoint VPC ... 250
Sottoreti condivise ... 251
AWS CloudFormation ... 252
Creazione di un segreto ... 253
JSON ... 253
YAML ... 254
Creare un segreto con le credenziali Amazon RDS con rotazione automatica ... 254
Crea un segreto con le credenziali Amazon Redshift ... 254
Crea un segreto con le credenziali Amazon DocumentDB ... 254
JSON ... 255
YAML ... 259
Come Secrets Manager utilizza AWS CloudFormation ... 262
AWS CDK ... 263

Monitorare i segreti 264
Accedi conAWS CloudTrail 264
AWS CLI 265
CloudTrail voci 265
Abbina gli eventi di Secrets Manager con EventBridge 271
Associa tutte le modifiche a un segreto specificato 271
Abbina gli eventi quando un valore segreto ruota 271
Monitora con CloudWatch 272
Parametri e dimensioni di Secrets Manager 272
Creare gli allarmi per monitorare i parametri di Secrets Manager 273
Amazon CloudWatch Synthetics canarini 273
Monitorare segreti programmati per l'eliminazione 274
Fase 1: Configurazione della consegna di file di log CloudTrail ai log CloudWatch 274
Fase 2: Creazione di allarme CloudWatch 275
Fase 3: Esecuzione del test dell'allarme CloudWatch 276
Convalida della conformità 277
Controllare i segreti per la conformità 279
..... 279
Aggrega segreti dal tuo Account AWS e Regioni AWS 280
Sicurezza in Secrets Manager 281
Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager 281
Protezione dei dati in Secrets Manager 284
Crittografia dei dati inattivi 284
Crittografia dei dati in transito 285
Riservatezza del traffico Internet 285
Gestione delle chiavi di crittografia 286
Crittografia e decrittografia del segreto 286
Che viene crittografato? 287
Processi di crittografia e decrittografia 288
Autorizzazioni per la chiave KMS 288
Come Secrets Manager utilizza la chiave KMS 289
Policy chiave della Chiave gestita da AWS (`aws/secretsmanager`) 291
Contesto di crittografia di Secrets Manager 293
Monitora l'interazione di Secrets Manager con AWS KMS 295
Sicurezza dell'infrastruttura 299
Resilienza 299

TLS post-quantistico 300
Risoluzione dei problemi 302
Messaggi di “Accesso negato” quando si inviano richieste a Secrets Manager 302
“Accesso negato” per le credenziali di sicurezza temporanee 303
Le modifiche apportate non sono sempre immediatamente visibili. 303
“Impossibile generare una chiave dati con una chiave KMS asimmetrica” durante la creazione di un segreto 304
Un AWS CLI o AWS operazione SDK non riesce a trovare il mio segreto da un ARN parziale .. 304
Questo segreto è gestito da un servizio AWS ed è necessario utilizzare tale servizio per aggiornarlo. 305
Quote 306
Quote di Secrets Manager 306
Aggiungi tentativi alla tua applicazione 309
Cronologia dei documenti 311
Aggiornamenti precedenti 311
..... cccxii

Che cos'è AWS Secrets Manager?

AWS Secrets Manager ti aiuta a gestire, recuperare e ruotare le credenziali del database, le credenziali delle applicazioni, i token OAuth, le chiavi API e altri segreti durante il loro ciclo di vita. Molti AWS servizi archiviano e utilizzano segreti in Secrets Manager.

Secrets Manager consente di migliorare l'assetto di sicurezza, perché non sono più necessarie credenziali a codifica fissa nel codice sorgente dell'applicazione. L'archiviazione delle credenziali in Secrets Manager aiuta a evitare possibili compromissioni da parte di chiunque che possa esaminare la tua applicazione o i tuoi componenti. Puoi sostituire le credenziali a codifica fissa con una chiamata a runtime al servizio Secrets Manager per recuperare le credenziali in modo dinamico quando ne hai bisogno.

Con Secrets Manager è possibile impostare un programma automatico di rotazione automatica per i segreti. In questo modo puoi sostituire i segreti a lungo termine con altri a breve termine, riducendo notevolmente il rischio di compromissione. Poiché le credenziali non sono più archiviate con l'applicazione, la rotazione delle credenziali non richiede più l'aggiornamento delle applicazioni e l'implementazione di modifiche ai client delle applicazioni.

Per altri segreti che potresti avere nell'organizzazione:

- AWS credenziali: consigliamo AWS Identity and Access Management.
- Chiavi di crittografia: consigliamo AWS Key Management Service.
- Chiavi SSH: consigliamo Amazon EC2 Instance Connect.
- Chiavi e certificati privati: consigliamo AWS Certificate Manager.

Inizia a usare Secrets Manager

Se sei un nuovo utente di Secrets Manager, inizia conConcetti uno dei seguenti tutorial:

- the section called “Sostituzione dei segreti codificati ”
- the section called “Sostituzione delle credenziali del database codificato ”
- the section called “Rotazione a utenti alternati”
- the section called “Rotazione a utente singolo”

Altre attività che puoi eseguire con i segreti:

- Creazione e gestione di segreti
- Controllare l'accesso ai tuoi segreti
- Recuperare segreti
- Rotazione dei segreti
- Monitorare i segreti
- Controllare i segreti per la conformità
- Crea segreti in AWS CloudFormation

Conformità agli standard

AWS Secrets Manager è stato sottoposto a controlli per i diversi standard e può far parte della vostra soluzione quando è necessario ottenere la certificazione di conformità. Per ulteriori informazioni, consulta Convalida della conformità.

Prezzi

Quando usi Secrets Manager paghi soltanto per ciò che utilizzi, senza tariffe minime o per la configurazione. Non è previsto alcun costo per i segreti che sono contrassegnati per l'eliminazione. Per l'elenco completo dei prezzi aggiornati, consulta la pagina dei prezziAWS Secrets Manager.

Puoi utilizzare il Chiave gestita da AWS `aws/secretsmanager` programma creato da Secrets Manager per crittografare i tuoi segreti gratuitamente. Se crei le tue chiavi KMS per crittografare i tuoi segreti, ti AWS addebiterà la tariffa attuale. AWS KMS Per ulteriori informazioni, consulta la sezione Prezzi diAWS Key Management Service.

Quando attivi la rotazione automatica (eccetto la rotazione gestita), Secrets Manager utilizza una AWS Lambda funzione per ruotare il segreto e ti viene addebitato il costo della funzione di rotazione alla velocità Lambda corrente. Per ulteriori informazioni, consulta la sezione Prezzi diAWS Lambda.

Se abiliti AWS CloudTrail il tuo account, puoi ottenere i log delle chiamate API inviate da Secrets Manager. Secrets Manager registra tutti gli eventi come eventi di gestione. AWS CloudTrail archivia gratuitamente la prima copia di tutti gli eventi di gestione. Tuttavia, ti potrebbero venire addebitati dei costi per lo storage dei log per Amazon S3 e Amazon SNS se abiliti la notifica. Inoltre, se imposti ulteriori trail, le copie aggiuntive di eventi di gestione potrebbero comportare dei costi. Per ulteriori informazioni, consultare Prezzi diAWS CloudTrail.

AWS servizi che utilizzano AWS Secrets Manager segreti

- AWS App Runner: consulta le sezioni Riferimenti alle variabili di ambiente e Gestione delle variabili di ambiente nella Guida per gli sviluppatori diAWS App Runner .
- AWS App2Container: consulta Gestisci i segreti per AWS App2Container nella Guida all'uso di App2Container.AWS
- AWS AppConfig: consulta la sezione Creazione di un profilo di configurazione in formato libero nella Guida per l'utente diAWS AppConfig .
- Amazon AppFlow — VediSegreti gestiti da altri servizi.
- AWS AppSync: consulta il Tutorial: Aurora Serverless nella Guida per gli sviluppatori diAWS AppSync .
- Amazon Athena: consulta la sezione Uso di Amazon Athena Federated Query nella Guida per l'utente di Amazon Athena.
- Amazon Aurora: vedi See. Segreti gestiti da altri servizi
- AWS CodeBuild— Vedi Registro privato con AWS Secrets Manager esempio di modulo CodeBuild nella Guida per l'AWS CodeBuild utente.
- AWS DataSync - Consultare Segreti gestiti da altri servizi.
- Amazon DataZone: consulta Creare un'origine dati per un database Amazon Redshift utilizzando una nuova AWS Glue connessione nella Amazon DataZone User Guide.
- AWS Direct Connect - Consultare Segreti gestiti da altri servizi.
- AWS Directory Service— Vedi Unire senza interruzioni un'istanza Linux EC2 alla tua directory AWS Managed Microsoft AD, Unire senza soluzione di continuità un'istanza Linux EC2 alla tua directory AD Connector e Unire senza problemi un'istanza Linux EC2 alla tua directory Simple AD nella Guida per l'utente.AWS Direct Connect
- Amazon DocumentDB (compatibile con MongoDB): consulta le sezioni the section called “Creazione di un segreto del database” e Gestione degli utenti di Amazon DocumentDB nella Guida per gli sviluppatori di Amazon DocumentDB.
- AWS Elastic Beanstalk: consulta la configurazione di Docker nella Guida per gli sviluppatori diAWS Elastic Beanstalk .
- Amazon Elastic Container Registry: consulta Creazione di una regola pull through cache nella Guida per l'utente di Amazon ECR.
- Amazon Elastic Container Service: consulta le varie sezioni Tutorial: specificazione dei dati sensibili utilizzando i segreti di Secrets Manager, Recupero dei segreti a livello di programmazione

tramite l'applicazione, Recupero dei segreti tramite variabili di ambiente, Recupero dei segreti per la configurazione della registrazione, Tutorial: utilizzo dei file system FSx per Windows File Server con Amazon ECS, Volumi FSx per Windows File Server e Autenticazione del registro privato per le attività nella Guida per gli sviluppatori di Amazon Elastic Container Service.

- Amazon Elastic Container Service Service Connect — VediSegreti gestiti da altri servizi.
- Amazon ElastiCache: consulta la ruotazione automatica delle password per gli utenti nella Amazon ElastiCache User Guide.
- AWS Elemental Live— Scopri come MediaConnect funziona la consegna da AWS Elemental Live a in fase di esecuzione nella Guida per l'utente di Elemental Live.
- AWS Elemental MediaConnect: consulta Crittografia a chiave statica in AWS Elemental MediaConnect nella Guida per l'utente diAWS Elemental MediaConnect .
- AWS Elemental MediaConvert— Vedi Uso di Kantar per la filigrana audio nelle AWS Elemental MediaConvert uscite nella Guida per l'utente.AWS Elemental MediaConvert
- AWS Elemental MediaLive— Vedi Configurazione MediaLive come entità affidabile nella Guida per l'utente. MediaLive
- AWS Elemental MediaPackage: consulta Accesso a Secrets Manager per l'autorizzazione CDN nella Guida per l'utente diAWS Elemental MediaPackage .
- AWS Elemental MediaTailor— Vedere Configurazione dell'autenticazione con token di AWS Secrets Manager accesso nella Guida per l'AWS Elemental MediaTailor utente.
- Amazon EMR in esecuzione su Amazon EC2: consulta la sezione Archiviazione dei dati di configurazione sensibili in Secrets Manager e Aggiunta di un repository basato su Git ad Amazon EMR nella Guida per la gestione di Amazon EMR.
- EMR Serverless: consulta la sezione Secrets Manager per la protezione dei dati con EMR serverless nella Guida per l'utente di Amazon EMR serverless.
- Amazon EventBridge — VediSegreti gestiti da altri servizi.
- Amazon FSx: consulta le sezioni Condivisioni di file e Migrazione delle configurazioni di condivisione di file su Amazon FSx nella Guida per l'utente di Amazon FSx per Windows File Server.
- AWS Glue DataBrew - Consultare Segreti gestiti da altri servizi.
- AWS Glue Studio — Vedi Tutorial: Using the AWS Glue Connector for Elasticsearch nella AWS Glue Developer Guide.
- AWS IoT SiteWise: consulta la sezione Configurazione dell'autenticazione dell'origine dati nella Guida per l'utente diAWS IoT SiteWise .

- Amazon Kendra: consulta la sezione Utilizzo di un'origine dati di database nella Guida per l'utente di Amazon Kendra.
- Flusso di video Amazon Kinesis: consulta la sezione Implementazione di flusso di video Amazon Kinesis Edge Agent in AWS IoT Greengrass nella Guida per gli sviluppatori di flusso di video Amazon Kinesis.
- AWS Launch Wizard— Vedi Configurazione AWS Launch Wizard per Active Directory nella Guida per l'AWS Launch Wizard utente.
- Amazon Lookout per le metriche: consulta le sezioni Utilizzo di Amazon RDS con Lookout per le metriche e Utilizzo di Amazon Redshift con Lookout per le metriche nella Guida per gli sviluppatori di Amazon Lookout per le metriche.
- Grafana gestito da Amazon: consulta Configurazione di Amazon Redshift nella Guida per l'utente di Grafana gestito da Amazon.
- AWS Managed Services: consulta la sezione AWS Secrets Manager (provisioning self-service di AMS) nella Guida avanzata per l'utente di AMS.
- Amazon Managed Streaming per Apache Kafka: consulta la sezione Autenticazione nome utente e password con AWS Secrets Manager nella Guida per gli sviluppatori di Amazon Managed Streaming per Apache Kafka.
- Amazon Managed Workflows for Apache Airflow: consulta Configurazione di una connessione Apache Airflow utilizzando un segreto di Secrets Manager e Utilizzo di una chiave segreta per una variabile Apache Airflow nella Guida AWS Secrets Manager per l'utente di Amazon Managed Workflows for Apache Airflow.
- Marketplace AWS - Consultare Segreti gestiti da altri servizi.
- AWS Migration Hub— Vedi Migrazione AWSe rihosting NetWeaver delle applicazioni SAP su Amazon EC2 nella Guida per l'utente di Orchestrator.AWS Migration Hub
- AWS OpsWorks for Chef Automate - Consultare Segreti gestiti da altri servizi.
- AWS Panorama: consulta la sezione Gestione degli stream delle videocamere in AWS Panorama nella Guida per gli sviluppatori diAWS Panorama .
- AWS ParallelCluster: consulta la sezione Integrazione di Active Directory nella Guida per l'utente diAWS ParallelCluster .
- Amazon Q — Vedi Concepts - Authentication nella Amazon Q Developer Guide.
- Amazon QuickSight: consulta la sezione Utilizzo dei AWS Secrets Manager segreti al posto delle credenziali del database in Amazon QuickSight nella Amazon QuickSight User Guide.
- Amazon RDS: consulta Segreti gestiti da altri servizi.

- Amazon Redshift: consulta Segreti gestiti da altri servizi le sezioni Archiviazione delle credenziali del database AWS Secrets Manager, Utilizzo dell'API dati di Amazon Redshift e Interrogazione di un database utilizzando l'editor di query nella Amazon Redshift Management Guide. the section called “Creazione di un segreto del database”
- Editor di query Amazon Redshift v2: consulta Segreti gestiti da altri servizi.
- Amazon SageMaker: consulta Associare repository Git a Amazon SageMaker Notebook Instances, Importare dati da Databricks (JDBC) e Importare dati da Snowflake nella Amazon Developer Guide. SageMaker
- AWS Schema Conversion Tool— Vedi Utilizzo AWS Secrets Manager nell'interfaccia utente nella Guida per l'utente. AWS SCT AWS Schema Conversion Tool
- AWS Toolkit for JetBrains: consulta la sezione Accesso ai cluster Amazon Redshift nella Guida per l'utenteAWS Toolkit for JetBrains .
- AWS Transfer Family: consulta le sezioni Autenticazione di base per i connettori AS2, Utilizzo di provider di identità personalizzati e Generazione e gestione di chiavi PGP nella Guida per l'utente diAWS Transfer Family .
- AWS Wickr: vedi Avvio del bot di conservazione dei dati nella AWS Wickr Administration Guide.

Accesso AWS Secrets Manager

Puoi usare il servizio Secrets Manager in uno dei modi seguenti:

- Console Secrets Manager
- Strumenti a riga di comando
- AWS SDK
- API query HTTPS
- AWS Secrets Manager endpoint

Console Secrets Manager

Puoi gestire i segreti utilizzando la console Secrets Manager basata su browser ed eseguire quasi tutte le attività correlate ai segreti utilizzando la console.

Strumenti a riga di comando

Gli strumenti da riga di AWS comando consentono di impartire comandi dalla riga di comando del sistema per eseguire Secrets Manager e altre AWS attività. Questa modalità può risultare più veloce e semplice rispetto all'uso della console. Gli strumenti da riga di comando possono essere utili se si desidera creare script per eseguire AWS attività.

Quando immetti i comandi in una shell dei comandi, c'è il rischio che la cronologia dei comandi sia accessibile o che le utilità abbiano accesso ai parametri dei comandi. Per informazioni, consulta the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

Gli strumenti della riga di comando utilizzano automaticamente l'endpoint predefinito per il servizio in una AWS regione. Puoi specificare un endpoint diverso per le tue richieste API. Per informazioni, consulta the section called “Endpoint di Secrets Manager”.

AWS fornisce due set di strumenti da riga di comando:

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

AWS SDK

Gli AWS SDK sono costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione. Gli SDK includono attività quali la firma crittografica delle richieste, la gestione degli errori e la ripetizione automatica delle richieste. Per scaricare e installare uno qualsiasi degli SDK, consulta Strumenti per Amazon Web Services.

Gli AWS SDK utilizzano automaticamente l'endpoint predefinito per il servizio in una regione. AWS Puoi specificare un endpoint diverso per le tue richieste API. Per informazioni, consulta the section called “Endpoint di Secrets Manager”.

Per la documentazione SDK, consulta:

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

API query HTTPS

L'API HTTPS Query offre l'accesso programmatico a Secrets Manager e AWS. L'API Query HTTPS consente di inviare richieste HTTPS direttamente al servizio.

Sebbene sia possibile effettuare chiamate dirette all'API Query HTTPS di Secrets Manager, consigliamo di usare invece uno degli SDK. L'SDK esegue molte attività utili che altrimenti dovresti effettuare manualmente. Ad esempio, gli SDK firmano automaticamente le richieste e convertono le risposte in una struttura sintatticamente appropriata alla tua lingua.

Per effettuare chiamate HTTPS a Secrets Manager, ti connetti a ???.

AWS Secrets Manager endpoint

Per connettersi a livello di codice a Secrets Manager, si utilizza un endpoint e l'URL del punto di ingresso per il servizio. Gli endpoint di Secrets Manager sono dual-stack, ossia supportano sia IPv4 sia IPv6.

Secrets Manager offre endpoint che supportano gli standard FIPS (Federal Information Processing Standard, standard federali per l'elaborazione delle informazioni) 140-2 in alcune Regioni.

Secrets Manager supporta TLS 1.2 e 1.3. Secrets Manager supporta PQTLS in tutte le Regioni ad eccezione di quelle cinesi.

> **Note**
>
> L' AWS SDK Python e il AWS CLI tentativo di chiamare IPv6 e poi IPv4 in sequenza, quindi se non hai abilitato IPv6, potrebbe volerci del tempo prima che la chiamata scada e riprovi con IPv4. Per evitare questo problema, puoi disabilitare completamente IPv6 o migrare a IPv6.

Di seguito sono riportati gli endpoint del servizio per il Secrets Manager. Si noti che la denominazione è diversa dalla tipica convenzione di denominazione dual-stack.

Nome della regione	Regione	Endpoint	Protocollo	
US East (Ohio)	us-east-2	secretsmanager.us-east-2.amazonaws.com secretsmanager-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
US East (N. Virginia)	us-east-1	secretsmanager.us-east-1.amazonaws.com secretsmanager-fips.us-east-1.amazonaws.com	HTTPS HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Stati Uniti occidentali (California settentrionale)	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 (Città del Capo)	af-south-1	secretsmanager.af-south-1.amazonaws.com	HTTPS	
Asia Pacifico (Hong Kong)	ap-east-1	secretsmanager.ap-east-1.amazonaws.com	HTTPS	
Asia Pacific (Hyderabad)	ap-south-2	secretsmanager.ap-south-2.amazonaws.com	HTTPS	
Asia Pacifico (Giacarta)	ap-southeast-3	secretsmanager.ap-southeast-3.amazonaws.com	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacifico (Melbourne)	ap-southeast-4	secretsmanager.ap-southeast-4.amazonaws.com	HTTPS	
Asia Pacifico (Mumbai)	ap-south-1	secretsmanager.ap-south-1.amazonaws.com	HTTPS	
Asia Pacifico (Osaka-Locale)	ap-northeast-3	secretsmanager.ap-northeast-3.amazonaws.com	HTTPS	
Asia Pacifico (Seoul)	ap-northeast-2	secretsmanager.ap-northeast-2.amazonaws.com	HTTPS	
Asia Pacifico (Singapore)	ap-southeast-1	secretsmanager.ap-southeast-1.amazonaws.com	HTTPS	
Asia Pacific (Sydney)	ap-southeast-2	secretsmanager.ap-southeast-2.amazonaws.com	HTTPS	
Asia Pacifico (Tokyo)	ap-northeast-1	secretsmanager.ap-northeast-1.amazonaws.com	HTTPS	
Canada (Centrale)	ca-central-1	secretsmanager.ca-central-1.amazonaws.com secretsmanager-fips.ca-central-1.amazonaws.com	HTTPS HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Canada occidentale (Calgary)	ca-west-1	secretsmanager.ca-west-1.amazonaws.com secretsmanager-fips.ca-west-1.amazonaws.com	HTTPS HTTPS	
Europa (Francoforte)	eu-central-1	secretsmanager.eu-central-1.amazonaws.com	HTTPS	
Europa (Irlanda)	eu-west-1	secretsmanager.eu-west-1.amazonaws.com	HTTPS	
Europa (Londra)	eu-west-2	secretsmanager.eu-west-2.amazonaws.com	HTTPS	
Europa (Milano)	eu-south-1	secretsmanager.eu-south-1.amazonaws.com	HTTPS	
Europa (Parigi)	eu-west-3	secretsmanager.eu-west-3.amazonaws.com	HTTPS	
Europa (Spagna)	eu-south-2	secretsmanager.eu-south-2.amazonaws.com	HTTPS	
Europa (Stoccolma)	eu-north-1	secretsmanager.eu-north-1.amazonaws.com	HTTPS	
Europa (Zurigo)	eu-central-2	secretsmanager.eu-central-2.amazonaws.com	HTTPS	
Israele (Tel Aviv)	il-central-1	secretsmanager.il-central-1.amazonaws.com	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Medio Oriente (Bahrein)	me-south-1	secretsmanager.me-south-1.amazonaws.com	HTTPS	
Medio Oriente (Emirati Arabi Uniti)	me-central-1	secretsmanager.me-central-1.amazonaws.com	HTTPS	
Sud America (São Paulo)	sa-east-1	secretsmanager.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	secretsmanager.us-gov-east-1.amazonaws.com secretsmanager-fips.us-gov-east-1.amazonaws.com	HTTPS HTTPS	
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	secretsmanager.us-gov-west-1.amazonaws.com secretsmanager-fips.us-gov-west-1.amazonaws.com	HTTPS HTTPS	

Concetti AWS Secrets Manager

I seguenti concetti sono importanti per comprendere il funzionamento di Secrets Manager.

- Secret
- Versione
- Rotazione
- Strategia di rotazione

Secret

In Secrets Manager, un segreto è costituito dalle informazioni del segreto, dal valore del segreto e dai metadati sul segreto. Un valore segreto può essere di tipo stringa o binario. Per archiviare più valori stringa in un segreto, è consigliabile utilizzare una stringa di testo JSON con coppie chiave/valore, ad esempio:

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

I metadati di un segreto includono:

- Amazon Resource Name (ARN) ha il seguente formato:

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

  Secrets Manager include sei caratteri casuali alla fine del nome del segreto per garantire che l'ARN del segreto sia univoco. Se il segreto originale viene eliminato e quindi viene creato un nuovo segreto con lo stesso nome, i due segreti hanno ARN diversi grazie a questi caratteri. Gli utenti con accesso al vecchio segreto non ottengono automaticamente l'accesso al nuovo segreto perché gli ARN sono diversi.
- Il nome del segreto, una descrizione, una policy di risorse e i tag.

- L'ARN per un Chiave di crittografia, un AWS KMS key che Secrets Manager utilizza per crittografare e decrittografare il valore del segreto. Secrets Manager archivia sempre il testo del segreto in un modulo crittografato e crittografa sempre il segreto in transito. Per informazioni, consultare the section called “Crittografia e decrittografia del segreto”.
- Informazioni su come ruotare il segreto, se si imposta la rotazione. Per informazioni, consultare the section called “Rotazione”.

Secrets Manager utilizza le policy di autorizzazione IAM per garantire che solo gli utenti autorizzati possano accedere al segreto o modificarlo. Per informazioni, consultare Autenticazione e controllo degli accessi per AWS Secrets Manager.

Un segreto ha versioni che conservano copie del valore del segreto crittografato. Quando si cambia il valore del segreto o il segreto viene ruotato, Secrets Manager crea una nuova versione. Per informazioni, consultare the section called “Versione”.

È possibile utilizzare un segreto in più Regioni AWS tramite replica di esso. Quando si replica un segreto, si crea una copia dell'originale o segreto primario chiamato un Segreto di replica. Il segreto di replica rimane collegato al segreto primario. Per informazioni, consultare the section called “Replica un segreto ad altre Regioni”.

Per informazioni, consultare Creazione e gestione di segreti.

Versione

Un segreto ha versioni che conservano copie del valore del segreto crittografato. Quando si cambia il valore del segreto o il segreto viene ruotato, Secrets Manager crea una nuova versione.

Secrets Manager non memorizza la cronologia lineare dei segreti con le rispettive versioni. Tiene invece traccia di tre versioni specifiche, etichettandole come segue:

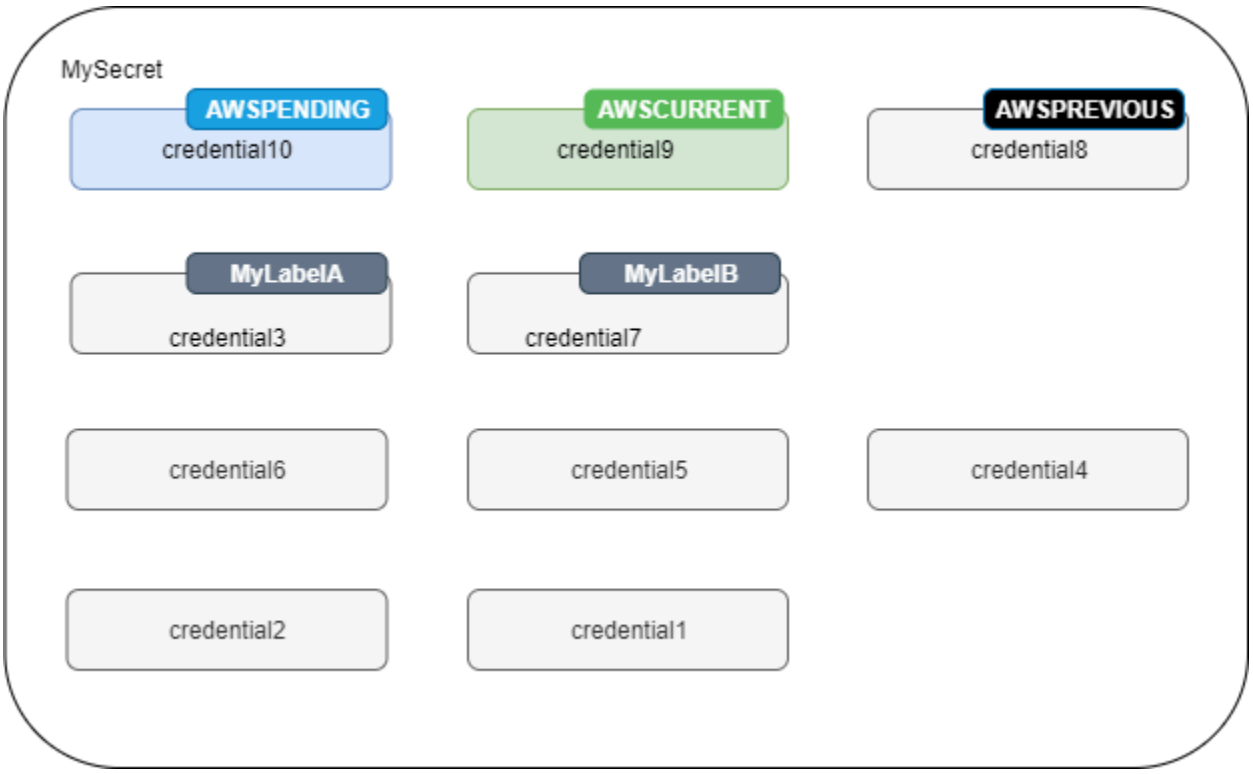
- Versione corrente: `AWSCURRENT`
- Versione precedente: `AWSPREVIOUS`
- Versione in sospeso (durante la rotazione): `AWSPENDING`

Un segreto ha sempre una versione etichettata `AWSCURRENT`; per impostazione predefinita, quando recuperi il valore del segreto, Secrets Manager restituisce tale versione.

Puoi anche etichettare le versioni con etichette personalizzate richiamando `update-secret-version-stage` nella AWS CLI. È possibile assegnare alle versioni di un segreto fino a 20 etichette. Due versioni di un segreto non possono avere la stessa etichetta di gestione temporanea. Le versioni possono avere più etichette.

Secrets Manager non rimuove mai le versioni etichettate, ma le versioni senza etichetta sono considerate obsolete. Secrets Manager rimuove le versioni obsolete quando il loro numero diventa maggiore di 100. Secrets Manager non rimuove le versioni create meno di 24 ore prima.

La figura seguente mostra un segreto che include versioni etichettate AWS e versioni etichettate dal cliente. Le versioni senza etichette sono considerate obsolete e verranno rimosse da Secrets Manager in un momento futuro.



Rotazione

Rotazione è il processo mediante il quale si aggiorna periodicamente il segreto per rendere più difficile l'accesso alle credenziali da parte di un utente malintenzionato. In Secrets Manager, è possibile impostare la rotazione automatica per i segreti. Quando Secrets Manager ruota un segreto, vengono aggiornate le credenziali sia del segreto che del database o del servizio. Per informazioni, consultare Rotazione dei segreti .

> **Tip**
>
> Per alcuni Segreti gestiti da altri servizi, si utilizza la rotazione gestita. Per utilizzare Rotazione gestita, è necessario dapprima creare il segreto tramite il servizio di gestione.

Strategia di rotazione

Gestione dei segreti offre due strategie di rotazione:

- Strategia di rotazione a utente singolo
- Strategia di rotazione a utenti alternati

Strategia di rotazione a utente singolo

Questa strategia aggiorna le credenziali per un utente in un unico segreto. Poiché gli utenti non possono modificare le proprie password, per le istanze Amazon RDS Db2 è necessario fornire le credenziali di amministratore in un segreto separato. Questa è la strategia di rotazione più semplice ed è adatta alla maggior parte dei casi d'uso. In particolare, ti consigliamo di utilizzare questa strategia per le credenziali per utenti occasionali (ad hoc) o interattivi.

Quando il segreto ruota, le connessioni aperte al database non vengono eliminate. Mentre si verifica la rotazione, tra la modifica della password nel database e l'aggiornamento del segreto corrispondente passa un breve periodo di tempo. Durante questo periodo di tempo, c'è un basso rischio che il database neghi le chiamate che utilizzano le credenziali ruotate. È possibile mitigare questo rischio con una strategia per nuovi tentativi appropriata. Dopo la rotazione, le nuove connessioni utilizzano le nuove credenziali.

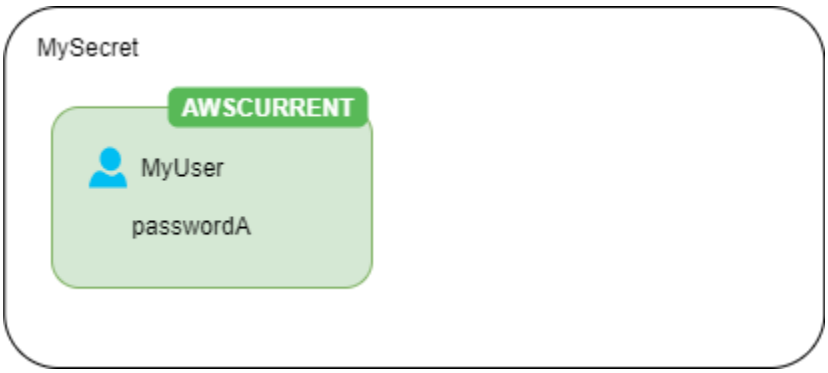
Strategia di rotazione a utenti alternati

Questa strategia aggiorna le credenziali per due utenti in un unico segreto. Viene creato il primo utente e, durante la prima rotazione, la funzione di rotazione lo clona creando il secondo. Ogni volta che il segreto ruota, la funzione di rotazione alterna quale password dell'utente viene aggiornata. Poiché la maggior parte degli utenti non dispone dell'autorizzazione per clonarsi, è necessario fornire le credenziali relative a un `superuser` in un altro segreto. Si consiglia di utilizzare la strategia di rotazione per singolo utente quando gli utenti clonati nel database non hanno le stesse autorizzazioni dell'utente originale e per le credenziali di utenti occasionali (ad hoc) o interattivi.

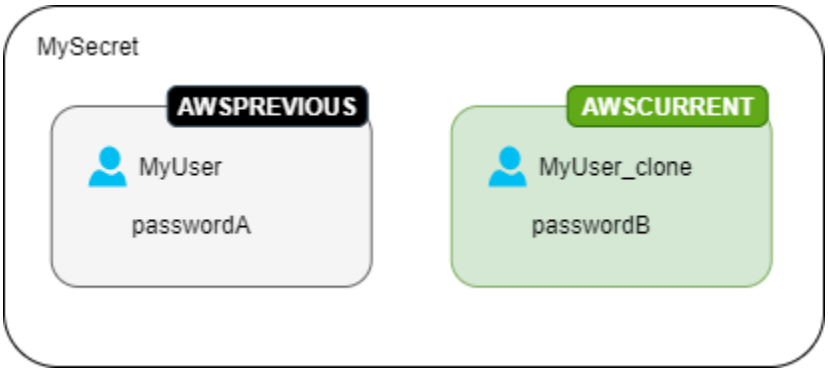
Questa strategia è adatta per i database con modelli di autorizzazione in cui un ruolo possiede le tabelle del database e un secondo ruolo ha l'autorizzazione per accedervi. È adatta anche all'uso in applicazioni che richiedono una disponibilità elevata. Se un'applicazione recupera il segreto durante la rotazione, ottiene comunque un set di credenziali valido. Dopo la rotazione, entrambe le credenziali `user` e `user_clone` sono valide. Ci sono anche meno possibilità che le applicazioni ottengano un rifiuto durante questo tipo di rotazione rispetto alla rotazione a utente singolo. Se il database è ospitato in una server farm dove la modifica della password richiede tempo per propagarsi a tutti i server, esiste il rischio che il database rifiuti le chiamate che utilizzano le nuove credenziali. È possibile mitigare questo rischio con una strategia per nuovi tentativi appropriata.

Secrets Manager crea l'utente clonato con le stesse autorizzazioni dell'utente originale. Se modifichi le autorizzazioni dell'utente originale dopo la creazione del clone, devi modificare anche le autorizzazioni dell'utente clonato.

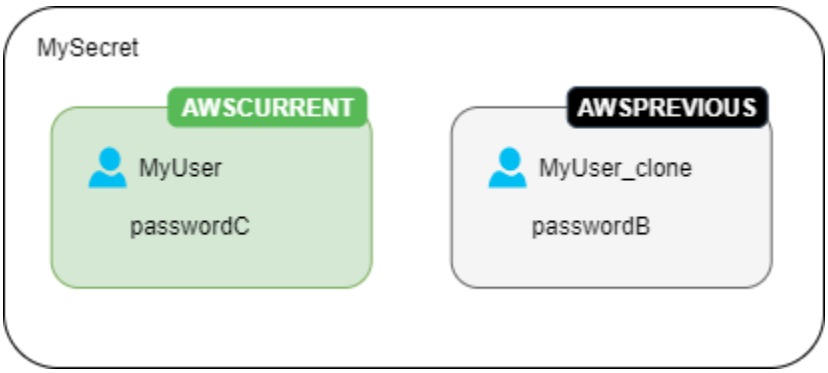
Ad esempio, se crei un segreto con le credenziali di un utente del database, il segreto contiene una versione con tali credenziali.



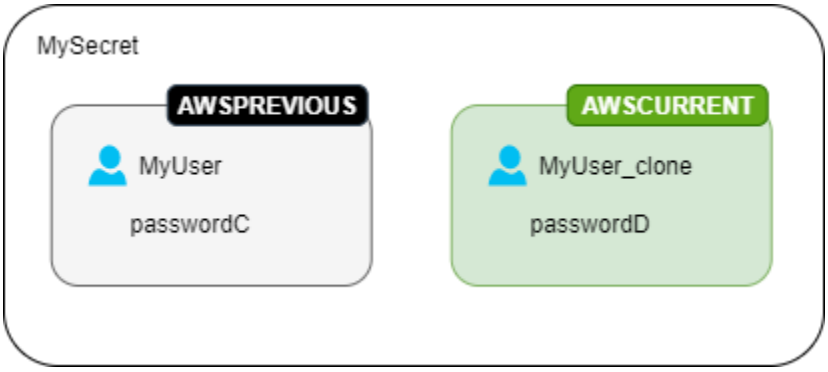
Prima rotazione: la funzione di rotazione crea un clone dell'utente con una password generata e tali credenziali diventano la versione del segreto corrente.



Seconda rotazione: la funzione di rotazione aggiorna la password dell'utente originale.



Terza rotazione: la funzione di rotazione aggiorna la password dell'utente clonato.



Tutorial di AWS Secrets Manager

Argomenti

- Ricerca di segreti non protetti nel codice con revisore Amazon CodeGuru
- Spostamento dei segreti codificati in AWS Secrets Manager
- Sposta le credenziali del database codificate in AWS Secrets Manager
- Configurazione della rotazione a utenti alternati per AWS Secrets Manager
- Configurazione di una rotazione a utente singolo per AWS Secrets Manager

Ricerca di segreti non protetti nel codice con revisore Amazon CodeGuru

Revisore Amazon CodeGuru è un servizio che utilizza l'analisi dei programmi e il machine learning per rilevare potenziali difetti difficili da trovare per gli sviluppatori e offre suggerimenti per migliorare il codice Java e Python. Revisore CodeGuru si integra con Secrets Manager per individuare i segreti non protetti nel tuo codice. Per i tipi di segreti che può trovare, consulta la sezione Types of secrets detected by CodeGuru Reviewer (Tipi di segreti rilevati da revisore CodeGuru) nella Guida per l'utente di revisore Amazon CodeGuru.

Dopo aver trovato i segreti codificati, completa le seguenti operazioni per sostituirli:

- the section called “Sostituzione delle credenziali del database codificato ”
- the section called “Sostituzione dei segreti codificati ”

Spostamento dei segreti codificati in AWS Secrets Manager

Se nel codice sono presenti segreti in testo semplice, si consiglia di ruotarli e poi archiviarli in Secrets Manager. Lo spostamento delle credenziali in Secrets Manager risolve il problema di visibilità del segreto a chiunque veda il codice, perché andando avanti il codice recupera il segreto direttamente da Secrets Manager. La rotazione del segreto revoca il segreto codificato corrente in modo che non sia più valido.

Per i segreti delle credenziali del database, consulta Sposta le credenziali del database codificate in AWS Secrets Manager.

Prima di iniziare, è necessario determinare chi ha bisogno di accedere al segreto. Consigliamo di utilizzare due ruoli IAM per gestire l'autorizzazione al tuo segreto:

- Un ruolo che gestisce i segreti nella tua organizzazione. Per ulteriori informazioni, consulta the section called “Autorizzazioni di amministrazione di Secrets Manager”. Questo ruolo sarà utilizzato per creare e ruotare il segreto.
- Un ruolo che può utilizzare le credenziali in fase di runtime, ad esempio *RoleToRetrieveSecretAtRuntime* in questo tutorial. Il tuo codice assume questo ruolo per recuperare il segreto. In questo tutorial, si concede al ruolo solo l'autorizzazione per recuperare il valore di un segreto e si concede l'autorizzazione utilizzando la policy delle risorse del segreto. Per le alternative, consulta the section called “Fasi successive”.

Fasi:

- Fase 1: creazione del segreto
- Fase 2: aggiornamento del codice
- Fase 3: aggiornamento del segreto
- Fasi successive

Fase 1: creazione del segreto

Il primo passo consiste nel copiare il segreto codificato esistente in Secrets Manager. Se il segreto è correlato a una risorsa AWS, archivialo nella stessa regione della risorsa. In caso contrario, conservalo nella regione con latenza più bassa per il tuo caso d'uso.

Creazione di un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:
   a. Per Secret type (Tipo di segreto), scegli Other type of secret (Altro tipo di segreto).
   b. Inserisci il tuo segreto come coppie chiave/valore o in testo semplice. Alcuni esempi:

Coppie chiave-valore della chiave API:

ClientID : *my_client_id*

ClientSecret : *wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

Coppia chiave-valore delle credenziali:

Username : *saanvis*

Password : *EXAMPLE-PASSWORD*

Testo semplice del token OAuth:

AKIAI44QH8DHBEXAMPLE

Testo semplice del certificato digitale:

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

Testo semplice della chiave privata:

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

c. In Encryption key (Chiave di crittografia), scegli aws/secretsmanager per utilizzare la Chiave gestita da AWS per Secrets Manager. L'utilizzo di questa chiave non prevede costi aggiuntivi. Puoi inoltre utilizzare la tua chiave gestita dal cliente, ad esempio per accedere al segreto da un altro Account AWS. Per informazioni sui costi di utilizzo di una chiave gestita dal cliente, consulta la sezione Prezzi.

d. Seleziona Successivo.

4. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:

   a. Inserisci un Secret name (Nome del segreto) e una Description (Descrizione) descrittivi.

b. In Resource permissions (Autorizzazioni della risorsa), scegli Edit permissions (Modifica autorizzazioni). Incolla la seguente policy, che consente a *RoleToRetrieveSecretAtRuntime* di recuperare il segreto, quindi scegli Save (Salva).

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

c. Nella parte inferiore della pagina scegli Next (Avanti).

5. Nella pagina Configure rotation (Configura la rotazione), mantieni la rotazione disattivata. Seleziona Successivo.
6. Nella pagina Review (Revisione), rivedi i dettagli dei segreti e quindi scegli Store (Archivia).

Fase 2: aggiornamento del codice

Per poter recuperare il segreto, il codice deve assumere il ruolo IAM *RoleToRetrieveSecretAtRuntime*. Per ulteriori informazioni, consulta Passaggio a un ruolo IAM (API AWS).

Successivamente, aggiorna il codice per recuperare il segreto da Secrets Manager utilizzando il codice di esempio fornito da Secrets Manager.

Ricerca del codice di esempio

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nella pagina dell'elenco Secrets (Segreti) scegli il segreto.
3. Scorrere verso il basso fino a Sample code (Codice di esempio). Scegli il linguaggio di programmazione, quindi copia il frammento di codice.

Nell'applicazione, rimuovi il segreto codificato e incolla il frammento di codice. A seconda del linguaggio del codice, potrebbe essere necessario aggiungere una chiamata alla funzione o al metodo nel frammento.

Verifica che la tua applicazione funzioni come previsto con il segreto al posto del segreto codificato.

Fase 3: aggiornamento del segreto

L'ultima fase è la revoca e l'aggiornamento del segreto codificato. Fai riferimento all'origine del segreto per trovare le istruzioni per revocare e aggiornare il segreto. Ad esempio, potrebbe essere necessario disattivare il segreto corrente e generarne uno nuovo.

Aggiornamento del segreto con il nuovo valore

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Secrets (Segreti), quindi scegli il segreto.
3. Nella pagina Secret details (Dettagli del segreto), scorri verso il basso e scegli Retrieve secret value (Recupera il valore del segreto), quindi Edit (Modifica).
4. Aggiorna il segreto, quindi scegli Save (Salva).

Quindi, verifica che la tua applicazione funzioni come previsto con il nuovo segreto.

Fasi successive

Dopo aver rimosso un segreto codificato dal codice, ecco alcuni concetti da considerare:

- Per trovare segreti codificati nelle applicazioni Java e Python, ti consigliamo Amazon CodeGuru Reviewer.
- È possibile migliorare le prestazioni e ridurre i costi tramite la memorizzazione nella cache dei segreti. Per ulteriori informazioni, consulta Recuperare segreti.
- Per i segreti a cui accedi da più regioni, considera la possibilità di replicare il tuo segreto per migliorare la latenza. Per ulteriori informazioni, consulta the section called “Replica un segreto ad altre Regioni”.
- In questo tutorial, hai concesso al ruolo *RoleToRetrieveSecretAtRuntime* solo l'autorizzazione per recuperare il valore del segreto. Per concedere al ruolo più autorizzazioni, ad

esempio per ottenere metadati sul segreto o per visualizzare un elenco di segreti, consulta the section called “Esempi di policy di autorizzazione”.

- In questo tutorial, hai concesso l'autorizzazione al ruolo *RoleToRetrieveSecretAtRuntime* tramite la policy della risorsa del segreto. Per altri modi per concedere l'autorizzazione, consulta the section called “Allegare un policy di autorizzazione a un'identità”.

Sposta le credenziali del database codificate in AWS Secrets Manager

Se nel codice sono presenti credenziali di database in testo semplice, si consiglia di spostare le credenziali in Secrets Manager e quindi ruotarle immediatamente. Lo spostamento delle credenziali in Secrets Manager risolve il problema di visibilità delle credenziali a chiunque veda il codice, perché andando avanti il codice recupera le credenziali direttamente da Secrets Manager. La rotazione del segreto aggiorna la password e quindi revoca la password codificata corrente in modo che non sia più valida.

Per i database Amazon RDS, Amazon Redshift e Amazon DocumentDB, utilizza la procedura descritta in questa pagina per spostare le credenziali codificate in Secrets Manager. Per altri tipi di credenziali e altri segreti, consulta the section called “Sostituzione dei segreti codificati ”.

Prima di iniziare, è necessario determinare chi ha bisogno di accedere al segreto. Consigliamo di utilizzare due ruoli IAM per gestire l'autorizzazione al tuo segreto:

- Un ruolo che gestisce i segreti nella tua organizzazione. Per ulteriori informazioni, consulta the section called “Autorizzazioni di amministrazione di Secrets Manager”. Questo ruolo sarà utilizzato per creare e ruotare il segreto.
- Un ruolo che può utilizzare le credenziali in fase di esecuzione, *RoleToRetrieveSecretAtRuntime*in questo tutorial. Il tuo codice assume questo ruolo per recuperare il segreto.

Fasi:

- Fase 1: creazione del segreto
- Fase 2: aggiornamento del codice
- Fase 3: rotazione del segreto
- Passaggi successivi

Fase 1: creazione del segreto

Il primo passo consiste nel copiare le credenziali codificate esistenti in un segreto in Secrets Manager. Per la latenza più bassa, archivia il segreto nella stessa regione del database.

Per creare un segreto

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:
   a. Per Secret type (Tipo di segreto), scegli il tipo di credenziali del database da archiviare:
      - Database Amazon RDS
      - Database di Amazon DocumentDB
      - Data warehouse Amazon Redshift.
      - Per altri tipi di segreti, consulta Sostituzione dei segreti codificati.
   b. Per Credenziali, inserisci le credenziali codificate per il database.
   c. In Encryption key (Chiave di crittografia), scegli aws/secretsmanager per utilizzare la Chiave gestita da AWS per Secrets Manager. L'utilizzo di questa chiave non prevede costi aggiuntivi. Puoi inoltre utilizzare la tua chiave gestita dal cliente, ad esempio per accedere al segreto da un altro Account AWS. Per informazioni sui costi di utilizzo di una chiave gestita dal cliente, consulta la sezione Prezzi.
   d. Per Database, scegli il database.
   e. Seleziona Successivo.
4. Nella pagina Configure secret (Configura il segreto), effettua le seguenti operazioni:
   a. Inserisci un Secret name (Nome del segreto) e una Description (Descrizione) descrittivi.
   b. In Resource permissions (Autorizzazioni della risorsa), scegli Edit permissions (Modifica autorizzazioni). Incolla la seguente policy, che ***RoleToRetrieveSecretAtRuntime***consente di recuperare il segreto, quindi scegli Salva.

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

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

c. Nella parte inferiore della pagina scegli Next (Avanti).

5. Nella pagina Configure rotation (Configura la rotazione), mantieni la rotazione disattivata per il momento. La attiverai più tardi. Seleziona Successivo.
6. Nella pagina Review (Revisione), rivedi i dettagli dei segreti e quindi scegli Store (Archivia).

Fase 2: aggiornamento del codice

Il codice deve assumere il ruolo IAM *RoleToRetrieveSecretAtRuntime*per poter recuperare il segreto. Per ulteriori informazioni, consulta Passare a un ruolo IAM (AWS API).

Successivamente, aggiorna il codice per recuperare il segreto da Secrets Manager utilizzando il codice di esempio fornito da Secrets Manager.

Ricerca del codice di esempio

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nella pagina dell'elenco Secrets (Segreti) scegli il segreto.
3. Scorrere verso il basso fino a Sample code (Codice di esempio). Scegli il linguaggio, quindi copia il frammento di codice.

Nell'applicazione, rimuovi le credenziali codificate e incolla il frammento di codice. A seconda del linguaggio del codice, potrebbe essere necessario aggiungere una chiamata alla funzione o al metodo nel frammento.

Verifica che la tua applicazione funzioni come previsto con il segreto al posto delle credenziali codificate.

Fase 3: rotazione del segreto

L'ultimo passo è revocare le credenziali codificate ruotando il segreto. La rotazione è il processo di aggiornamento periodico di un segreto. Quando si ruota un segreto, vengono aggiornate le credenziali sia nel segreto che nel database. Secrets Manager può ruotare automaticamente un segreto su un programma configurato.

Parte del processo di configurazione della rotazione è garantire che la funzione di rotazione Lambda possa accedere sia a Secrets Manager che al database. Quando si attiva la rotazione automatica, Secrets Manager crea la funzione di rotazione Lambda nello stesso VPC del database in modo che abbia accesso di rete al database. La funzione di rotazione Lambda deve anche essere in grado di effettuare chiamate a Secrets Manager per aggiornare il segreto. Ti consigliamo di creare un endpoint Secrets Manager nel VPC in modo che le chiamate da Lambda a Secrets Manager non lascino l'infrastruttura. AWS Per istruzioni, consulta Endpoint VPC.

Attivazione della rotazione

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nella pagina dell'elenco Secrets (Segreti) scegli il segreto.
3. Nella pagina Secret details (Dettagli del segreto), nella sezione Rotation configuration (Configurazione rotazione) scegli Edit rotation (Modifica rotazione).
4. Nella finestra di dialogo Edit rotation configuration (modifica configurazione rotazione), procedi come indicato di seguito:
    a. Attiva Automatic rotation (Rotazione automatica).
    b. In Rotation schedule (Programma di rotazione), inserisci il programma nel fuso orario UTC.
    c. Scegli Rotate immediately when the secret is stored (Ruota immediatamente quando viene memorizzato il segreto) per ruotare il segreto al salvataggio delle modifiche.
    d. In Rotation function (Funzione di rotazione), scegli Create a new Lambda function (Crea una nuova funzione Lambda) ed immetti un nome per la nuova funzione. Secrets Manager aggiunge "SecretsManager" all'inizio del nome della funzione.
    e. Per Strategia di rotazione scegli Utente singolo.
    f. Selezionare Salva.

Verifica che il segreto sia stato ruotato

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Secrets (Segreti), quindi scegli il segreto.
3. Nella pagina Secret details (Dettagli del segreto), scorri e scegli Retrieve secret value (Recupera il valore del segreto).

   Se il valore del segreto è cambiato, allora la rotazione è riuscita. Se il valore segreto non è cambiato, è necessario farlo Risoluzione dei problemi della rotazione consultando i CloudWatch log per verificare la funzione di rotazione.

Verifica che la tua applicazione funzioni come previsto con il segreto ruotato.

Passaggi successivi

Dopo aver rimosso un segreto codificato dal codice, ecco alcuni concetti da considerare:

- È possibile migliorare le prestazioni e ridurre i costi tramite la memorizzazione nella cache dei segreti. Per ulteriori informazioni, consulta Recuperare segreti.
- È possibile scegliere un diverso programma di rotazione. Per ulteriori informazioni, consulta the section called “Espressioni di pianificazione”.
- Per trovare segreti codificati nelle tue applicazioni Java e Python, ti consigliamo Amazon Reviewer. CodeGuru

Configurazione della rotazione a utenti alternati per AWS Secrets Manager

In questo tutorial, imparerai come impostare la rotazione a utenti alternati per un segreto che contiene le credenziali del database. La rotazione a utenti alternati è una strategia di rotazione in cui Secrets Manager clona l'utente e quindi alterna quali credenziali dell'utente vengono aggiornate. Questa strategia è una buona scelta se hai bisogno di disponibilità elevata per il segreto, perché uno degli utenti alternati ha le credenziali correnti per il database mentre l'altro è in fase di aggiornamento. Per ulteriori informazioni, consulta the section called “Utenti alternati”.

Per impostare la rotazione a utenti alternati, sono necessari due segreti:

- Un segreto con le credenziali da ruotare.
- Un secondo segreto con credenziali di amministratore.

  Questo utente dispone delle autorizzazioni per clonare il primo utente e modificare la password. In questo tutorial, chiedi ad Amazon RDS di creare questo segreto per un utente amministratore. Amazon RDS gestisce anche la rotazione delle password dell'amministratore. Per ulteriori informazioni, consulta the section called “Rotazione gestita”.

La prima parte di questo tutorial tratta la configurazione di un ambiente realistico. Per mostrare come funziona la rotazione, questo tutorial utilizza un esempio di database Amazon RDS MySQL. Per motivi di sicurezza, il database si trova in un VPC che limita l'accesso a Internet. Per connettersi al database dal computer locale tramite Internet, è necessario utilizzare un host bastione, un server nel VPC in grado di connettersi al database, ma che consente anche connessioni SSH da Internet. L'host bastione in questo tutorial è un'istanza Amazon EC2 e i gruppi di sicurezza per l'istanza impediscono altri tipi di connessioni.

Al termine del tutorial, si consiglia di ripulire le risorse del tutorial. Non utilizzarle in un ambiente di produzione.

La rotazione di Secrets Manager utilizza una funzione AWS Lambda per aggiornare il segreto e il database. Per ulteriori informazioni sui costi di utilizzo della funzione Lambda, consulta la sezione Prezzi.

Tutorial:

- Autorizzazioni
- Prerequisiti
- Fase 1: creazione di un utente del database Amazon RDS
- Fase 2: creazione di un segreto per le credenziali dell'utente
- Fase 3: eseguire il test del segreto ruotato
- Fase 4: eliminazione delle risorse
- Fasi successive

Autorizzazioni

Per i prerequisiti del tutorial, hai bisogno di autorizzazioni di amministrazione per il tuo Account AWS. In un ambiente di produzione, è consigliabile utilizzare ruoli diversi per ciascun passaggio.

Ad esempio, un ruolo con le autorizzazioni dell'amministratore del database creerebbe il database Amazon RDS e un ruolo con autorizzazioni di amministrazione di rete configurerebbe il VPC e i gruppi di sicurezza. Per i passaggi del tutorial, suggeriamo di continuare a utilizzare la stessa identità.

Per informazioni su come configurare le autorizzazioni in un ambiente di produzione, consulta Autenticazione e controllo degli accessi.

Prerequisiti

Per questo tutorial hai bisogno dei seguenti elementi:

- Prerequisito A: Amazon VPC
- Prerequisito B: istanza Amazon EC2
- Prerequisito C: database Amazon RDS e un segreto in Secrets Manager per le credenziali di amministratore
- Prerequisito D: consenti al computer locale di connettersi all'istanza EC2.

Prerequisito A: Amazon VPC

In questa fase viene creato un VPC in cui è necessario avviare un database Amazon RDS e un'istanza Amazon EC2. In una fase successiva, utilizzerai il tuo computer per connetterti tramite Internet all'host bastione e al database, quindi dovrai consentire al traffico di uscire dal VPC. Per fare ciò, Amazon VPC collega un gateway Internet al gateway Internet e aggiunge un routing alla tabella di routing in modo che il traffico destinato al di fuori del VPC venga inviato al gateway Internet.

All'interno del VPC, crei un endpoint Secrets Manager e un endpoint Amazon RDS. Quando si imposta la rotazione automatica in una fase successiva, Secrets Manager crea la funzione di rotazione Lambda all'interno del VPC in modo che abbia accesso al database. La funzione di rotazione Lambda chiama anche Secrets Manager per aggiornare il segreto e chiama Amazon RDS per ottenere le informazioni di connessione al database. Creando un endpoint all'interno del VPC, ci si assicura che le chiamate dalla funzione Lambda a Secrets Manager e Amazon RDS non escano dall'infrastruttura AWS. Invece, vengono instratade all'endpoint all'interno del VPC.

Per creare un VPC

1. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Seleziona Crea VPC.
3. Nella pagina Create VPC (Crea VPC), scegli VPC and more (VPC e altro).

4. In Name tag auto-generation (Generazione automatica del tag del nome), sotto a Auto-generate (genera automaticazione), inserisci **`SecretsManagerTutorial`**.
5. Per le opzioni DNS, scegli entrambi **`Enable DNS hostnames`** e **`Enable DNS resolution`**.
6. Seleziona Crea VPC.

Creare un endpoint di Secrets Manager all'interno del VPC

1. Nella console Amazon VPC, sotto Endpoints, scegli Create endpoint (crea endpoint).
2. In Endpoint settings (Impostazioni endpoint), per Name (Nome) inserisci **`SecretsManagerTutorialEndpoint`**.
3. Sotto a Services (Servizi), inserisci **`secretsmanager`** per filtrare l'elenco e quindi seleziona l'endpoint di Secrets Manager nella Regione AWS. Per esempio, negli Stati Uniti orientali (Virginia settentrionale), scegli `com.amazonaws.us-east-1.secretsmanager`.
4. Per VPC, scegli **`vpc**** (SecretsManagerTutorial)`**.
5. Per Subnets (Sottoreti), seleziona tutte le Availability Zones (Zone di disponibilità) e poi per ognuna di esse scegli un Subnet ID (ID sottorete) da includere.
6. Per Tipo di indirizzo IP, scegli **`IPv4`**.
7. Da Security Groups (Gruppi di sicurezza), scegli il gruppo di sicurezza di default.
8. Per Policy type (Tipo di policy), scegli **`Full access`**.
9. Seleziona Create endpoint (Crea endpoint).

Creare un endpoint di Amazon RDS all'interno del VPC

1. Nella console Amazon VPC, sotto Endpoints, scegli Create endpoint (crea endpoint).
2. In Endpoint settings (Impostazioni endpoint), per Name (Nome) inserisci **`RDSTutorialEndpoint`**.
3. Sotto a Services (Servizi), inserisci **`rds`** per filtrare l'elenco e quindi seleziona l'endpoint di Amazon RDS in Regione AWS. Per esempio, negli Stati Uniti orientali (Virginia settentrionale), scegli `com.amazonaws.us-east-1.rds`.
4. Per VPC, scegli **`vpc**** (SecretsManagerTutorial)`**.
5. Per Subnets (Sottoreti), seleziona tutte le Availability Zones (Zone di disponibilità) e poi per ognuna di esse scegli un Subnet ID (ID sottorete) da includere.
6. Per Tipo di indirizzo IP, scegli **`IPv4`**.

7. Da Security Groups (Gruppi di sicurezza), scegli il gruppo di sicurezza di default.
8. Per Policy type (Tipo di policy), scegli **`Full access`**.
9. Seleziona Create endpoint (Crea endpoint).

Prerequisito B: istanza Amazon EC2

Il database Amazon RDS che creerai in una fase successiva sarà nel cloud VPC, per accedervi è necessario un host bastione. L'host bastione è presente anche nel VPC, ma in una fase successiva, configurerai un gruppo di sicurezza per consentire al computer locale di connettersi all'host bastione con SSH.

Creare un'istanza EC2 per un host bastione

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Scegliere Instances (Istanze), quindi scegliere Launch Instances (Avvia istanze).
3. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`SecretsManagerTutorialInstance`**.
4. In Application and OS Images (Immagini dell'applicazione e del sistema operativo), mantieni l'impostazione predefinita **`Amazon Linux 2 AMI (HMV) Kernel 5.10`**.
5. In Instance type (Tipo di istanza), mantieni l'impostazione predefinita **`t2.micro`**.
6. In Key pair (coppia di chiavi), scegli Crea coppia di chiavi.

   Nella finestra di dialogo Create Key Pair (Crea coppia di chiavi), per Key pair name (Nome coppia di chiavi), inserisci **`SecretsManagerTutorialKeyPair`**, quindi scegli Create (Crea).

   La chiave privata viene scaricata automaticamente.
7. In Network settings (Impostazioni di rete), scegli Edit (Modifica) ed esegui le operazioni qui descritte:

   a. Per VPC, scegliere **`vpc-**** SecretsManagerTutorial`**.

   b. Per Auto-assign Public IP (Assegna automaticamente IP pubblico), scegliere **`Enable`**.

   c. Per Firewall, scegli Select existing security group (seleziona gruppo di sicurezza esistente).

   d. Per i gruppi di sicurezza comuni, scegli **`default`**.
8. Scegliere Launch Instance (Avvia istanza).

Prerequisito C: database Amazon RDS e un segreto in Secrets Manager per le credenziali di amministratore

In questa fase, crei un database Amazon MySQL e lo configuri in modo che Amazon RDS crei un segreto per contenere le credenziali di amministratore. Quindi Amazon RDS gestisce automaticamente la rotazione del segreto di amministratore per te. Per ulteriori informazioni, consulta Rotazione gestita.

Durante la creazione del database, è necessario specificare l'host bastione creato nel passaggio precedente. Quindi Amazon RDS configura i gruppi di sicurezza in modo che il database e l'istanza possano accedere vicendevolmente. Si aggiunge una regola al gruppo di sicurezza collegato all'istanza per consentire anche al computer locale di connettersi ad essa.

Per creare un database Amazon RDS con un segreto di Secrets Manager contenente le credenziali di amministratore

1. Nella console Amazon RDS scegli Create databases (Crea database).
2. Nella sezione Engine options (Opzioni motore) per Engine type (tipo motore) scegli **`MySQL`**.
3. Nella sezione Templates (Modelli), seleziona **`Free tier`**.
4. Nella sezione Rule settings (Impostazioni regole), procedi nel seguente modo:
    a. Per l'identificatore dell'istanza DB, inserisci **`SecretsManagerTutorial`**.
    b. In Impostazioni credenziali, seleziona Gestisci le credenziali principali in AWS Secrets Manager.
5. Nella sezione Connectivity (Connettività), per Computer resource (Risorsa computer), scegli Connect to an EC2 computer resource (Connetti a una risorsa computer EC2) e quindi, per l'istanza EC2, scegli **`SecretsManagerTutorialInstance`**.
6. Scegliere Crea database.

Prerequisito D: consenti al computer locale di connettersi all'istanza EC2.

In questo passaggio, configuri l'istanza EC2 creata nel Prerequisito B per consentire al computer locale di connettersi ad essa. A tale scopo, modifichi il gruppo di sicurezza aggiunto da Amazon RDS in Prereq C per includere una regola che consenta all'indirizzo IP del tuo computer di connettersi a SSH. La regola consente al computer locale (identificato dal tuo attuale indirizzo IP) di connettersi all'host bastione utilizzando SSH su Internet.

Prerequisito D: consenti al computer locale di connettersi all'istanza EC2.

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'istanza EC2 SecretsManagerTutorialInstance, nella scheda Security (Sicurezza), in Security groups (Gruppi di sicurezza), scegli **`sg-*** (ec2-rds-X)`**.
3. Sotto a Inbound rules (Regole in entrata), seleziona Edit inbound rules (Modifica regole in entrata).
4. Scegli Add rule (Aggiungi regola), poi per la regola esegui le seguenti operazioni:
   a. In Type (Tipo) scegliere **`SSH`**.
   b. Per Source (origine), scegli **`My IP`**.

Fase 1: creazione di un utente del database Amazon RDS

Innanzitutto, è necessario un utente le cui credenziali saranno memorizzate nel segreto. Per creare l'utente, accedi al database Amazon RDS con le credenziali di amministratore. Per semplicità, nel tutorial, si crea un utente con piena autorizzazione per un database. In un ambiente di produzione questo non succede normalmente e ti consigliamo di seguire il principio del privilegio minimo.

Per connettersi al database, si utilizza uno strumento client MySQL. In questa esercitazione, sarà possibile usare MySQL Workbench, un'applicazione basata su GUI. Scaricare e installare MySQL Workbench dalla pagina di Download MySQL Workbench (Scarica MySQL Workbench).

Per connettersi al database, è necessario creare una configurazione di connessione in MySQL Workbench. Per la configurazione, sono necessarie alcune informazioni sia da Amazon EC2 che Amazon RDS.

Creare una connessione al database in MySQL Workbench

1. In MySQL Workbench, accanto a MySQL Connections (Connessioni MySQL), scegli il pulsante (+).
2. Nella finestra di dialogo Setup New Connection (Configura una nuova connessione), segui questi passaggi:
   a. Per Connection Name (Nome connessione), inserisci **`SecretsManagerTutorial`**.
   b. Per Connection Method (Metodo di connessione), scegli **`Standard TCP/IP over SSH`**.
   c. Nella scheda Parameters (Parametri), procedi come segue:

i. Per Hostname SSH (Nome host SSH), inserisci l'indirizzo IP pubblico dell'istanza Amazon EC2.

   Puoi trovare l'indirizzo IP sulla console Amazon EC2 scegliendo l'istanza SecretsManagerTutorialInstance. Copia l'indirizzo IP in Public IPv4 DNS (DNS IPv4 pubblico).

ii. Per SSH Username (Nome utente SSH), inserisci **`ec2-user`**.

iii. Per SSH Keyfile (Keyfile SSH), scegli il file della coppia di chiavi SecretsManagerTutorialKeyPair.pem scaricato nel prerequisito precedente.

iv. Per MySQL Hostname (Home host MySQL), inserisci l'indirizzo dell'endpoint Amazon RDS.

   Puoi trovare l'indirizzo endpoint sulla console Amazon RDS scegliendo l'istanza di database secretsmanagertutorialdb. Copia l'indirizzo in Endpoint.

v. Per Username (Nome utente), inserisci **`admin`**.

d. Scegli OK.

Recuperare la password dell'amministratore

1. Nella console Amazon RDS scegli il database.
2. Nella scheda Configuration (Configurazione), in Master Credentials ARN (ARN delle credenziali principali), scegli Manage in Secrets Manager (gestisci in secrets manager).

   Si apre la console Secrets Manager.
3. Nella pagina dei dettagli del segreto, scegli Retrieve secret value (Recupera il valore di un segreto).
4. La password viene visualizzata nella sezione Secret value (valore segreto).

Per creare un utente del database

1. In MySQL Workbench, scegli la connessione SecretsManagerTutorial.
2. Inserisci la password dell'amministratore che hai recuperato dal segreto.
3. In MySQL Workbench, nella finestra Query, inserisci i seguenti comandi (inclusa una password sicura) e quindi scegli Execute (Esegui).

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

Nella finestra Output, viene visualizzato l'esito positivo dei comandi.

Fase 2: creazione di un segreto per le credenziali dell'utente

Successivamente, creerai un segreto per archiviare le credenziali dell'utente appena creato. Questo è il segreto che verrà ruotato. Attiva la rotazione automatica e, per indicare la strategia a utenti alternati, scegli un segreto di un utente con privilegi avanzati separato che dispone dell'autorizzazione per modificare la password del primo utente.

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:
   a. Per Secret type (Tipo segreto), scegli Credentials for Amazon RDS database (Credenziali per il database Amazon RDS).
   b. Per Credentials (Credenziali), inserisci il nome utente **`appuser`** e la password inseritaper l'utente del database creato utilizzando MySQL Workbench.
   c. Per Database, scegli secretsmanagertutorialdb.
   d. Seleziona Successivo.
4. Nella pagina Configure secret (Configura il segreto), per Secret name (Nome del segreto), inserisci **`SecretsManagerTutorialAppuser`** e scegli Next (Successivo).
5. Nella pagina Configure rotation (Configura la rotazione), effetua le seguenti operazioni:
   a. Attiva Automatic rotation (Rotazione automatica).
   b. Per Rotation schedule (Pianificazione della rotazione), imposta una pianificazione di Days (Giorni): **2** Days with Duration: **2h** (2 giorni con durata: 2 h). Mantieni selezionato Rotate immediately (Ruota immediatamente).
   c. Per Rotation function (Funzione di rotazione), scegli Create a rotation function (Crea una funzione di rotazione) e per il nome della funzione inserisci **`tutorial-alternating-users-rotation`**.

d. Per Strategia di rotazione, scegli Utenti alternati, quindi in Segreto credenziali amministratore scegli il segreto denominato rds!cluster... che ha una Descrizione che include il nome del database creato in questo tutorial **`secretsmanagertutorial`**, ad esempio `Secret associated with primary RDS DB instance: arn:aws:rds:`*`Region`*`:`*`AccountId`*`:db:secretsmanagertutorial`.

e. Seleziona Successivo.

6. Nella pagina Review (Rivedi), scegli Store (Archivia).

   Secrets Manager torna alla pagina dei dettagli del segreto. Nella parte superiore della pagina, è possibile visualizzare lo stato della configurazione di rotazione. Secrets Manager utilizza CloudFormation per creare risorse come la funzione di rotazione Lambda e un ruolo di esecuzione che esegue la funzione Lambda. Quando CloudFormation termina l'operazione, il banner cambia in Secret scheduled for rotation (Segreto pianificato per la rotazione). La prima rotazione è completa.

Fase 3: eseguire il test del segreto ruotato

Ora che il segreto è stato ruotato, è possibile verificare che il segreto contenga ancora credenziali valide. La password nel segreto è cambiata rispetto alle credenziali originali.

Recuperare la nuova password dal segreto

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Secrets (Segreti) e quindi scegli il segreto **`SecretsManagerTutorialAppuser`**.
3. Nella pagina Secret details (Dettagli del segreto), scorri e scegli Retrieve secret value (Recupera il valore del segreto).
4. Nella tabella Key/value (Chiave/valore), copia il Secret value (Valore segreto) per la **`password`**.

Testare le credenziali

1. In MySQL Workbench, fai clic con il pulsante destro del mouse sulla connessione SecretsManagerTutorial, quindi scegli Edit Connection (Modifica connessione).
2. Nella finestra di dialogo Manage Server Connections (Gestisci connessioni al server), per Username (Nome utente), inserisci **`appuser`** e scegli Close (Chiudi).
3. In MySQL Workbench, scegli la connessione SecretsManagerTutorial.

4. Nella finestra di dialogo Open SSH Connection (Apri connessione SSH), per Password incolla la password recuperata dal segreto, quindi scegli OK.

   Se le credenziali sono valide, MySQL Workbench si apre alla pagina di progettazione del database.

Questa mostra che la rotazione del segreto ha esito positivo. Le credenziali nel segreto sono state aggiornate e consistono in una password valida per connettersi al database.

Fase 4: eliminazione delle risorse

Per provare un'altra strategia di rotazione, la rotazione a utente singolo, salta la pulizia delle risorse e vai a the section called “Rotazione a utente singolo”.

Per evitare potenziali addebiti e per rimuovere l'istanza EC2 che ha accesso a Internet, elimina le seguenti risorse create in questo tutorial e i relativi prerequisiti:

- Istanza database Amazon RDS Per istruzioni, consulta Eliminazione di un'istanza DB nella Guida per l'utente di Amazon RDS.
- Istanza Amazon EC2. Per istruzioni, consulta Terminazione di un'istanza nella Guida per l'utente di Amazon EC2 per le istanze Linux.
- Segreto di Secrets Manager `SecretsManagerTutorialAppuser`. Per istruzioni, consultare the section called “Eliminare un segreto”.
- Endpoint di Secrets Manager. Per istruzioni, consulta Eliminazione di un endpoint VPC nella Guida AWS PrivateLink.
- Endpoint VPC. Per istruzioni, consulta Eliminazione di un VPC nella Guida AWS PrivateLink.

Fasi successive

- Ottieni informazioni su come recuperare i segreti nelle applicazioni.
- Scopri altri programmi di rotazione.

Configurazione di una rotazione a utente singolo per AWS Secrets Manager

In questo tutorial, imparerai come impostare la rotazione a utente singolo per un segreto che contiene le credenziali del database. La rotazione a utente singolo è una strategia di rotazione in cui Secrets Manager aggiorna le credenziali di un utente sia nel segreto che nel database. Per ulteriori informazioni, consulta the section called “Utente singolo”.

Al termine del tutorial, si consiglia di ripulire le risorse del tutorial. Non utilizzarle in un ambiente di produzione.

La rotazione di Secrets Manager utilizza una funzione AWS Lambda per aggiornare il segreto e il database. Per ulteriori informazioni sui costi di utilizzo della funzione Lambda, consulta la sezione Prezzi.

Indice

- Autorizzazioni
- Prerequisiti
- Fase 1: creazione di un utente del database Amazon RDS
- Fase 2: creazione di un segreto per le credenziali utente del database
- Fase 3: esecuzione del test della password ruotata
- Fase 4: eliminazione delle risorse
- Fasi successive

Autorizzazioni

Per i prerequisiti del tutorial, hai bisogno di autorizzazioni di amministrazione per il tuo Account AWS. In un ambiente di produzione, è consigliabile utilizzare ruoli diversi per ciascun passaggio. Ad esempio, un ruolo con le autorizzazioni dell'amministratore del database creerebbe il database Amazon RDS e un ruolo con autorizzazioni di amministrazione di rete configurerebbe il VPC e i gruppi di sicurezza. Per i passaggi del tutorial, suggeriamo di continuare a utilizzare la stessa identità.

Per informazioni su come configurare le autorizzazioni in un ambiente di produzione, consulta Autenticazione e controllo degli accessi.

Prerequisiti

Il prerequisito per questo tutorial è the section called “Rotazione a utenti alternati”. Non eliminare le risorse alla fine del primo tutorial. Dopo questo tutorial, disporrai di un ambiente realistico con un database Amazon RDS e un segreto di Secrets Manager che contiene le credenziali di amministratore per il database. Hai anche un secondo segreto che contiene le credenziali per un utente del database, ma non lo usi in questo tutorial.

È inoltre configurata una connessione in MySQL Workbench per connettersi al database con le credenziali dell'amministratore.

Fase 1: creazione di un utente del database Amazon RDS

Innanzitutto, è necessario un utente le cui credenziali saranno memorizzate nel segreto. Per creare l'utente, accedi al database Amazon RDS con le credenziali di amministratore archiviate in un segreto. Per semplicità, nel tutorial, si crea un utente con piena autorizzazione per un database. In un ambiente di produzione questo non succede normalmente e ti consigliamo di seguire il principio del privilegio minimo.

Recuperare la password dell'amministratore

1. Nella console Amazon RDS scegli il database.
2. Nella scheda Configuration (Configurazione), in Master Credentials ARN (ARN delle credenziali principali), scegli Manage in Secrets Manager (gestisci in secrets manager).

   Si apre la console Secrets Manager.
3. Nella pagina dei dettagli del segreto, scegli Retrieve secret value (Recupera il valore di un segreto).
4. La password viene visualizzata nella sezione Secret value (valore segreto).

Per creare un utente del database

1. In MySQL Workbench, fai clic con il pulsante destro del mouse sulla connessione SecretsManagerTutorial, quindi scegli Edit Connection (Modifica connessione).
2. Nella finestra di dialogo Manage Server Connections (Gestisci connessioni al server), per Username (Nome utente), inserisci **`admin`** e scegli Close (Chiudi).
3. In MySQL Workbench, scegli la connessione SecretsManagerTutorial.

4. Inserisci la password dell'amministratore che hai recuperato dal segreto.
5. In MySQL Workbench, nella finestra Query, inserisci i seguenti comandi (inclusa una password sicura) e quindi scegli Execute (Esegui).

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

   Nella finestra Output, viene visualizzato l'esito positivo dei comandi.

Fase 2: creazione di un segreto per le credenziali utente del database

Successivamente, creerai un segreto per archiviare le credenziali dell'utente appena creato e attiverai la rotazione automatica, inclusa una rotazione immediata. Secrets Manager ruota il segreto, il che significa che la password viene generata programmaticamente: nessun essere umano ha visto questa nuova password. L'avvio immediato della rotazione può anche aiutarti a determinare se la rotazione è impostata correttamente.

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:
   a. Per Secret type (Tipo segreto), scegli Credentials for Amazon RDS database (Credenziali per il database Amazon RDS).
   b. Per Credentials (Credenziali), inserisci il nome utente **`dbuser`** e la password inseritaper l'utente del database creato utilizzando MySQL Workbench.
   c. Per Database, scegli secretsmanagertutorialdb.
   d. Seleziona Successivo.
4. Nella pagina Configure secret (Configura il segreto), per Secret name (Nome del segreto), inserisci **`SecretsManagerTutorialDbuser`** e scegli Next (Successivo).
5. Nella pagina Configure rotation (Configura la rotazione), effetua le seguenti operazioni:
   a. Attiva Automatic rotation (Rotazione automatica).
   b. Per Rotation schedule (Pianificazione della rotazione), imposta una pianificazione di Days (Giorni): **2** Days with Duration: **2h** (2 giorni con durata: 2 h). Mantieni selezionato Rotate immediately (Ruota immediatamente).

c. Per Rotation function (Funzione di rotazione), scegli Create a rotation function (Crea una funzione di rotazione) e per il nome della funzione inserisci **`tutorial-single-user-rotation`**.

d. Per Strategia di rotazione scegli Utente singolo.

e. Seleziona Successivo.

6. Nella pagina Review (Rivedi), scegli Store (Archivia).

   Secrets Manager torna alla pagina dei dettagli del segreto. Nella parte superiore della pagina, è possibile visualizzare lo stato della configurazione di rotazione. Secrets Manager utilizza CloudFormation per creare risorse come la funzione di rotazione Lambda e un ruolo di esecuzione che esegue la funzione Lambda. Quando CloudFormation termina l'operazione, il banner cambia in Secret scheduled for rotation (Segreto pianificato per la rotazione). La prima rotazione è completa.

Fase 3: esecuzione del test della password ruotata

Dopo la prima rotazione segreta, che potrebbe richiedere alcuni secondi, è possibile verificare che il segreto contenga ancora credenziali valide. La password nel segreto è cambiata rispetto alle credenziali originali.

Recuperare la nuova password dal segreto

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Secrets (Segreti) e quindi scegli il segreto **`SecretsManagerTutorialDbuser`**.
3. Nella pagina Secret details (Dettagli del segreto), scorri e scegli Retrieve secret value (Recupera il valore del segreto).
4. Nella tabella Key/value (Chiave/valore), copia il Secret value (Valore segreto) per la **`password`**.

Testare le credenziali

1. In MySQL Workbench, fai clic con il pulsante destro del mouse sulla connessione SecretsManagerTutorial, quindi scegli Edit Connection (Modifica connessione).
2. Nella finestra di dialogo Manage Server Connections (Gestisci connessioni al server), per Username (Nome utente), inserisci **`dbuser`** e scegli Close (Chiudi).

3. In MySQL Workbench, scegli la connessione SecretsManagerTutorial.
4. Nella finestra di dialogo Open SSH Connection (Apri connessione SSH), per Password incolla la password recuperata dal segreto, quindi scegli OK.

   Se le credenziali sono valide, MySQL Workbench si apre alla pagina di progettazione del database.

Fase 4: eliminazione delle risorse

Per evitare potenziali addebiti, elimina il segreto creato in questo tutorial. Per istruzioni, consultare the section called “Eliminare un segreto”.

Per ripulire le risorse create nel tutorial precedente, consulta the section called “Fase 4: eliminazione delle risorse”.

Fasi successive

- Ottieni informazioni su come recuperare i segreti nelle applicazioni. Per informazioni, consultare Recuperare segreti.
- Scopri altri programmi di rotazione. Per informazioni, consultare the section called “Espressioni di pianificazione”.

Autenticazione e controllo degli accessi per AWS Secrets Manager

Utilizza Secrets ManagerAWS Identity and Access Management (IAM) per proteggere l'accesso ai segreti. IAM fornisce autenticazione e controllo degli accessi. Autenticazione verifica l'identità di coloro che effettuano le richieste. Secrets Manager utilizza un processo di accesso con le password, le chiavi di accesso e l'autenticazione a più fattori (MFA) per verificare l'identità degli utenti. Consulta Accesso aAWS. Autorizzazione assicura che solo gli individui approvati possano eseguire operazioni sulle risorse AWS ad esempio sui segreti. Secrets Manager utilizza le policy per definire chi ha accesso a quali risorse e quali azioni l'identità può intraprendere su tali risorse. Vedere Autorizzazioni e policy in IAM.

Puoi utilizzare AWS Identity and Access Management Roles Anywhere per ottenere credenziali di sicurezza temporanee in IAM per carichi di lavoro come server, container e applicazioni che vengono eseguiti all'esterno di AWS. I tuoi carichi di lavoro possono utilizzare le stesse policy IAM e gli stessi ruoli IAM che utilizzi con le applicazioni AWS per accedere alle risorse AWS. Con IAM Roles Anywhere, puoi utilizzare Secrets Manager per archiviare e gestire le credenziali alle quali possono accedere sia le risorse in AWS sia i dispositivi on-premise, come i server delle applicazioni. Per ulteriori informazioni, consulta la Guida per l'utente di IAM Roles Anywhere.

Autorizzazioni di amministrazione di Secrets Manager

Per concedere le autorizzazioni di amministratore di Secrets Manager, seguire le istruzioni riportate in Aggiunta e rimozione di autorizzazioni per identità IAM, e allegare i seguenti criteri:

- SecretsManagerReadWrite
- IAMFullAccess

Si consiglia di non concedere autorizzazioni di amministratore agli utenti finali. Sebbene ciò consente agli utenti di creare e gestire i propri segreti, l'autorizzazione necessaria per abilitare la rotazione (IAMFullAccess) concede autorizzazioni significative che non sono appropriate per gli utenti finali.

Autorizzazioni per accedere ai segreti

Utilizzando le policy di autorizzazione IAM, puoi controllare quali utenti o servizi possono accedere ai segreti. Una policy di autorizzazioni descrive chi può eseguire quali azioni su quali risorse. È possibile:

- the section called “Allegare un policy di autorizzazione a un'identità”
- the section called “Allegare una policy di autorizzazione a un segreto”

Autorizzazioni per le funzioni di rotazione Lambda

Secrets Manager utilizza AWS Lambda funzioni per ruotare i segreti. La funzione Lambda deve avere accesso al segreto e al database o al servizio per cui il segreto contiene le credenziali. Per informazioni, consultare Autorizzazioni per la rotazione.

Autorizzazioni per le chiavi di crittografia

Secrets Manager utilizza AWS Key Management Service (AWS KMS) chiavi per crittografia dei segreti. La Chiave gestita da AWS `aws/secretsmanager` ha automaticamente le autorizzazioni corrette. Se si utilizza una chiave KMS diversa, Secrets Manager necessita delle autorizzazioni per tale chiave. Per informazioni, consultare the section called “Autorizzazioni per la chiave KMS”.

Allegare un policy di autorizzazione a un'identità

Come allegare policy di autorizzazioni a identità, utenti, gruppi, ruoli, servizi e risorse IAM. In una policy basata sull'identità, specifichi a quali segreti può accedere l'identità e le azioni che l'identità può eseguire sui segreti. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM.

Puoi concedere autorizzazioni a un ruolo che rappresenta un'applicazione o un utente in un altro servizio. Ad esempio, un'applicazione in esecuzione su un'istanza Amazon EC2 potrebbe dover accedere a un database. È possibile creare un ruolo IAM associato al profilo dell'istanza EC2 e quindi utilizzare una policy di autorizzazioni per concedere al ruolo l'accesso al segreto che contiene le credenziali per il database. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2. Altri servizi a cui è possibile allegare i ruoli includono Amazon Redshift, AWS Lambda ed Amazon ECS.

Puoi concedere autorizzazioni agli utenti autenticati da un sistema di identità diverso da IAM. Ad esempio, è possibile associare ruoli IAM; a utenti che utilizzano app per dispositivi mobili e che effettuano l'accesso con Amazon Cognito. Il ruolo concede all'app le credenziali provvisorie con le autorizzazioni nella policy di autorizzazioni del ruolo. È quindi possibile utilizzare una policy di autorizzazione per concedere al ruolo l'accesso al segreto. Per ulteriori informazioni, consulta Provider di identità e federazione.

Si possono utilizzare policy basate su identità per:

- Concedi un accesso alle identità a più segreti.
- Controllare chi può creare nuovi segreti e chi può accedere a segreti che non sono ancora stati creati.
- Concedi un accesso a un gruppo IAM ai segreti.

Per ulteriori informazioni, consulta the section called “Esempi di policy di autorizzazione”.

Allegare una policy di autorizzazione a un segreto AWS Secrets Manager

In una policy basata sulle risorse, si specifica chi può accedere al segreto e le azioni che è possibile eseguire sul segreto. Le policy basate su risorse possono essere utilizzate per:

- Concedre l'accesso a più utenti o ruoli ad un singolo segreto.
- Concedi l'accesso a utenti o ruoli in altri AWS account.

Per informazioni, consulta the section called “Esempi di policy di autorizzazione”.

Quando si allega una policy basata sulle risorse a un segreto nella console, Secrets Manager utilizza il motore di ragionamento automatico Zelkova e l'API `ValidateResourcePolicy` per impedirti di concedere a una vasta gamma di entità IAM l'accesso ai tuoi segreti. In alternativa, è possibile chiamare `PutResourcePolicy` API con `BlockPublicPolicy` parametro da CLI o SDK.

> **Important**
>
> La convalida della politica delle risorse e del `BlockPublicPolicy` parametro aiutano a proteggere le risorse impedendo che l'accesso pubblico venga concesso attraverso le politiche relative alle risorse direttamente collegate ai segreti dell'utente. Oltre a utilizzare

queste funzionalità, esamina attentamente le seguenti politiche per verificare che non garantiscano l'accesso pubblico:

- Politiche basate sull'identità collegate ai AWS principali associati (ad esempio, ruoli IAM)
- Politiche basate sulle risorse collegate alle AWS risorse associate (ad esempio, () chiavi) AWS Key Management Service AWS KMS

Per rivedere le autorizzazioni relative ai tuoi segreti, consulta. Determinazione di chi ha le autorizzazioni per i segreti

Come visualizzare, modificare o eliminare la policy di risorse per un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Dall'elenco dei segreti, scegli il segreto.
3. Nella pagina dei dettagli del segreto, nella sezione Panoramica, nella sezione Autorizzazioni risorse, scegli Modifica autorizzazioni.
4. Nel campo Codice, eseguire una delle operazioni seguenti, quindi scegliere Save (Salva):
   - Per allegare o modificare una policy delle risorse, immettere la policy.
   - Per eliminare la policy, deselezionare il campo del codice.

AWS CLI

Example Recupero di una politica sulle risorse

L'esempio di `get-resource-policy` seguente mostra come recuperare la policy basata su risorse collegata a un segreto.

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

Example Eliminazione di una policy sulle risorse

L'esempio di `delete-resource-policy` seguente mostra come eliminare la policy basata su risorse collegata a un segreto.

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

Example Aggiunta di una policy sulle risorse

L'esempio di `put-resource-policy` seguente mostra come aggiungere una policy di autorizzazioni a un segreto, verificando innanzitutto che la policy non fornisca un accesso ampio al segreto. La policy viene letta da un file. Per ulteriori informazioni, consulta Caricamento AWS CLI dei parametri da un file nella Guida per l' AWS CLI utente.

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

Contenuto di `mypolicy.json`.

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

AWS SDK

Per recuperare la policy collegata a un segreto, utilizzare `GetResourcePolicy`.

Per eliminare una policy collegata a un segreto, utilizzare `DeleteResourcePolicy`.

Per collegare una policy a un segreto, utilizzare `PutResourcePolicy`. Se è già associata una policy, il comando la sostituisce con la nuova policy. La policy deve essere formattata come testo strutturato JSON. Vedere Struttura dei documenti di policy JSON. Utilizzo dell' the section called “Esempi di policy di autorizzazione” per iniziare a scrivere la tua policy.

Per ulteriori informazioni, consulta the section called “AWS SDK”.

AWS politica gestita per AWS Secrets Manager

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: SecretsManagerReadWrite

Questa policy fornisce l'accesso in lettura/scrittura AWS Secrets Manager, inclusa l'autorizzazione a descrivere, le risorse Amazon RDS, Amazon Redshift e Amazon DocumentDB e l'autorizzazione all'uso per crittografare e decrittografare i segreti. AWS KMS Questa policy consente inoltre di creare set di AWS CloudFormation modifiche, ottenere modelli di rotazione da un bucket Amazon S3 gestito da AWS, elencare AWS Lambda funzioni e descrivere i VPC Amazon EC2. Queste autorizzazioni sono richieste dalla console per impostare la rotazione con le funzioni di rotazione esistenti.

Per creare nuove funzioni di rotazione, devi inoltre disporre dell'autorizzazione a creare AWS CloudFormation stack e ruoli di esecuzione. AWS Lambda Puoi assegnare la policy FullAccess gestita da IAM. Per informazioni, consulta Autorizzazioni per la rotazione.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `secretsmanager`: consente ai principali di eseguire tutte le azioni di Secrets Manager.

- `cloudformation`— Consente ai principali di creare AWS CloudFormation stack. Ciò è necessario affinché i principali che utilizzano la console per attivare la rotazione possano creare funzioni AWS CloudFormation di rotazione Lambda tramite pile. Per ulteriori informazioni, consulta the section called “Come Secrets Manager utilizza AWS CloudFormation”.
- `ec2`: consente ai principali di descrivere i VPC Amazon EC2. Ciò è necessario affinché i principali che utilizzano la console possano creare funzioni di rotazione nello stesso VPC del database delle credenziali che stanno archiviando in un segreto.
- `kms`— Consente ai principali di utilizzare le AWS KMS chiavi per le operazioni crittografiche. Ciò è necessario per consentire a Secrets Manager di crittografare e decrittografare i segreti. Per ulteriori informazioni, consulta the section called “Crittografia e decrittografia del segreto”.
- `lambda`: consente ai principali di elencare le funzioni di rotazione Lambda. Ciò è necessario affinché i principali che utilizzano la console possano scegliere le funzioni di rotazione esistenti.
- `rds`: consente ai principali di descrivere i cluster e le istanze in Amazon RDS. Ciò è necessario affinché i principali che utilizzano la console possano scegliere cluster o istanze Amazon RDS.
- `redshift`: consente ai principali di descrivere i cluster in Amazon Redshift. Ciò è necessario affinché i principali che utilizzano la console possano scegliere cluster Amazon Redshift.
- `redshift-serverless`— Consente ai responsabili di descrivere i namespace in Amazon Redshift Serverless. Ciò è necessario affinché i responsabili che utilizzano la console possano scegliere i namespace Serverless di Amazon Redshift.
- `docdb-elastic`: consente ai principali di descrivere cluster elastici in Amazon DocumentDB. Ciò è necessario affinché i principali che utilizzano la console possano scegliere cluster elastici Amazon DocumentDB.
- `tag`: consente ai principali di ottenere tutte le risorse dell'account che sono contrassegnate.
- `serverlessrepo`— Consente ai principali di creare set di modifiche. AWS CloudFormation Ciò è necessario affinché i principali che utilizzano la console possano creare funzioni di rotazione Lambda. Per ulteriori informazioni, consulta the section called “Come Secrets Manager utilizza AWS CloudFormation”.
- `s3`— Consente ai principali di ottenere oggetti da un bucket Amazon S3 gestito da. AWS Questo bucket contiene Lambda Modelli di funzione di rotazione. Questo permesso è necessario affinché i principali che utilizzano la console possano creare funzioni di rotazione Lambda basate sui modelli nel bucket. Per ulteriori informazioni, consulta the section called “Come Secrets Manager utilizza AWS CloudFormation”.

Per visualizzare la policy, consulta il documento sulla policy SecretsManagerReadWrite JSON.

Secrets Manager: aggiornamenti alle policy AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Secrets Manager.

Modifica	Descrizione	Data
SecretsManagerReadWrite: aggiornamento a una policy esistente	Questa policy è stata aggiornata per consentire l'accesso descrittivo ad Amazon Redshift Serverless in modo che gli utenti della console possano scegliere uno spazio dei nomi Amazon Redshift Serverless quando creano un segreto Amazon Redshift.	12 marzo 2024
SecretsManagerReadWrite: aggiornamento a una policy esistente	Questa policy è stata aggiornata per consentire l'accesso descrittivo ai cluster elastici di Amazon DocumentDB in modo che gli utenti della console possano scegliere un cluster elastico quando creano un segreto Amazon DocumentDB.	12 settembre 2023
SecretsManagerReadWrite: aggiornamento a una policy esistente	Questa policy è stata aggiornata per consentire l'accesso descrittivo ad Amazon Redshift in modo che gli utenti della console possano scegliere un cluster Amazon Redshift quando creano un segreto Amazon Redshift. L'aggiornamento ha inoltre aggiunto nuove	24 giugno 2020

Modifica	Descrizione	Data
	autorizzazioni per consentire l'accesso in lettura a un bucket Amazon S3 gestito AWS da che memorizza i modelli delle funzioni di rotazione Lambda.	
SecretsManagerReadWrite: aggiornamento a una policy esistente	Questa policy è stata aggiornata per consentire l'accesso descrittivo ai cluster Amazon RDS in modo che gli utenti della console possano scegliere un cluster quando creano un segreto Amazon RDS.	3 maggio 2018
SecretsManagerReadWrite: nuova policy	Secrets Manager ha creato una policy per concedere le autorizzazioni necessarie per utilizzare la console con tutti gli accessi in lettura/scrittura a Secrets Manager.	04 Aprile 2018
Secrets Manager ha iniziato a tenere traccia delle modifiche	Secrets Manager ha iniziato a tenere traccia delle modifiche per le politiche AWS gestite.	04 Aprile 2018

Determinazione di chi ha le autorizzazioni per i segreti AWS Secrets Manager

Per impostazione predefinita, le identità IAM non dispongono dell'autorizzazione per accedere ai segreti. Quando si autorizza l'accesso a un segreto, Secret Manager valuta la policy basata su risorse collegata al segreto e tutte le policy basato sull'identità collegate all'utente IAM o al ruolo da cui proviene la richiesta. Per eseguire questa operazione, Secret Manager utilizza un processo simile a quello descritto in Determinare se una richiesta è consentita o rifiutata nella Guida per l'utente IAM.

Quando a una richiesta si applicano varie policy, Gestione dei segreti utilizza una gerarchia per controllare le autorizzazioni:

1. Se una dichiarazione in qualsiasi politica con un esplicito `deny` corrisponde all'azione della richiesta e alla risorsa:

   L'esplicito `deny` sovrascrive tutto il resto e blocca l'azione.
2. Se non c'è esplicito `deny`, ma una dichiarazione con un esplicito `allow` corrisponde all'azione della richiesta e alla risorsa:

   L'esplicito `allow` concede l'azione nella richiesta di accesso alle risorse nell'istruzione.

   Se l'identità e il segreto si trovano in due account diversi, deve esserci un `allow` sia nella politica delle risorse per il segreto che nella politica associata all'identità, altrimenti AWS rifiuta la richiesta. Per ulteriori informazioni, consulta Accesso multi-account.
3. Se non vi è alcuna dichiarazione con un esplicito `allow` che corrisponde all'azione di richiesta e alla risorsa:

   AWS nega la richiesta per impostazione predefinita, che viene chiamata implicito diniego.

Per visualizzare le policy basate sulle risorse per un segreto

- Completa una delle seguenti operazioni:
  - Apri la console Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/. Nella pagina dei dettagli segreti per il segreto, nella sezione Autorizzazioni risorse, scegliereModifica autorizzazioni.
  - Utilizzo della AWS CLI o della `get-resource-policy` o dell'SDK AWS per chiamare `GetResourcePolicy`.

Per determinare chi ha accesso tramite policy basate sull'identità

- Usa il simulatore di policy IAM. Vedere Test delle policy &IAM; con il simulatore di policy &IAM

Autorizzazioni ai segreti AWS Secrets Manager per gli utenti in un account diverso

Per consentire agli utenti in un account di accedere ai segreti in un altro account (accesso tra account), è necessario consentire l'accesso sia in una policy delle risorse che in una policy di identità. Ciò è diverso dalla concessione dell'accesso alle identità nello stesso account del segreto.

È inoltre necessario consentire all'identità di utilizzare la chiave KMS con cui il segreto è crittografato. Questo perché non è possibile utilizzare l'opzione Chiave gestita da AWS (`aws/secretsmanager`) per l'accesso tra account. È invece necessario crittografare il segreto con una chiave KMS creata e quindi allegare ad esso un criterio chiave. È previsto un addebito per la creazione di chiavi KMS. Per modificare la chiave di crittografia per un segreto, vedere the section called “Modificare un segreto”.

I criteri di esempio seguenti presuppongono di disporre di una chiave segreta e di crittografia in Account1, e un'identità in Account2 che vuoi consentire ad accedere al valore segreto.

Fase 1: Allegare una policy delle risorse al segreto in Account1

- La policy seguente permette *`ApplicationRole`* in *`Account2`* per accedere al segreto in *`Account1`*. Per visualizzare questa policy, vedere the section called “Allegare una policy di autorizzazione a un segreto”.

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

Fase 2: Aggiungere un'istruzione alla policy della chiave per la chiave KMS in Account1

- La seguente dichiarazione di criteri chiave consente *ApplicationRole* in *Account2* per utilizzare il tasto del servizio di gestione delle chiavi in *Account1* per decifrare il segreto in *Account1*. Per utilizzare questa istruzione, aggiungerla al criterio chiave per la chiave KMS. Per ulteriori informazioni, vedere Modifica di una policy delle chiavi.

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

Fase 3: Allegare una policy di identità all'identità in Account2

- La policy seguente permette *ApplicationRole* in *Account2* per accedere al segreto in *Account1* e decrittografare il valore segreto utilizzando la chiave di crittografia che è anche in *Account1*. Per utilizzare questa policy, vedere the section called “Allegare un policy di autorizzazione a un'identità”. Puoi trovare l'ARN per il tuo segreto nella console di Secrets Manager nella pagina dei dettagli segreti sotto ARN del segreto. In alternativa, è possibile chiamare `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"
```

```
    }
  ]
}
```

Autorizzazioni del ruolo di esecuzione della funzione di rotazione Lambda per AWS Secrets Manager

Secrets Manager utilizza una funzione Lambda per ruotare un segreto. Per l'esecuzione della funzione Lambda, Lambda assume un ruolo di esecuzione IAM e fornisce tali credenziali al codice della funzione Lambda. Per istruzioni sull'impostazione della rotazione automatica, consulta:

- Rotazione automatica per i segreti del database (console)
- Rotazione automatica (console)
- Rotazione automatica (AWS CLI)

Negli esempi seguenti vengono illustrate le policy in linea per i ruoli di esecuzione della funzione di rotazione Lambda. Per creare un ruolo di esecuzione e allegare una policy di autorizzazione, vedere Ruolo di esecuzione di AWS Lambda.

Esempi:

- Policy per un ruolo di esecuzione della funzione di rotazione Lambda
- Istruzione della policy per una chiave gestita dal cliente
- Istruzione della policy per la strategia a utenti alternati

Policy per un ruolo di esecuzione della funzione di rotazione Lambda

La seguente policy di esempio consente alla funzione di rotazione di:

- Eseguire le operazioni di Gestione dei segreti per *SecretARN*.
- Creare una nuova password.
- Impostare la configurazione richiesta se il database o il servizio è in esecuzione in un VPC. Consulta Configurazione di una funzione Lambda per accedere alle risorse in un 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"
        }
    ]
}
```

Istruzione della policy per una chiave gestita dal cliente

Se il segreto è crittografato con una chiave KMS diversa da Chiave gestita da AWS `aws/secretsmanager`, allora è necessario concedere l'autorizzazione per utilizzare la chiave al ruolo di esecuzione Lambda. È possibile utilizzare il contesto di crittografia SecretARN per limitare l'uso della funzione di decrittografia, in modo che il ruolo della funzione di rotazione sia autorizzato ad accedere soltanto per decrittografare il segreto della cui rotazione è responsabile. Nell'esempio seguente viene illustrata un'istruzione da aggiungere alla policy del ruolo di esecuzione per decrittografare il segreto utilizzando la chiave KMS.

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

Per utilizzare la funzione di rotazione per più segreti crittografati con una chiave gestita dal cliente, aggiungi un'istruzione come l'esempio seguente per consentire al ruolo di esecuzione di decrittografare il segreto.

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

Istruzione della policy per la strategia a utenti alternati

Per informazioni sulla strategia di rotazione a utenti alternati, consulta la sezione the section called “Strategia di rotazione”.

Per un segreto che contiene credenziali Amazon RDS, se utilizzi la strategia degli utenti alternati e il segreto del superutente è gestito da Amazon RDS, devi anche consentire alla funzione di rotazione di chiamare le API di sola lettura su Amazon RDS in modo che possa ottenere le informazioni di connessione per il database. Ti consigliamo di allegare la AWSpolitica gestita AmazonRDSReadOnlyAccess.

La seguente policy di esempio consente alla funzione di:

- Eseguire le operazioni di Gestione dei segreti per *SecretARN*.
- Recupera le credenziali nel segreto del superutente. Gestione dei segreti utilizza le credenziali presenti nel segreto del superutente per aggiornare le credenziali nel segreto che viene ruotato.
- Creare una nuova password.
- Impostare la configurazione richiesta se il database o il servizio è in esecuzione in un VPC. Per ulteriori informazioni, consulta Configurazione di una funzione Lambda per accedere alle risorse in un 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"
        }
    ]
}
```

Esempi di policy di autorizzazione per AWS Secrets Manager

Una policy di autorizzazione è il testo strutturato JSON. Vedere Struttura dei documenti di policy JSON

Le policy di autorizzazione associati alle risorse e alle identità sono molto simili. Alcuni elementi inclusi in una policy per l'accesso ai segreti includono:

- `Principal`: a chi concedere l'accesso. Vedere Specifica di un principale nel manuale utente IAM. Quando si allega una policy a un'identità, non si include un elemento `Principal` nella policy.
- `Action`: cosa possono fare. Per informazioni, consultare the section called “Operazioni di Secrets Manager”.
- `Resource`: a quali segreti possono accedere. Per informazioni, consultare the section called “Risorse di Secrets Manager”.

  Il carattere jolly (*) ha un significato diverso in base a cui si collega la policy:

  - In una policy collegata a un segreto, * significa che la policy si applica a questo segreto.
  - In un criterio collegato a un'identità, * significa che il criterio si applica a tutte le risorse, inclusi i segreti, nell'account.

Per collegare una policy a un segreto, consultare the section called “Allegare una policy di autorizzazione a un segreto”.

Per collegare una policy a un'identità, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

Argomenti

- Esempio: Autorizzazione per recuperare valori segreti individuali
- Autorizzazione per il recupero di un gruppo di valori segreti in un batch
- Esempio: il carattere jolly
- Esempio: Autorizzazione per creare segreti
- Esempio: autorizzazioni e VPC
- Esempio: Controllare l'accesso ai segreti utilizzando i tag
- Esempio: Limitare l'accesso alle identità con tag che corrispondono ai tag dei segreti
- Esempio: Principale del servizio

Esempio: Autorizzazione per recuperare valori segreti individuali

Per concedere il permesso di recuperare valori segreti, è possibile allegare policy a segreti o identità. Per informazioni sul tipo di criterio da utilizzare, vedere Policy basate su identità e policy basate su risorse. Per informazioni sul collegamento di una policy a un'identità, vedere the section called “Allegare una policy di autorizzazione a un segreto” e the section called “Allegare un policy di autorizzazione a un'identità”.

Negli esempi seguenti vengono illustrati due modi differenti per concedere l'accesso a un segreto. Il primo esempio è una policy basata su risorse che possono essere collegate a un segreto. Questo esempio è utile quando si desidera concedere l'accesso a un singolo segreto a più utenti o ruoli. Il secondo esempio è un criterio basato sull'identità che è possibile associare a un utente o a un ruolo in IAM. Questo esempio è utile quando si desidera concedere l'accesso a un gruppo IAM. Per concedere l'autorizzazione a recuperare un gruppo di segreti in una chiamata API batch, consulta the section called “Autorizzazione per il recupero di un gruppo di valori segreti in un batch”.

Example Leggi un segreto (allega un segreto)

È possibile concedere l'accesso a un segreto allegando a tale segreto la policy seguente. Per utilizzare questa policy, consultare the section called “Allegare una policy di autorizzazione a un segreto”.

```
{
```

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

Example Leggi un segreto (allega un'identità)

È possibile concedere l'accesso a un segreto allegando a un'identità la policy seguente. Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”. Se si allega questa policy al ruolo *EC2RoletoAccessSecrets*, concede le stesse autorizzazioni della policy precedente.

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

Example Leggere un segreto crittografato utilizzando una chiave gestita dal cliente (allegarla all'identità)

Se un segreto viene crittografato utilizzando una chiave gestita dal cliente, puoi concedere l'accesso per leggere il segreto allegando la seguente policy a un'identità. Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

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

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

Autorizzazione per il recupero di un gruppo di valori segreti in un batch

Example Leggi un gruppo di segreti in un batch (allega all'identità)

Allegando a un'identità la policy seguente, è possibile concedere l'accesso per recuperare un gruppo di segreti in una chiamata API batch. La policy limita il chiamante in modo che possa recuperare solo i segreti specificati da *SecretARN1*, *SecretARN2* e *SecretARN3* anche se la chiamata batch include altri segreti. Se il chiamante richiede anche altri segreti nella chiamata API batch, Secrets Manager non li restituirà. Per ulteriori informazioni, consulta the section called “Recupera segreti in un batch”. Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

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

```
        "SecretARN2",
        "SecretARN3"
        ]
    }
  ]
}
```

Esempio: il carattere jolly

È possibile utilizzare caratteri jolly per includere un set di valori in un elemento della policy.

Example Accedi a tutti i segreti di un percorso (allega all'identità)

La policy seguente concede l'accesso al recupero di tutti i segreti con un nome che inizia con "*TestEnv/*". Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

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

Example Accedi ai metadati su tutti i segreti (allegati all'identità)

Le seguenti sovvenzioni `DescribeSecret` e le autorizzazioni che iniziano con `List`: `ListSecrets` e `ListSecretVersionIds`. Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

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

```
    }
}
```

Example Corrispondenza al nome segreto (allega all'identità)

La policy seguente concede tutte le autorizzazioni di Secrets Manager per un segreto, per nome. Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

Per abbinare un nome segreto, creare l'ARN per il segreto mettendo insieme la regione, l'ID dell'Account, il nome segreto e il carattere jolly (?) per abbinare singoli caratteri casuali. Secrets Manager aggiunge sei caratteri casuali ai nomi segreti come parte del loro ARN, per poter utilizzare questo carattere jolly per abbinare tali caratteri. Se si utilizza la sintassi `"another_secret_name-*"`, Secret Manager individua la corrispondenza non solo del determinato segreto con i 6 caratteri casuali, ma anche di `"another_secret_name-<anything-here>a1b2c3"`.

Perché puoi prevedere tutte le parti dell'ARN di un segreto eccetto i 6 caratteri casuali, usando il carattere jolly `'??????'` la sintassi ti consente di concedere le autorizzazioni in modo sicuro a un segreto che non esiste ancora. Tuttavia, tieni presente che, se elimini il segreto e lo ricrei con lo stesso nome, l'utente riceve automaticamente l'autorizzazione per il nuovo segreto, anche se i 6 caratteri sono cambiati.

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

Esempio: Autorizzazione per creare segreti

Per concedere a un utente le autorizzazioni per creare un segreto, allegare una policy di autorizzazioni a un gruppo IAM a cui appartiene l'utente. Consultare Gruppi di utenti IAM.

Example Creare segreti (allegati all'identità)

La policy seguente concede l'autorizzazione per creare segreti e visualizzare un elenco di segreti. Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

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

Esempio: autorizzazioni e VPC

Se è necessario accedere a Secrets Manager da un VPC, è possibile assicurarsi che le richieste a Secrets Manager provengano dal VPC includendo una condizione nelle policy di autorizzazione. Per ulteriori informazioni, consultare Condizioni dell'endpoint VPC e Endpoint VPC.

Assicurati che le richieste di accesso al segreto da altri AWS provengono anche dal VPC, altrimenti questa policy negherà loro l'accesso.

Example Richiedere che le richieste vengano inviate tramite un endpoint VPC (collegamento a segreto)

La seguente policy consente all'utente di eseguire operazioni Secret Manager solo quando la richiesta proviene tramite l'endpoint VPC specificato *vpce-1234a5678b9012c*. Per utilizzare questa policy, consultare the section called “Allegare una policy di autorizzazione a un segreto”.

```
{
  "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 Richiedere che le richieste provengano da un VPC (allegati al segreto)

I seguenti comandi della policy consentono di creare e gestire i segreti solo quando la loro provenienza è *vpc-12345678*. Inoltre, la policy consente le operazioni che utilizzano il valore del segreto crittografato solo quando le richieste provengono da vpc-2b2b2b2b. Puoi usare una policy come questa se esegui un'applicazione in un VPC, ma utilizzi un secondo VPC separato per le funzioni di gestione. Per utilizzare questa policy, consultare the section called “Allegare una policy di autorizzazione a un segreto”.

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

Esempio: Controllare l'accesso ai segreti utilizzando i tag

Puoi utilizzare i tag per controllare l'accesso ai segreti. Usare i tag per controllare le autorizzazioni è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa complicata. Una strategia è quella di allegare tag ai segreti e quindi concedere le autorizzazioni a un'identità quando un segreto ha un tag specifico.

Example Consenti l'accesso ai segreti con un tag specifico (allegata a un'identità)

La policy seguente consente `DescribeSecret` sui segreti con un tag con la chiave "*ServerName (Nome server)*" e il valore "*ServerABC*". Per utilizzare questa policy, consultare the section called “Allegare un policy di autorizzazione a un'identità”.

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

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

Esempio: Limitare l'accesso alle identità con tag che corrispondono ai tag dei segreti

Una strategia è quella di allegare tag sia ai segreti che alle identità IAM. Quindi creare policy di autorizzazioni per consentire operazioni quando il tag dell'identità corrisponde al tag del segreto. Per un tutorial completo, consultare Definire le autorizzazioni per accedere ai segreti in base ai tag.

Utilizzare i tag per controllare le autorizzazioni è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa complicata. Per ulteriori informazioni, consulta Che cos'è ABAC per AWS?

Example Consentire l'accesso a ruoli con gli stessi tag dei segreti (allegati a un segreto)

La seguente policy garantisce `GetSecretValue` all'account *123456789012* solo se il tag *AccessProject* ha lo stesso valore per il segreto e il ruolo. Per utilizzare questa policy, consultare the section called “Allegare una policy di autorizzazione a un segreto”.

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

Esempio: Principale del servizio

Se la policy sulle risorse collegata al tuo segreto include un principale del servizio AWS, ti consigliamo di utilizzare le chiavi di condizione globali aws:SourceArn e aws:SourceAccount. I valori ARN e account sono inclusi nel contesto di autorizzazione solo quando una richiesta arriva a Secrets Manager da un altro servizio AWS. Questa combinazione di condizioni evita un potenziale confused deputy scenario (scenario "deputy confused").

Se una risorsa ARN include caratteri non consentiti in una policy di risorse, non potrai utilizzare l'ARN di tale risorsa nel valore della chiave di condizione `aws:SourceArn`. Devi invece utilizzare la chiave di condizione `aws:SourceAccount`. Per ulteriori informazioni, consulta Requisiti IAM.

I principali del servizio in genere non vengono utilizzati come principali in un criterio collegato a un segreto, ma alcuni servizi AWS lo richiedono. Per informazioni sulle policy delle risorse che un servizio richiede di allegare a un segreto, consultare la documentazione del servizio.

Example Consenti a un servizio di accedere a un segreto utilizzando un principale di servizio (collegamento a un segreto)

```
{
  "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"
        }
      }

    }
  ]
```

```
}
```

Riferimento per le autorizzazioni per AWS Secrets Manager

Per visualizzare gli elementi che costituiscono una policy di autorizzazioni, consulta Struttura dei documenti di policy JSON e Riferimento agli elementi delle policy JSON IAM.

Per iniziare a scrivere la propria policy di autorizzazione, vedere the section called “Esempi di policy di autorizzazione”.

Operazioni di Secrets Manager

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
CancelRotateSecret	Concede l'autorizzazione per annullare una rotazione di segreti in avanzamento	Scrittura	Secret*	secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey}	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
CreateSecret	Concede l'autorizzazione per creare un segreto che memorizza dati crittografati su cui è possibile effettuare query e rotazioni	Scrittura	Secret*	secretsmanager:Name secretsmanager:Description secretsmanager:KmsKeyId aws:RequestTag/${TagKey} aws:ResourceTag/${TagKey} aws:TagKeys secretsmanager:ResourceTag/tag-key secretsmanager:AddReplicaRegions	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:ForceOverwriteReplicaSecret	
DeleteResourcePolicy	Concede l'autorizzazione per eliminare la policy della risorsa collegata a un segreto	Gestione delle autorizzazioni	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
DeleteSecret	Concede l'autorizzazione per eliminare un segreto	Scrittura	Secret*		

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:RecoveryWindowInDays secretsmanager:ForceDeleteWithoutRecovery secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:Sec	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				retPrimaryRegion	
DescribeSecret	Concede l'autorizzazione per recuperare i metadati di un segreto, ma non i dati crittografati	Lettura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetRandomPassword	Concede l'autorizzazione per generare una stringa casuale da usare per la creazione di password	Lettura			

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
GetResourcePolicy	Concede l'autorizzazione per ottenere la policy della risorsa collegata a un segreto	Lettura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
GetSecretValue	Concede l'autorizzazione per recuperare e decrittare i dati crittografati	Lettura	Secret*		

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:SecretId secretsmanager:VersionId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
ListSecretVersionIds	Concede l'autorizzazione per elencare le versioni disponibili di un segreto	Lettura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
ListSecrets	Concede l'autorizzazione per elencare i segreti disponibili	Elenco			
PutResourcePolicy	Concede l'autorizzazione per allegare una policy della risorsa a un segreto	Gestione delle autorizzazioni	Secret*		

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:BlockPublicPolicy secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
PutSecretValue	Concede l'autorizzazione per creare una nuova versione del segreto con i nuovi dati crittografati	Scrittura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RemoveRegionsFromReplication	Concede l'autorizzazione per rimuovere le regioni dalla replica	Scrittura	Secret*		

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
ReplicateSecretToRegions	Concede l'autorizzazione per convertire un segreto esistente in un segreto multi-regione e iniziare a replicarlo in un elenco di nuove regioni	Scrittura	Secret*	secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:AddReplicaRegions secretsmanager:For	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				ceOverwriteReplicaSecret	
RestoreSecret	Concede l'autorizzazione per annullare l'eliminazione di un segreto	Scrittura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	
RotateSecret	Concede l'autorizzazione per avviare la rotazione di un segreto	Scrittura	Secret*		

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				secretsmanager:SecretId secretsmanager:RotationLambdaARN secretsmanager:resource/AllowRotationLambdaAr n secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion secretsmanager:Mod	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				ifyRotationRules secretsmanager:RotateImmediately	
StopReplicationToReplica	Concede l'autorizzazione per rimuovere il segreto dalla replica e promuoverlo a segreto regionale nella regione della replica	Scrittura	Secret*		
				secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
TagResource	Concede l'autorizzazione per aggiungere tag a un segreto	Assegnazione di tag	Secret*	secretsmanager:SecretId aws:RequestTag/${TagKey} aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
UntagResource	Concede l'autorizzazione per rimuovere tag da un segreto	Assegnazione di tag	Secret*	secretsmanager:SecretId aws:TagKeys secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
UpdateSecret	Concede l'autorizzazione per aggiornare un segreto con nuovi metadati o con una nuova versione dei dati crittografati	Scrittura	Secret*	secretsmanager:SecretId secretsmanager:Description secretsmanager:KmsKeyId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:Sec	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
				retPrimaryRegion	
UpdateSecretVersionStage	Concede l'autorizzazione per spostare una fase da un segreto a un altro	Scrittura	Secret*		
				secretsmanager:SecretId secretsmanager:VersionStage secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Operazioni	Descrizione	Livello di accesso	Tipi di risorsa (*obbligatorio)	Chiavi di condizione	Operazioni dipendenti
ValidateResourcePolicy	Concede l'autorizzazione per convalidare una policy della risorsa prima di allegare la policy	Gestione delle autorizzazioni	Secret*	secretsmanager:SecretId secretsmanager:resource/AllowRotationLambdaArn secretsmanager:ResourceTag/tag-key aws:ResourceTag/${TagKey} secretsmanager:SecretPrimaryRegion	

Risorse di Secrets Manager

Tipi di risorsa	ARN	Chiavi di condizione
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 genera l'ultima parte dell'ARN segreto aggiungendo un trattino e sei caratteri alfanumerici casuali alla fine del nome del segreto. Se elimini un segreto e ne ricrei un altro con lo stesso nome, questa formattazione garantisce che gli utenti con autorizzazioni di accesso al segreto originale non ottengono automaticamente l'accesso al nuovo segreto perché Secrets Manager genera sei nuovi caratteri casuali.

È possibile trovare l'ARN per un segreto nella console di Secrets Manager nella pagina dei dettagli segreti o chiamando `DescribeSecret`.

Chiavi di condizione

Se si includono condizioni di stringa dalla tabella seguente nella policy delle autorizzazioni, i chiamanti di Secrets Manager devono specificare il parametro corrispondente oppure l'accesso viene negato. Per evitare la negazione dell'accesso ai chiamanti per un parametro mancante, aggiungi `IfExists` alla fine del nome operatore di condizione, ad esempio `StringLikeIfExists`. Per ulteriori informazioni, consulta Elementi della policy JSON IAM: operatori di condizione.

Chiavi di condizione	Descrizione	Tipo
aws:RequestTag/${TagKey}	Filtra l'accesso in base a una chiave presente nella richiesta effettuata dall'utente al servizio Secrets Manager	Stringa
aws:ResourceTag/${TagKey}	Filtra l'accesso per i tag associati alla risorsa	Stringa
aws:TagKeys	Filtra l'accesso in base all'elenco di tutti i nomi delle chiavi di tag presenti nella richiesta effettuata dall'utente al servizio Secrets Manager	ArrayOfString
secretsmanager:AddReplicaRegions	Filtra l'accesso in base all'elenco delle regioni in cui replicare il segreto	ArrayOfString
secretsmanager:BlockPublicPolicy	Filtra l'accesso in base alla policy delle risorse che blocca o meno l'accesso all'Account AWS	Bool
secretsmanager:Description	Filtra l'accesso in base al testo di descrizione nella richiesta	Stringa
secretsmanager:ForceDeleteWithoutRecovery	Filtra l'accesso in base al fatto che il segreto debba essere eliminato immediatamente senza alcuna finestra di ripristino	Bool
secretsmanager:ForceOverwriteReplicaSecret	Filtra l'accesso in base alla sovrascrittura di un segreto con lo stesso nome nella regione di destinazione	Bool
secretsmanager:KmsKeyId	Filtra l'accesso in base all'ARN della chiave di KMS nella richiesta	Stringa

Chiavi di condizione	Descrizione	Tipo
secretsmanager:ModifyRotationRules	Filtra l'accesso in base al fatto che le regole di rotazione del segreto debbano essere modificate	Bool
secretsmanager:Name	Filtra l'accesso in base al nome descrittivo del segreto nella richiesta	Stringa
secretsmanager:RecoveryWindowInDays	Filtra l'accesso in base al numero di giorni che Secrets Manager attende prima di eliminare il segreto	Numerico
secretsmanager:ResourceTag/tag-key	Filtra l'accesso in base a una coppia chiave/valore di tag	Stringa
secretsmanager:RotateImmediately	Filtra l'accesso in base al fatto che il segreto debba essere ruotato immediatamente	Bool
secretsmanager:RotationLambdaARN	Filtra l'accesso in base all'ARN della funzione Lambda di rotazione nella richiesta	ARN
secretsmanager:SecretId	Filtra l'accesso in base al valore SecretID nella richiesta	ARN
secretsmanager:SecretPrimaryRegion	Filtra l'accesso in base alla regione primaria in cui viene creato il segreto	Stringa
secretsmanager:VersionId	Filtra l'accesso in base all'identificatore univoco della versione del segreto nella richiesta	Stringa

Chiavi di condizione	Descrizione	Tipo
secretsmanager:VersionStage	Filtra l'accesso in base all'elenco delle fasi della versione nella richiesta	Stringa
secretsmanager:resource/AllowRotationLambdaArn	Filtra l'accesso in base all'ARN della funzione Lambda di rotazione associata al segreto	ARN

Blocca un accesso ampio ai segreti con la condizione `BlockPublicPolicy`

Nelle policy di identità che consentono l'operazione `PutResourcePolicy`, si consiglia di utilizzare `BlockPublicPolicy: true`. Con questa condizione gli utenti possono allegare una policy delle risorse a un segreto se tale policy non consente un accesso ampio.

Secrets Manager utilizza il ragionamento automatizzato di Zelkova per analizzare le policy delle risorse per un accesso ampio. Per ulteriori informazioni su Zelkova, consulta Come AWS utilizza il ragionamento automatico per aiutarti a raggiungere la sicurezza su larga scala sul AWSBlog sulla sicurezza.

L'esempio seguente mostra come utilizzare `BlockPublicPolicy`.

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

Condizioni indirizzo IP

Fai attenzione quando specifichi gli operatori di condizione con indirizzo IP o la chiave di condizione `aws:SourceIp` nella stessa istruzione di policy che consente o rifiuta l'accesso a Secrets Manager. Ad esempio, e si collega una policy che limita AWS operazioni che richiede dagli indirizzi IP della rete aziendale al segreto, le richieste come un utente IAM che richiama la richiesta dalla rete aziendale funziona come previsto. Tuttavia, se abiliti altri servizi per accedere al segreto a tuo nome, come quando abiliti la funzione di rotazione con una funzione Lambda, quella funzione chiama le operazioni Secrets Manager da uno spazio di indirizzo interno AWS. Le richieste influenzate dalla policy con il filtro degli indirizzi IP non vanno a buon fine.

Inoltre, la chiave di condizione `aws:sourceIP` diventa meno efficace quando la richiesta proviene da un endpoint Amazon VPC. Per limitare le richieste a un determinato endpoint VPC, utilizza the section called “Condizioni dell'endpoint VPC”.

Condizioni dell'endpoint VPC

Per consentire o negare l'accesso alle richieste provenienti da un determinato VPC o endpoint VPC, utilizza `aws:SourceVpc` per limitare l'accesso alle richieste dal VPC specificato o `aws:SourceVpce` per limitare l'accesso alle richieste dall'endpoint VPC specificato. Per informazioni, consultare the section called “Esempio: autorizzazioni e VPC”.

- `aws:SourceVpc` limita l'accesso alle richieste dal VPC specificato.
- `aws:SourceVpce` limita l'accesso alle richieste dall'endpoint VPC specificato.

Se si utilizzano queste chiavi di condizione in una istruzione di policy di risorsa che consente o rifiuta l'accesso ai segreti Secrets Manager puoi inavvertitamente negare l'accesso ai servizi che utilizzano Secrets Manager per accedere ai segreti a tuo nome. Solo alcuni servizi AWS possono essere eseguiti con un endpoint all'interno di VPC. Se limiti le richieste di un segreto a un VPC o endpoint VPC, le chiamate a Secrets Manager da un servizio non configurato per il servizio possono non andare a buon fine.

Per informazioni, consultare Endpoint VPC.

Creazione e gestione dei segreti con AWS Secrets Manager

Un segreto può essere costituito da una password, da un insieme di credenziali, ad esempio un nome utente e una password, da un token OAuth o da altre informazioni del segreto archiviate in un formato crittografato in Secrets Manager.

Argomenti

- Creare un AWS Secrets Manager database segreto
- Struttura dei segreti JSON AWS Secrets Manager
- Creazione di un segreto AWS Secrets Manager
- Aggiorna il valore di un segreto AWS Secrets Manager
- Modificare la chiave di crittografia per un AWS Secrets Manager segreto
- Modificare un segreto AWS Secrets Manager
- Scopri i segreti in AWS Secrets Manager
- Eliminare un segreto AWS Secrets Manager
- Ripristino di un segreto AWS Secrets Manager
- Replica di un segreto AWS Secrets Manager per altri Regioni AWS
- Promozione di un segreto di replica a segreto autonomo in AWS Secrets Manager
- Tag segreti AWS Secrets Manager

Creare un AWS Secrets Manager database segreto

Dopo aver creato un utente in Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB, puoi archiviare le rispettive credenziali in Secrets Manager mediante la seguente procedura. Quando utilizzi l'SDK AWS CLI o uno degli SDK per archiviare il segreto, devi fornire il segreto nella struttura JSON corretta. Quando utilizzi la console per archiviare un segreto del database, Secrets Manager lo crea automaticamente nella struttura JSON corretta.

> **Tip**
>
> Per le credenziali utente amministratore di Amazon RDS e Amazon Redshift, ti consigliamo di utilizzare i segreti gestiti. Il segreto gestito viene creato tramite il servizio di gestione, quindi è possibile utilizzare la rotazione gestita.

Quando archivi le credenziali del database per un database di origine replicato in altre regioni, il segreto contiene informazioni di connessione per il database di origine. Se poi replichi il segreto, le repliche saranno copie del segreto di origine e conterranno le stesse informazioni di connessione. Puoi aggiungere altre coppie chiave/valore al segreto per le informazioni sulla connessione della regione.

Per creare un segreto, sono necessarie le autorizzazioni concesse da. SecretsManagerReadWriteAWS politiche gestite

Secrets Manager genera una voce di CloudTrail registro quando si crea un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Come creare un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:
   a. Per Secret type (Tipo di segreto), scegli il tipo di credenziali del database da archiviare:
      - Database di Amazon RDS (include Aurora)
      - Database di Amazon DocumentDB
      - Magazzino di dati Amazon Redshift
   b. Per Credentials (Credenziali), inserisci le credenziali per il database.
   c. Per la chiave di crittografia, scegli AWS KMS key quella utilizzata da Secrets Manager per crittografare il valore segreto. Per ulteriori informazioni, consulta Crittografia e decrittografia del segreto.
      - Per la maggior parte dei casi, scegli aws/secretmanager per utilizzare la Chiave gestita da AWS per Secrets Manager. L'utilizzo di questa chiave non prevede costi aggiuntivi.
      - Se devi accedere al segreto da un'altra persona Account AWS o se desideri utilizzare la tua chiave KMS in modo da poterla ruotare o applicare una politica di chiave, scegli una chiave gestita dal cliente dall'elenco o scegli Aggiungi nuova chiave per crearne una. Per informazioni sui costi di utilizzo di una chiave gestita dal cliente, consulta la sezione Prezzi.

È necessario avere le the section called “Autorizzazioni per la chiave KMS”. Per informazioni sull'accesso tra account, consulta the section called “Accesso multi-account”.

d. Per Database, scegli il database.

e. Seleziona Successivo.

4. Nella pagina Configure secret (Configura il segreto), effettua le seguenti operazioni:

a. Inserisci un Secret name (Nome del segreto) e una Description (Descrizione) descrittivi. I nomi dei segreti devono contenere un numero di caratteri Unicode compreso tra 1 e 512.

b. (Facoltativo) Nella sezione Tags (Tag) aggiungere tag al segreto. Per le strategie di assegnazione dei tag, vedere the section called “Tag segreti ”. Non archiviare informazioni sensibili nei tag perché non sono crittografate.

c. (Facoltativo) In Permessi delle risorse, per aggiungere una policy delle risorse al tuo segreto, scegli Modifica delle autorizzazioni. Per ulteriori informazioni, consulta the section called “Allegare una policy di autorizzazione a un segreto”.

d. (Facoltativo) In Replica segreto, per replicare il tuo segreto su un altro Regione AWS, scegli Replica segreto. Puoi replicare il tuo segreto immediatamente o tornare e replicarlo in un secondo momento. Per ulteriori informazioni, consulta la pagina Replica un segreto ad altre Regioni.

e. Seleziona Next (Successivo).

5. (Facoltativo) Nella pagina Configure rotation (Configura la rotazione), puoi attivare la rotazione automatica. Puoi anche disattivare la rotazione e poi riattivarla in un secondo momento. Per ulteriori informazioni, consulta la pagina Rotazione dei segreti . Seleziona Next (Successivo).

6. Nella pagina Review (Revisione), rivedi i dettagli dei segreti e quindi scegli Store (Archivia).

Secrets Manager ritorna all'elenco dei segreti. Se il segreto nuovo non viene visualizzato, scegli il pulsante aggiorna.

AWS CLI

Quando immetti i comandi in una shell dei comandi, c'è il rischio che la cronologia dei comandi sia accessibile o che le utilità abbiano accesso ai parametri dei comandi. Per informazioni, consulta the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

Example Creazione di un segreto dalle credenziali in un file JSON

L'esempio di `create-secret` seguente mostra come creare un segreto partendo dalle credenziali in un file. Per ulteriori informazioni, consultate Caricamento AWS CLI dei parametri da un file nella Guida per l' AWS CLI utente.

Affinché Secrets Manager sia in grado di ruotare il segreto, devi assicurarti che il JSON corrisponda alla Struttura JSON di un segreto .

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

Contenuti di 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

Per creare un segreto utilizzando uno degli AWS SDK, usa l'azione `CreateSecret`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Struttura dei segreti JSON AWS Secrets Manager

Puoi archiviare qualsiasi testo o binario nei segreti di Secrets Manager. Se desideri attivare la rotazione automatica per un segreto in Secrets Manager, il segreto deve trovarsi nella struttura JSON corretta. Durante la rotazione, Secrets Manager utilizza le informazioni nel segreto per connettersi all'origine delle credenziali e aggiornare le credenziali. I nomi delle chiavi JSON fanno distinzione tra maiuscole e minuscole.

Quando utilizzi la console per archiviare un segreto del database, Secrets Manager lo crea automaticamente nella struttura JSON corretta.

Puoi aggiungere altre coppie chiave/valore a un segreto, ad esempio in un segreto di un database, per contenere informazioni sulla connessione per i database di replica in altre Regioni.

Argomenti

- Struttura del segreto di Amazon RDS Db2
- Struttura del segreto di MariaDB di Amazon RDS
- Struttura del segreto di Amazon RDS e Amazon Aurora MySQL
- Struttura del segreto di Oracle di Amazon RDS
- Struttura del segreto di Amazon RDS e Amazon Aurora PostgreSQL
- Struttura del segreto di Microsoft SQLServer di Amazon RDS
- Struttura del segreto di Amazon DocumentDB
- Struttura del segreto di Amazon Redshift
- Struttura segreta di Amazon Redshift Serverless
- Struttura ElastiCache segreta di Amazon

Struttura del segreto di Amazon RDS Db2

Poiché gli utenti non possono modificare le proprie password, per le istanze Amazon RDS Db2 è necessario fornire le credenziali di amministratore in un segreto separato.

```
{
  "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>"
}
```

Struttura del segreto di MariaDB di Amazon RDS

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

```
  "password": "<password>",
  "dbname": "<database name. If not specified, defaults to None>",
  "port": <TCP port number. If not specified, defaults to 3306>
}
```

Per utilizzarethe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura del segreto di Amazon RDS e Amazon Aurora MySQL

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

Per utilizzare ilthe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura del segreto di Oracle di Amazon RDS

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

Per utilizzare ilthe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura del segreto di Amazon RDS e Amazon Aurora PostgreSQL

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

Per utilizzare ilthe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura del segreto di Microsoft SQLServer di Amazon RDS

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

Per utilizzare ilthe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura del segreto di 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>
}
```

Per utilizzare ilthe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>
}
```

Struttura del segreto di 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>
}
```

Per utilizzare ilthe section called “Utenti alternati”, includi il segreto che contiene `masterarn` le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura segreta di Amazon Redshift Serverless

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

Per utilizzarethe section called “Utenti alternati”, includi il comando `masterarn` for the secret che contiene le credenziali di amministratore o superutente.

```
{
  "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>"
}
```

Struttura ElastiCache segreta di Amazon

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

Per ulteriori informazioni, consulta Rotazione automatica delle password per gli utenti nella Amazon ElastiCache User Guide.

Creazione di un segreto AWS Secrets Manager

Per archiviare le chiavi API, i token di accesso, le credenziali che non sono per i database e altri segreti in Secrets Manager, segui questi passaggi. Se si desidera attivare la rotazione per un segreto Amazon ElastiCache, è necessario archiviare il segreto nella struttura JSON prevista.

Per creare un segreto, sono necessarie le autorizzazioni dalla AWS politiche gestite SecretsManagerReadWrite.

Secrets Manager crea una voce nel file di log di CloudTrail quando crea un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Come creare un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:

   a. Per Secret type (Tipo di segreto), scegli Other type of secret (Altro tipo di segreto).

   b. In Coppie chiave/valore, inserisci il segreto sotto forma di coppie Chiave/valore JSON oppure scegli la scheda Testo normale e inserisci il segreto in qualsiasi formato. Puoi archiviare fino a 65536 byte nel segreto.

   c. Per Encryption key (Chiave di crittografia), scegli la AWS KMS key che utilizza Secrets Manager per crittografare il valore del segreto. Per ulteriori informazioni, consulta Crittografia e decrittografia del segreto.

      - Per la maggior parte dei casi, scegli aws/secretmanager per utilizzare la Chiave gestita da AWS per Secrets Manager. L'utilizzo di questa chiave non prevede costi aggiuntivi.
      - Se hai bisogno di accedere al segreto da un altro Account AWS oppure se si desideri utilizzare la tua chiave KMS in modo da poterla ruotare o applicare ad essa un policy di chiavi, scegli una chiave gestita dal cliente dall'elenco o seleziona Add new key (Aggiungi nuova chiave) per crearne una. Per informazioni sui costi di utilizzo di una chiave gestita dal cliente, consulta la sezione Prezzi.

        È necessario avere le the section called “Autorizzazioni per la chiave KMS”. Per informazioni sull'accesso tra account, consulta the section called “Accesso multi-account”.

d. Seleziona Successivo.

4. Nella pagina Configure secret (Configura il segreto), effettua le seguenti operazioni:

   a. Inserisci un Secret name (Nome del segreto) e una Description (Descrizione) descrittivi. I nomi dei segreti devono contenere un numero di caratteri Unicode compreso tra 1 e 512.

   b. (Facoltativo) Nella sezione Tags (Tag) aggiungere tag al segreto. Per le strategie di assegnazione dei tag, vedere the section called “Tag segreti ”. Non archiviare informazioni sensibili nei tag perché non sono crittografate.

   c. (Facoltativo) In Permessi delle risorse, per aggiungere una policy delle risorse al tuo segreto, scegli Modifica delle autorizzazioni. Per ulteriori informazioni, consulta the section called “Allegare una policy di autorizzazione a un segreto”.

   d. (Facoltativo) In Replica il segreto, per replicare il tuo segreto a un altro Regione AWS, scegliere Replica il segreto. Puoi replicare il tuo segreto immediatamente o tornare e replicarlo in un secondo momento. Per ulteriori informazioni, consulta Replica un segreto ad altre Regioni.

   e. Seleziona Successivo.

5. (Facoltativo) Nella pagina Configure rotation (Configura la rotazione), puoi attivare la rotazione automatica. Puoi anche disattivare la rotazione e poi riattivarla in un secondo momento. Per ulteriori informazioni, consulta Rotazione dei segreti . Seleziona Successivo.

6. Nella pagina Review (Revisione), rivedi i dettagli dei segreti e quindi scegli Store (Archivia).

   Secrets Manager ritorna all'elenco dei segreti. Se il segreto nuovo non viene visualizzato, scegli il pulsante aggiorna.

AWS CLI

Quando immetti i comandi in una shell dei comandi, c'è il rischio che la cronologia dei comandi sia accessibile o che le utilità abbiano accesso ai parametri dei comandi. Per informazioni, consultare the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

Example Creazione di un segreto

L'esempio di `create-secret` seguente mostra come creare un segreto con due coppie chiave-valore.

```
aws secretsmanager create-secret \
```

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

Example Creazione di un segreto dalle credenziali in un file JSON

L'esempio di `create-secret` seguente mostra come creare un segreto partendo dalle credenziali in un file. Per ulteriori informazioni, consulta la sezione Caricamento di parametri della AWS CLI da un file nella Guida per l'utente della AWS CLI.

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

Contenuti di mycreds.json:

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

AWS SDK

Per creare un segreto utilizzando uno dei AWS SDK, utilizzare l'`CreateSecret` operazione. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Aggiorna il valore di un segreto AWS Secrets Manager

Per aggiornare il valore del tuo segreto, puoi utilizzare la console, la CLI o un SDK. Quando aggiorni il valore del segreto, Secrets Manager crea una nuova versione del segreto con l’etichetta temporanea `AWSCURRENT`. Puoi ancora accedere alla versione precedente con l'etichetta `AWSPREVIOUS`. Puoi anche aggiungere le tue etichette. Per ulteriori informazioni, consulta Controllo delle versioni di Secrets Manager.

Per aggiornare il valore segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Dall'elenco dei segreti, scegli il segreto.

3. Nella pagina dei dettagli del segreto, nella sezione Panoramica in Valore del segreto, scegli Recupera il valore del segreto, quindi scegli Modifica.

AWS CLI

Per aggiornare il valore del segreto (AWS CLI)

- Quando immetti i comandi in una shell dei comandi, c'è il rischio che la cronologia dei comandi sia accessibile o che le utilità abbiano accesso ai parametri dei comandi. Per informazioni, consultare the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

  L'esempio di `put-secret-value` seguente mostra come creare una nuova versione di un segreto con due coppie chiave-valore.

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

  L’esempio di `put-secret-value` mostra come creare una nuova versione di un'etichetta temporanea personalizzata. La nuova versione avrà le etichette `MyLabel` e `AWSCURRENT`.

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

AWS SDK

Ti consigliamo di evitare di chiamare `PutSecretValue` o `UpdateSecret` ad una velocità sostenuta di più di una volta ogni 10 minuti. Quando chiami `PutSecretValue` o `UpdateSecret` per aggiornare il valore del segreto, Secrets Manager crea una nuova versione del segreto. Secrets Manager rimuove le versioni obsolete quando sono più di 100, ma non rimuove le versioni create da meno di 24 ore. Se aggiorni il valore segreto più di una volta ogni 10 minuti, crei più versioni di quelle che Secrets Manager rimuove e raggiungerai la quota massima prevista per le versioni di un segreto.

Per aggiornare un valore segreto, utilizza le seguenti azioni: `UpdateSecret` o `PutSecretValue`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Modificare la chiave di crittografia per un AWS Secrets Manager segreto

Secrets Manager utilizza la crittografia a busta con AWS KMS chiavi e chiavi dati per proteggere ogni valore segreto. Per ogni segreto, puoi scegliere quale chiave KMS usare. Puoi usare Chiave gestita da AWS aws/secretsmanager oppure puoi usare una chiave gestita dal cliente. Nella maggior parte dei casi, consigliamo di usare aws/secretsmanager, che non presenta costi aggiuntivi per l'utilizzo. Se devi accedere al segreto da un altro Account AWS o se desideri utilizzare la tua chiave KMS in modo da poterla ruotare o applicare una politica chiave, usa a. chiave gestita dal clienteÈ necessario avere le the section called “Autorizzazioni per la chiave KMS”. Per informazioni sui costi di utilizzo di una chiave gestita dal cliente, consulta la sezione Prezzi.

Puoi modificare la chiave di crittografia per il segreto. Ad esempio, se desideri accedere al segreto da un altro account e il segreto è attualmente crittografato utilizzando la chiave AWS gestita`aws/secretsmanager`, puoi passare a un. chiave gestita dal cliente

> **Tip**
>
> Se desideri ruotare il tuo chiave gestita dal cliente, ti consigliamo di utilizzare la rotazione AWS KMS automatica dei tasti. Per ulteriori informazioni, consulta Rotazione AWS KMS dei tasti.

Quando si modifica la chiave di crittografia, Secrets Manager cripta nuovamente `AWSCURRENT` e `AWSPENDING` le `AWSPREVIOUS` versioni con la nuova chiave. Per evitare di nasconderti il segreto, Secrets Manager mantiene tutte le versioni esistenti crittografate con la chiave precedente. Ciò significa che è possibile decrittografare `AWSCURRENT` `AWSPENDING` le `AWSPREVIOUS` versioni con la chiave precedente o quella nuova.

Per fare in modo che `AWSCURRENT` possa essere decrittografato solo con la nuova chiave di crittografia, crea una nuova versione del segreto con la nuova chiave. Quindi, per poter decifrare la versione `AWSCURRENT` segreta, devi avere l'autorizzazione per la nuova chiave.

Se disattivi la chiave di crittografia precedente, non potrai decrittografare nessuna versione segreta ad eccezione di `AWSCURRENT`, `AWSPENDING` e `AWSPREVIOUS`. Se disponi di altre versioni segrete etichettate per le quali desideri mantenere l'accesso, devi ricreare tali versioni con la nuova chiave di crittografia utilizzando the section called “AWS CLI”.

Per modificare la chiave di crittografia per un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Dall'elenco dei segreti, scegli il segreto.
3. Nella pagina dei dettagli del segreto, nella sezione Secrets details (Dettagli segreti), scegli Actions (Operazioni), quindi scegli Edit encryption key (Modifica chiave di crittografia).

AWS CLI

Se modifichi la chiave di crittografia per un segreto e disattivi la chiave di crittografia precedente, non sarà possibile decrittografare nessuna versione segreta ad eccezione di `AWSCURRENT`, `AWSPENDING` e `AWSPREVIOUS`. Se disponi di altre versioni segrete etichettate per le quali desideri mantenere l'accesso, devi ricreare tali versioni con la nuova chiave di crittografia utilizzando the section called “AWS CLI”.

Per modificare la chiave di crittografia per un segreto (AWS CLI)

1. L'esempio di `update-secret` seguente mostra come aggiornare la chiave KMS utilizzata per crittografare il valore del segreto. La chiave KMS deve trovarsi nella stessa Regione del segreto.

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

2. (Facoltativo) Se disponi di versioni segrete con etichette personalizzate, per poterle crittografare nuovamente utilizzando la nuova chiave è necessario ricrearle.

   Quando immetti i comandi in una shell dei comandi, c'è il rischio che la cronologia dei comandi sia accessibile o che le utilità abbiano accesso ai parametri dei comandi. Per informazioni, consulta the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

   a. Ottieni il valore della versione segreta.

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

Annota il valore segreto.

b. Crea una nuova versione con quel valore.

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

Modificare un segreto AWS Secrets Manager

Puoi modificare i metadati di un segreto dopo che è stato creato in base a chi ha creato il segreto. Per i segreti creati da altri servizi, potrebbe essere necessario utilizzare l'altro servizio per aggiornarlo o ruotarlo.

Per determinare chi gestisce un segreto, puoi rivedere il nome del segreto. I segreti gestiti da altri servizi sono preceduti dall'ID di quel servizio. Oppure, nella AWS CLI esegui descrive-secret e rivedi il campo `OwningService`. Per ulteriori informazioni, consulta Segreti gestiti da altri servizi.

Per i segreti da te gestiti, puoi modificare la descrizione, la policy basata sulle risorse, la chiave di crittografia e i tag. Puoi anche modificare il valore delle informazioni crittografate del segreto, sebbene sia consigliabile utilizzare la rotazione per aggiornare i valori del segreto che contengono credenziali. La rotazione aggiorna sia il segreto in Secrets Manager che le credenziali del database o del servizio. Questo consente di mantenere i segreti sincronizzati automaticamente in modo che quando i client richiedono un valore del segreto, recuperano sempre un set di credenziali funzionante. Per ulteriori informazioni, consulta Rotazione dei segreti .

Secrets Manager genera una voce del file di log di CloudTrail quando modifichi un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Aggiornamento di un segreto da te gestito (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Dall'elenco dei segreti, scegli il segreto.
3. Nella pagina dei dettagli del segreto, completa una delle seguenti operazioni:

   Tieni presente che non puoi modificare il nome o l'ARN di un segreto.

- Per aggiornare la descrizione, nella sezione Secrets details (Dettagli segreti) scegli Actions (Operazioni), quindi scegli Edit description (Modifica descrizione).
- Per aggiornare la chiave di crittografia, consulta the section called “Modifica la chiave di crittografia per un segreto”.
- Per aggiornare i tag, nella sezione Tag, scegli Modifica tag. Per informazioni, consultare the section called “Tag segreti ”.
- Per aggiornare il valore del segreto, consulta the section called “Aggiorna un valore segreto”.
- Per aggiornare le autorizzazioni per il segreto, nella sezione Panoramica scegli Modifica autorizzazioni. Per informazioni, consultare the section called “Allegare una policy di autorizzazione a un segreto”.

- Per aggiornare la rotazione per il segreto, nella sezione Rotazione scegli Modifica rotazione. Per informazioni, consultare Rotazione dei segreti .
- Per replicare il tuo segreto in altre regioni, consulta Replica un segreto ad altre Regioni.
- Se il tuo segreto contiene repliche, puoi modificare la chiave di crittografia per una replica. Nella sezione Replica, seleziona il pulsante di opzione per la replica e quindi dal menu Operazioni, scegli Modifica la chiave di crittografia. Per informazioni, consultare the section called “Crittografia e decrittografia del segreto”.
- Per modificare un segreto in modo che sia gestito da un altro servizio, è necessario ricrearlo in tale servizio. Per informazioni, consultare Segreti gestiti da altri servizi.

AWS CLI

Example Aggiornamento della descrizione di un segreto

L'esempio di `update-secret` seguente mostra come aggiornare la descrizione di un segreto.

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

AWS SDK

Ti consigliamo di evitare di chiamare `PutSecretValue` o `UpdateSecret` ad una velocità sostenuta di più di una volta ogni 10 minuti. Quando chiami `PutSecretValue` o `UpdateSecret` per

aggiornare il valore del segreto, Secrets Manager crea una nuova versione del segreto. Secrets Manager rimuove le versioni obsolete quando sono più di 100, ma non rimuove le versioni create da meno di 24 ore. Se aggiorni il valore segreto più di una volta ogni 10 minuti, crei più versioni di quelle che Secrets Manager rimuove e raggiungerai la quota massima prevista per le versioni di un segreto.

Per aggiornare un segreto, utilizza le seguenti azioni: `UpdateSecret` o `ReplicateSecretToRegions`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Scopri i segreti in AWS Secrets Manager

Quando cerchi segreti senza filtro, Secrets Manager fa corrispondere le parole chiave nel nome del segreto, nella descrizione, nella chiave tag e nel valore del tag. La ricerca senza filtri non fa distinzione tra maiuscole e minuscole e ignora caratteri speciali, come spazio, /, _, =, # e utilizza solo numeri e lettere. Quando esegui una ricerca senza filtri, Secrets Manager analizza la stringa di ricerca per convertirla in parole separate. Le parole vengono separate in base a qualsiasi cambiamento da lettera maiuscola a lettera minuscola, da lettera a numero o da numero/lettera a punteggiatura. Ad esempio, inserendo il termine di ricerca `credsDatabase#892`, viene effettuata la ricerca di `creds`, `Database` e 892 in nome, descrizione e chiave e valore di tag.

Secrets Manager genera una voce del file di log di CloudTrail quando si elencano i segreti. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Puoi applicare i seguenti filtri alla ricerca:

Nome

Corrisponde all'inizio dei nomi dei segreti; distinzione tra maiuscole e minuscole. Ad esempio: Nome: **`Data`** restituisce un segreto chiamato `DatabaseSecret`, ma non `databaseSecret` o `MyData`.

Descrizione

Corrisponde alle parole nelle descrizioni segrete, nessuna distinzione tra lettere maiuscole e minuscole. Ad esempio:Descrizione: **`My Description`** abbina i segreti con le seguenti descrizioni:

- `My Description`
- `my description`
- `My basic description`

- `Description of my secret`

Servizio di proprietà

Corrisponde all'inizio dei prefissi ID del servizio, nessuna distinzione tra lettere maiuscole e minuscole. Ad esempio, **`my-ser`** abbina i segreti gestiti dai servizi al prefisso `my-serv` e `my-service`. Per ulteriori informazioni, consulta Segreti gestiti da altri servizi.

Segreti replicati

Puoi filtrare i segreti principali, di replica o quelli che non vengono replicati.

Tasti Tag

Corrisponde all'inizio dei tasti tag; distinzione tra maiuscole e minuscole. Ad esempio: Tasto tag: **`Prod`** restituisce segreti con il tag `Production` e `Prod1`, ma non segreti con il tag `prod` o `1 Prod`.

Valori Tag

Corrisponde all'inizio dei valori dei tag; distinzione tra maiuscole e minuscole. Ad esempio: Valore tag: **`Prod`** restituisce segreti con il tag `Production` e `Prod1`, ma non segreti con il valore del tag `prod` o `1 Prod`.

Secrets Manager è un servizio regionale e restituisce solo i segreti entro la regione selezionata.

AWS CLI

Example Elencazione dei segreti nell'account

L'esempio di `list-secrets` seguente mostra come ottenere un elenco dei segreti del proprio account.

```
aws secretsmanager list-secrets
```

Example Filtraggio dell'elenco dei segreti nell'account

L'esempio di `list-secrets` seguente mostra come ottenere un elenco dei segreti del proprio account che contengono Test nel nome. Il filtro per nome fa distinzione tra maiuscole e minuscole.

```
aws secretsmanager list-secrets \
```

```
    --filter Key="name",Values="Test"
```

Example Ricerca di segreti gestiti da altri servizi AWS

L'esempio `list-secrets` seguente ottiene un elenco di segreti gestiti da un servizio. Si specifica il servizio in base all'ID. Per ulteriori informazioni, consulta Segreti gestiti da altri servizi.

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

AWS SDK

Per trovare segreti usando uno dei SDK AWS, utilizza `ListSecrets`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Eliminare un segreto AWS Secrets Manager

A causa della natura critica dei segreti, AWS Secrets Manager intenzionalmente rende complicata l'eliminazione di un segreto. Secrets Manager non elimina immediatamente i segreti. Invece, Secrets Manager rende immediatamente inaccessibili i segreti e li pianifica per l'eliminazione dopo un intervallo di recupero di un minimo di sette giorni. Fino al termine dell'intervallo di recupero, puoi recuperare un segreto eliminato in precedenza. Non è previsto alcun costo per i segreti contrassegnati per l'eliminazione.

Non è possibile eliminare un segreto primario se viene replicato in altre regioni. Eliminare prima le repliche di lettura, poi eliminare il segreto primario. Quando si elimina una replica, questa viene eliminata immediatamente.

Non è possibile eliminare direttamente una versione di un segreto. Tuttavia, è possibile rimuovere tutte le etichette temporanee dalla versione utilizzando la AWS CLI o AWS SDK. In questo modo, la versione viene contrassegnata come obsoleta e Secrets Manager può eliminarla automaticamente in background.

Se non sai se un segreto è ancora utilizzato da un'applicazione, puoi creare un allarme Amazon CloudWatch che ti avvisi di eventuali tentativi di accesso a un segreto durante la finestra di recupero. Per ulteriori informazioni, consulta Monitoraggio dei segreti AWS Secrets Manager pianificati per l'eliminazione utilizzando Amazon CloudWatch.

Per eliminare un segreto, devi disporre di `secretsmanager:ListSecrets` e `secretsmanager:DeleteSecret`autorizzazioni.

Secrets Manager genera una voce nel file di log di CloudTrail quando si elimina un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Come aggiornare un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nell'elenco di segreti, scegli il segreto che desideri eliminare.
3. Nella sezione Secrets details (Dettagli Segreti) scegli Actions (Operazioni), quindi scegli Edit description (Modifica descrizione).
4. Nella finestra di dialogo Disabilitare l'eliminazione segreta e pianificare, in Periodo di attesa, inserisci il numero di giorni di attesa prima che l'eliminazione diventi definitiva. Secrets Manager collega un campo denominato `DeletionDate` e lo imposta sulla data e sull'ora correnti, a cui somma il numero di giorni specificati per l'intervallo di recupero.
5. Scegliere Schedule deletion (Pianifica eliminazione).

Come visualizzare i segreti eliminati

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Sulla pagina Segreti, seleziona Preferenze ( ).
3. Nella finestra di dialogo Preferences (Preferenze) seleziona Show secrets scheduled for deletion (Mostra segreti pianificati per la cancellazione) e quindi scegli Save (Salva).

Per eliminare un segreto di replica

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli il segreto primario.
3. Nella sezione Replica di segreto, scegli il segreto di replica.
4. Dal menu Actions (Operazioni), scegliere Delete Replica (Elimina replica).

AWS CLI

Example Eliminare un segreto

L'esempio di `delete-secret` seguente mostra come eliminare un segreto. È possibile recuperare il segreto con `restore-secret` fino alla data e all'ora presenti nel campo di risposta DeletionDate. Per eliminare un segreto replicato in altre regioni, è necessario dapprima rimuovere le relative repliche con `remove-regions-from-replication`, quindi chiamare `delete-secret`.

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

Example Eliminazione immediata di un segreto

L'esempio di `delete-secret` seguente mostra come eliminare immediatamente il secreto senza un intervallo di recupero. Non è possibile recuperare questo segreto.

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

Example Eliminazione di un segreto di replica

L'esempio di `remove-regions-from-replication` seguente mostra come eliminare un segreto di replica nella Regione eu-west-3. Per eliminare un segreto primario replicato in altre Regioni, è necessario dapprima eliminare le relative repliche e poi chiamare `delete-secret`.

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

AWS SDK

Per eliminare un segreto, utilizza il comando `DeleteSecret`. Per eliminare la versione di un segreto, usa il comando `UpdateSecretVersionStage`. Per eliminare una replica, utilizza il comando `StopReplicationToReplica`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Ripristino di un segreto AWS Secrets Manager

Secrets Manager considera un segreto pianificato per l'eliminazione obsoleto e non più accessibile direttamente. Una volta trascorso l'intervallo di recupero, Secrets Manager elimina il segreto definitivamente. Dopo che Secrets Manager ha eliminato il segreto, non è possibile recuperarlo. Prima della fine dell'intervallo di recupero, puoi recuperare il segreto e renderlo nuovamente accessibile. Il campo `DeletionDate` viene rimosso e l'eliminazione permanente pianificata viene annullata.

Per ripristinare un segreto e i metadati nella console devi disporre di `secretsmanager:ListSecrets` e `secretsmanager:RestoreSecret` autorizzazioni.

Secrets Manager genera una voce del file di log di CloudTrail quando ripristini un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Come ripristinare un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nell'elenco di segreti, scegli il segreto che desideri modificare.

   Se i segreti eliminati non vengono visualizzati nell'elenco dei segreti, scegli Preferenze ( ).
   Nella finestra di dialogo Preferences (Preferenze) seleziona Show secrets scheduled for deletion (Mostra segreti pianificati per l'eliminazione) e quindi scegli Save (Salva)
3. Nella sezione Secret details (Dettagli segreto) scegli Cancel deletion (Annulla eliminazione).
4. Nella finestra di dialogo Cancel secret deletion (Annulla eliminazione segreto) scegli Cancel deletion (Annulla eliminazione).

AWS CLI

Example Ripristino di un segreto precedentemente eliminato

L'esempio di `restore-secret` seguente mostra il ripristino di un segreto per il quale in precedenza era stata pianificata l'eliminazione.

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

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

AWS SDK

Per ripristinare un segreto contrassegnato per l'eliminazione, utilizzare il comando `RestoreSecret`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Replica di un segreto AWS Secrets Manager per altri Regioni AWS

Puoi replicare i tuoi segreti in più Regioni AWS per supportare le applicazioni distribuite in queste Regioni per soddisfare i requisiti di accesso regionale e bassa latenza. Se necessario, è possibile promuovere un segreto di replica in uno autonomo e quindi configurarlo per la replica in modo indipendente. Secrets Manager replica i dati segreti crittografati e i metadati, ad esempio tag e policy delle risorse tra le Regioni specificate.

L'ARN di un segreto replicato è lo stesso del segreto principale ad eccezione della regione, ad esempio:

- Segreto primario: `arn:aws:secretsmanager:Region1:123456789012:secret:MySecret-a1b2c3`
- Segreto di replica: `arn:aws:secretsmanager:Region2:123456789012:secret:MySecret-a1b2c3`

Per informazioni sui prezzi dei segreti di replica, consulta Prezzi di AWS Secrets Manager.

Quando archivi le credenziali del database per un database di origine replicato in altre regioni, il segreto contiene informazioni di connessione per il database di origine. Se poi replichi il segreto, le repliche saranno copie del segreto di origine e conterranno le stesse informazioni di connessione. Puoi aggiungere altre coppie chiave/valore al segreto per le informazioni sulla connessione della regione.

Se si attiva il segreto primario per la rotazione, Secrets Manager esegue la rotazione segreta nella regione principale e il nuovo valore segreto si propaga a tutti i segreti di replica associati. Non è possibile gestire la rotazione singolarmente per tutti i segreti di replica.

È possibile replicare i segreti in tutte le AWS Regioni abilitate. Tuttavia, se si utilizza Secrets Manager in specifiche AWS Regioni come AWS GovCloud (US) o regioni della Cina, è possibile configurare

solo i segreti e le repliche all'interno di queste specializzate AWSRegioni. Non è possibile replicare un segreto nelle Regioni abilitate AWS a una regione specializzata o replicare segreti da una regione specializzata a una regione commerciale.

Prima di poter replicare un segreto in un'altra regione, devi abilitare tale regione. Per ulteriori informazioni, consulta Gestire AWS Regioni.

Puoi utilizzare un segreto in più regioni senza replicarlo chiamando l'endpoint di Secrets Manager nella regione in cui è archiviato il segreto. Per un elenco di endpoint, consulta the section called "Endpoint di Secrets Manager". Per utilizzare la replica per migliorare la resilienza del carico di lavoro, consulta Architettura del ripristino di emergenza (DR) su AWS, Parte I: Strategie per il ripristino nel cloud.

Secrets Manager genera una voce di CloudTrail registro quando si replica un segreto. Per ulteriori informazioni, consulta the section called "Accedi conAWS CloudTrail ".

Come replicare un segreto in altre regioni (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Dall'elenco dei segreti, scegli il segreto.
3. Nella pagina dei dettagli del segreto, nella sezione Replica completa una delle seguenti operazioni:
   - Se il tuo segreto non viene replicato, scegli Replica il segreto.
   - Se il tuo segreto viene replicato, nella sezione Replica il segreto, scegli Aggiungi regione.
4. Nella finestra di dialogo Aggiungi valore di registro, effettua una delle operazioni indicate di seguito:

   a. Per Regione AWS, scegli la regione in cui desideri replicare il segreto.

   b. (Opzionale) In Chiave di crittografia, scegli una chiave KMS con cui crittografare il segreto. La chiave deve trovarsi nella Regione della replica.

   c. (Facoltativo) Per aggiungere un'altra regione, scegli Aggiungi altre regioni.

   d. Scegliere Replicate (Replica).

   Torna alla pagina dei dettagli del segreto. Nella sezione Replica segreto, lo Stato di replica viene visualizzato per ogni regione.

AWS CLI

Example Replica di un segreto in un'altra Regione

Nell'esempio `replicate-secret-to-regions` seguente, un segreto viene replicato nella Regione eu-west-3. La replica è crittografata con la chiave gestita AWS aws/secretsmanager.

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

SDK AWS

Per replicare un segreto, utilizzare il comando `ReplicateSecretToRegions`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Risoluzione dei problemi

Di seguito sono riportati alcuni dei motivi per cui la replica può fallire.

Esiste un segreto con lo stesso nome nella Regione selezionata

Per risolvere questo problema è possibile sovrascrivere il nome duplicato del segreto nella Regione di replica. Riprova la replica e poi nella casella di dialogo Riprova la replica scegli Sovrascrivi.

Nessuna autorizzazione disponibile sulla chiave KMS per completare la replica

Secrets Manager innanzitutto crittografa il segreto prima di crittografarlo nuovamente con la nuova chiave KMS nella Regione di replica. Se non disponi dell'autorizzazione `kms:Decrypt` per la chiave di crittografia nella Regione principale, riscontrerai questo errore. Per crittografare il segreto replicato con una chiave KMS diversa da `aws/secretsmanager`, è necessario `kms:GenerateDataKey` e `kms:Encrypt` alla chiave. Per informazioni, consulta the section called “Autorizzazioni per la chiave KMS”.

La chiave KMS è stata disattivata o non è stata trovata

Se la chiave di crittografia nella regione principale è disabilitata o eliminata, Secrets Manager non può replicare il segreto. Se il segreto presenta versioni con etichetta personalizzata crittografate con la chiave di crittografia disabilitata o eliminata, questo errore può verificarsi anche se la chiave

di crittografia è stata modificata. Per informazioni su come Secrets Manager esegue la crittografia, consulta the section called “Crittografia e decrittografia del segreto”. Per risolvere questo problema, è possibile ricreare le versioni segrete in modo che Secrets Manager possa crittografarle con la chiave di crittografia corrente. Per ulteriori informazioni, consulta Modificare la chiave di crittografia per un segreto. Quindi riprova la replica.

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

Non è stata abilitata la Regione in cui si verifica la replica

Per informazioni su come abilitare una Regione, consulta Gestione delle Regioni AWS nella Guida di riferimento alla gestione degli account AWS.

Promozione di un segreto di replica a segreto autonomo in AWS Secrets Manager

Un segreto di replica è un segreto replicato da un primario in un altro Regione AWS. Ha lo stesso valore segreto e metadati del primario, ma può essere crittografato con una chiave KMS diversa. Un segreto di replica non può essere aggiornato indipendentemente dal segreto principale, con l'eccezione della chiave di crittografia. La promozione di un segreto di replica disconnette tale segreto da quello primario e rende la replica segreta un segreto autonomo. Modifiche al segreto primario non vengono replicate nel segreto autonomo.

Puoi promuovere un segreto di replica a un segreto autonomo come soluzione di ripristino di emergenza in caso di errore del segreto primario. In alternativa, è utile promuovere una replica in un segreto autonomo se si desidera attivare la rotazione per la replica.

Se si promuove una replica, per utilizzare il segreto autonomo, assicurati di aggiornare le applicazioni corrispondenti.

Secrets Manager genera una voce del file di log di CloudTrail quando promuovi un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Per promuovere un segreto di replica (console)

1. Accedi a Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.

2. Accesso alla Regione di replica.
3. Nella pagina Secrets (Segreti), scegli il segreto di replica.
4. Nella pagina dei dettagli dei segreti di replica, scegli Promote to standalone secret (Promuovi a segreto autonomo).
5. Nella finestra di dialogo Promote replica to standalone secret (Promuovi replica a segreto autonomo), inserisci la Regione e quindi scegli Promote replica (Promuovi replica).

AWS CLI

Example Promozione di un segreto di replica a primario

L'esempio di `stop-replication-to-replica` seguente mostra come rimuovere il collegamento tra un segreto di replica e quello primario. Il segreto di replica viene promosso a segreto primario nella Regione della replica. È necessario effettuare una chiamata `stop-replication-to-replica` dall'interno della Regione della replica.

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

AWS SDK

Per promuovere una replica in un segreto autonomo, usare il comando `StopReplicationToReplica`. È necessario richiamare questa azione dalla regione segreta di replica. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Tag segreti AWS Secrets Manager

Secrets Manager definisce un tag come un'etichetta che consiste di una chiave definita e un valore facoltativo. Puoi utilizzare i tag per rendere più facile la gestione, ricerca e filtro dei segreti e delle altre risorse nel tuo account AWS. Quando utilizzi tag per i tuoi segreti, utilizza uno schema di denominazione standard per tutte le risorse. Per ulteriori informazioni, consulta il whitepaper Best practice relative al tagging.

È possibile concedere o negare l'accesso a un segreto controllando i tag collegati al segreto. Per ulteriori informazioni, consulta the section called “Esempio: Controllare l'accesso ai segreti utilizzando i tag”.

I segreti sono disponibili per tag nella console, nella AWS CLI, e negli SDK. AWS inoltre fornisce lo strumento Resource Groups per creare una console personalizzata che consolida e organizza le risorse in base ai tag. Per individuare i segreti con un tag specifico, consulta the section called "Scopri i segreti". Gestione dei segreti non supporta l'allocazione dei costi basata su tag.

Non archiviare le informazioni sensibili di un segreto in un tag.

Per le quote dei tag e le restrizioni sulla denominazione, consulta Quote di servizio per il tagging nella Guida di riferimento generale per AWS. I tag rispettano la distinzione tra maiuscole e minuscole.

Secrets Manager genera una voce del file di log di CloudTrail quando si tagga o si rimuove un segreto. Per ulteriori informazioni, consulta the section called "Accedi conAWS CloudTrail ".

Come modificare i tag per il segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Dall'elenco dei segreti, scegli il segreto.
3. Nella pagina dei dettagli del segreto, nella sezione Tag, scegli Modifica tag. I nomi e i valori delle chiavi dei tag richiedono una distinzione tra lettere maiuscole e minuscole e i tag delle chiavi devono essere unici.

AWS CLI

Example Aggiunta di un tag a un segreto

L'esempio di `tag-resource` seguente mostra come collegare un tag con una sintassi abbreviata.

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

Example Aggiunta di più tag a un segreto

L'esempio di `tag-resource` seguente mostra come collegare due tag chiave-valore a un segreto.

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

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

Example Rimozione di tag da un segreto

L'esempio di `untag-resource` seguente mostra come rimuovere due tag da un segreto. Per ogni tag, vengono rimossi sia la chiave che il valore.

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

AWS SDK

Per modificare i tag per il segreto, usa `TagResource` o `UntagResource`. Per ulteriori informazioni, consulta the section called “AWS SDK”.

Recuperare segreti da AWS Secrets Manager

Puoi recuperare i tuoi segreti:

- In codice
- In altri servizi
- In AWS CLI
- Nella console AWS

Secrets Manager genera una voce del file di log di CloudTrail quando recuperi un segreto. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

In codice

In applicazioni, puoi recuperare i segreti chiamando `GetSecretValue` o `BatchGetSecretValue` in qualsiasi SDK AWS . Per esempi, consulta Ottieni un valore segreto nella Libreria di esempi di codice SDK di AWS. Tuttavia, ti consigliamo di memorizzare nella cache i valori del segreto utilizzando la caching lato client. Memorizzare i segreti nella cache migliora la velocità e riduce i costi.

- Per applicazioni Java:
  - Se memorizzi le credenziali del database nel segreto, utilizza i driver di connessione SQL di Secrets Manager per eseguire la connessione a un database utilizzando le credenziali nel segreto.
  - Per altri tipi di segreti, utilizza il componente di caching basato su Java di Secrets Manager o chiama direttamente l'SDK con `GetSecretValue`.
- Per le applicazioni Python, utilizza il componente di caching basato su Python di Secrets Manager o chiama direttamente l'SDK con `get_secret_value` o `batch_get_secret_value`.
- Per le applicazioni .NET, utilizza il componente di caching basato su .NET di Secrets Manager o chiama direttamente l'SDK con `GetSecretValue` o `BatchGetSecretValue`.
- Per le applicazioni Go, utilizza il componente di caching basato su Go di Secrets Manager o chiama direttamente l'SDK con `GetSecretValue` o `BatchGetSecretValue`.
- Per le applicazioni JavaScript, effettua direttamente una chiamata all'SDK con `getSecretValue` o `batchGetSecretValue`.

- Per le applicazioni PHP, effettua direttamente una chiamata all'SDK con `GetSecretValue` o `BatchGetSecretValue`.
- Per le applicazioni Ruby, effettua direttamente una chiamata all'SDK con `get_secret_value` o `batch_get_secret_value`.
- Per le operazioni GitHub, consulta the section called “GitHub lavori”.

All'interno di altri sistemi e servizi AWS

Puoi anche recuperare i segreti nei seguenti ambiti:

- Per AWS Batch, puoi fare riferimento ai segreti in una definizione del processo.
- Per AWS CloudFormation, puoi creare segreti e fare riferimento a essi in una pila CloudFormation.
- Per Amazon ECS, puoi fare riferimento ai segreti in una definizione del container.
- Per Amazon EKS, puoi utilizzare AWS Secrets and Configuration Provider (ASCP) per montare i segreti come file in Amazon EKS.
- Per GitHub, puoi usare l'operazione Gestione dei segreti per GitHub per aggiungere segreti come variabili di ambiente nei tuoi processi GitHub.
- Per AWS IoT Greengrass, puoi fare riferimento ai segreti in un gruppo Greengrass.
- Per AWS Lambda, puoi fare riferimento ai segreti in una funzione Lambda.
- Per Parameter Store, puoi fare riferimento ai segreti in un parametro.

AWS CLI

Example Recupero del valore del segreto crittografato di un segreto

L'esempio di `get-secret-value` seguente mostra come recuperare il valore corrente del segreto.

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

Example Recupero del valore del segreto precedente

L'esempio di `get-secret-value` seguente mostra come recuperare il valore precedente del segreto.

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

Console AWS

Per recuperare un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nell'elenco di segreti, scegli il segreto che desideri recuperare.
3. Nella sezione Secret value (Valore segreto) scegli Retrieve secret value (Recupera valore segreto).

   Secrets Manager visualizza la versione corrente (`AWSCURRENT`) del segreto. Per visualizzare altre versioni del segreto, ad esempio `AWSPREVIOUS` versioni con etichetta personalizzata, usa the section called “AWS CLI”.

Recupera un gruppo di segreti in un batch da AWS Secrets Manager

Secrets Manager offre l'API batch `BatchGetSecretValue` per recuperare un gruppo di segreti in un'unica chiamata API. Per scegliere quali segreti recuperare, puoi specificare un elenco di segreti per nome o ARN oppure puoi utilizzare i filtri. Se Secrets Manager riscontra errori, ad esempio `AccessDeniedException` durante il tentativo di recupero di uno dei segreti, puoi visualizzare gli errori nella risposta `Errors`.

Autorizzazioni per il recupero di segreti in un batch

È necessario disporre dell'autorizzazione `secretsmanager:GetSecretValue` per ogni segreto che desideri recuperare. Devi anche disporre dell'autorizzazione `secretsmanager:BatchGetSecretValue`. Se usi i filtri, devi avere anche `secretsmanager:ListSecrets`. Per un esempio di policy delle autorizzazioni, consulta the section called “Autorizzazione per il recupero di un gruppo di valori segreti in un batch”.

> **Important**
>
> Se hai una policy VPCE che nega l'autorizzazione per recuperare un singolo segreto nel gruppo su cui stai agendo, `BatchGetSecretValue` non restituirà alcun valore segreto e restituirà un errore.

AWS CLI

Example Recupera il valore segreto per un gruppo di segreti elencati per nome

L'esempio `batch-get-secret-value` seguente mostra come recuperare il valore corrente del segreto.

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

Example Recupera il valore segreto per un gruppo di segreti selezionati dal filtro

L'esempio `batch-get-secret-value` seguente ottiene il valore segreto per i segreti che hanno un tag denominato "Test".

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

Connettersi a un database SQL con credenziali in un segreto AWS Secrets Manager

Nelle applicazioni Java, è possibile utilizzare i driver Secrets Manager SQL Connection per connettersi ai database MySQL, PostgreSQL, Oracle, MSSQLServer, Db2 e Redshift utilizzando le credenziali archiviate in Secrets Manager. Ogni driver esegue il wrapping del driver JDBC di base per consentire l'utilizzo delle chiamate JDBC per accedere al database. Tuttavia, invece di specificare un nome utente e una password per la connessione, si fornisce l'ID di un segreto. Il driver chiama Secrets Manager per recuperare il valore del segreto, quindi utilizza le credenziali nel segreto per connettersi al database. Il driver inoltre memorizza le credenziali nella cache utilizzando la libreria di caching lato client Java, in modo che per le connessioni future non sia necessaria una

chiamata a Secrets Manager. Per impostazione predefinita, la cache viene aggiornata ogni ora e anche quando un segreto viene ruotato. Per configurare la cache, consulta the section called "SecretCacheConfiguration".

È possibile scaricare GitHubil codice sorgente da.

Per utilizzare i driver di connessione SQL di Secrets Manager:

- L'applicazione deve essere in Java 8 o versioni successive.
- Il segreto deve essere uno fra i seguenti:
  - Un segreto di database nella struttura JSON prevista. Per verificare il formato, nella console di Secrets Manager, visualizza il tuo segreto e seleziona Retrieve secret value (Recupera valore segreto). In alternativa AWS CLI, nella chiamata get-secret-value.
  - Un segreto gestito da Amazon RDS. Per questo tipo di segreto, quando si stabilisce la connessione è necessario specificare un endpoint e una porta.
  - Un segreto gestito da Amazon Redshift. Per questo tipo di segreto, quando si stabilisce la connessione è necessario specificare un endpoint e una porta.

Se il database viene replicato in altre regioni, per connettersi a un database di replica in una regione differente, è necessario specificare l'endpoint e la porta della regione al momento della creazione della connessione. Puoi archiviare le informazioni sulla connessione della regione nel segreto come coppie chiave/valore aggiuntive, nei parametri dell'Archivio parametri SSM o nella configurazione del codice.

Per aggiungere il driver al progetto, nel file di build Maven `pom.xml`, aggiungi la seguente dipendenza per il driver. Per ulteriori informazioni, consulta Secrets Manager SQL Connection Library sul sito Web di Maven Central Repository.

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

Il driver utilizza la catena di provider delle credenziali predefinita. Se esegui il driver su Amazon EKS, potrebbe raccogliere le credenziali del nodo su cui è in esecuzione anziché il ruolo dell'account di servizio. Per risolvere questo problema, aggiungi la versione 1 di `com.amazonaws:aws-java-sdk-sts` al tuo file di progetto Gradle o Maven come dipendenza.

Per impostare un URL di endpoint AWS PrivateLink DNS e una regione nel file: `secretsmanager.properties`

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

Per sovrascrivere la regione primaria, imposta la variabile d'ambiente `AWS_SECRET_JDBC_REGION` o apporta la seguente modifica al file `secretsmanager.properties`:

```
drivers.region = region
```

Autorizzazioni richieste:

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

Per ulteriori informazioni, consulta Riferimento per le autorizzazioni.

Esempi:

- Stabilire una connessione a un database
- Impostazione di una connessione specificando l'endpoint e la porta
- Utilizzare il pooling di connessioni c3p0 per stabilire una connessione
- Utilizzo del pooling di connessioni c3p0 per stabilire una connessione specificando l'endpoint e la porta

Stabilire una connessione a un database

Nell'esempio seguente viene illustrato come stabilire una connessione a un database utilizzando le credenziali e le informazioni di connessione in un segreto. Una volta acquisita la connessione, puoi utilizzare le chiamate JDBC per accedere al database. Per ulteriori informazioni, consulta JDBC Basics sul sito Web della documentazione 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);
```

Impostazione di una connessione specificando l'endpoint e la porta

Nell'esempio seguente viene illustrato come stabilire una connessione a un database utilizzando le credenziali in un segreto con un endpoint e una porta specificati dall'utente.

I segreti gestiti da Amazon RDS non includono l'endpoint e la porta del database. Per connettersi a un database utilizzando le credenziali master in un segreto gestito da Amazon RDS, è necessario specificarle nel codice.

I segreti replicati in altre Regioni possono migliorare la latenza per la connessione al database regionale, ma non contengono informazioni di connessione diverse dal segreto di origine. Ogni replica è una copia del segreto di origine. Per archiviare le informazioni sulla connessione regionale nel segreto, aggiungi altre coppie chiave/valore per le informazioni sull'endpoint e sulla porta per le regioni.

Una volta acquisita la connessione, puoi utilizzare le chiamate JDBC per accedere al database. Per ulteriori informazioni, consulta JDBC Basics sul sito Web della documentazione 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);
```

Utilizzare il pooling di connessioni c3p0 per stabilire una connessione

Nell'esempio seguente viene illustrato come stabilire un pool di connessioni con un file `c3p0.properties` che utilizza il driver per recuperare le credenziali e le informazioni sulla connessione dal segreto. Per `user` e `jdbcUrl`, inserisci l'ID segreto per configurare il pool di connessioni. Puoi quindi recuperare le connessioni dal pool e utilizzarle come qualsiasi altra connessione al database. Per ulteriori informazioni, consulta JDBC Basics sul sito Web della documentazione Java.

Per ulteriori informazioni su c3p0, consulta c3p0 sul sito 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
```

Utilizzo del pooling di connessioni c3p0 per stabilire una connessione specificando l'endpoint e la porta

Nell'esempio seguente viene illustrato come stabilire un pool di connessioni con un file `c3p0.properties` che utilizza il driver per recuperare le credenziali in un segreto con un endpoint e una porta specificati dall'utente. Puoi quindi recuperare le connessioni dal pool e utilizzarle come qualsiasi altra connessione al database. Per ulteriori informazioni, consulta JDBC Basics sul sito Web della documentazione Java.

I segreti gestiti da Amazon RDS non includono l'endpoint e la porta del database. Per connettersi a un database utilizzando le credenziali master in un segreto gestito da Amazon RDS, è necessario specificarle nel codice.

I segreti replicati in altre Regioni possono migliorare la latenza per la connessione al database regionale, ma non contengono informazioni di connessione diverse dal segreto di origine. Ogni replica è una copia del segreto di origine. Per archiviare le informazioni sulla connessione regionale nel segreto, aggiungi altre coppie chiave/valore per le informazioni sull'endpoint e sulla porta per le regioni.

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
```

Recuperare segreti AWS Secrets Manager nelle applicazioni Java

Quando si recupera un segreto, è possibile utilizzare il componente di caching basato su Java di Secrets Manager per memorizzarlo nella cache per un uso futuro. Il recupero di un segreto memorizzato nella cache è più veloce rispetto al recupero da Secrets Manager. Poiché è previsto un costo per chiamare le API di Secrets Manager, l'utilizzo di una cache può ridurre i costi. Per tutti i modi in cui puoi recuperare i segreti, vedi Recuperare segreti.

La policy della cache è Least Recently Used (LRU), quindi quando la cache deve eliminare un segreto, elimina il segreto usato meno di recente. Di default, la cache aggiorna i segreti ogni ora. È possibile configurare la frequenza con cui il segreto viene aggiornato nella cache ed è possibile collegarsi al recupero del segreto per aggiungere altre funzionalità.

La cache non impone la rimozione di oggetti inutili (garbage collection) una volta liberati i riferimenti alla cache. L'implementazione della cache non include l'invalidazione della cache. L'implementazione

della cache è incentrata sulla cache stessa e non è rafforzata o focalizzata sulla sicurezza. Se hai bisogno di un livello di sicurezza aggiuntivo, come la crittografia degli elementi nella cache, usa le interfacce e i metodi astratti forniti.

Per usare il componente, devi disporre dei seguenti elementi:

- Ambiente di sviluppo Java 8 o versioni successive. Consulta Java SE Downloads sul sito Web di Oracle.
- SDK AWS 1.x per Java. Nei tuoi progetti puoi utilizzare entrambe le versioni dell'SDK AWS per Java. Per ulteriori informazioni, consulta Using the SDK for Java 1.x and 2.x side-by-side (Utilizzo degli SDK per Java 1.x e 2.x affiancati).

Per scaricare il codice sorgente, consulta Componente di caching basato su Java di Secrets Manager su GitHub.

Per aggiungere il componente al progetto, nel file Maven pom.xml, includi la seguente dipendenza. Per ulteriori informazioni su Maven, consulta la Guida alle operazioni di base sul sito Web Apache Maven Project.

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

Autorizzazioni richieste:

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

Per ulteriori informazioni, consulta Riferimento per le autorizzazioni.

Riferimento

- SecretCache
- SecretCacheConfiguration
- SecretCacheHook

Example Recupero di un segreto

L'esempio di codice riportato di seguito mostra una funzione Lambda che recupera una stringa del segreto. Segue la best practice di creare un'istanza della cache al di fuori del gestore della funzione quindi non continua a chiamare l'API se si chiama nuovamente la funzione Lambda.

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

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

    import com.amazonaws.secretsmanager.caching.SecretCache;

    public class SampleClass implements RequestHandler<String, String> {

         private final SecretCache cache  = new SecretCache();

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

            // Use the secret, return success;

        }
    }
```

SecretCache

Una cache in memoria per i segreti richiesti da Secrets Manager. Si usa the section called “getSecretString” o the section called “getSecretBinary” per recuperare un segreto dalla cache. È possibile configurare le impostazioni della cache specificando un oggetto the section called “SecretCacheConfiguration” nel costruttore.

Per ulteriori informazioni, inclusi esempi, consulta the section called “Applicazioni Java”.

Costruttori

`public SecretCache()`

Costruttore di default per un oggetto `SecretCache`.

```
public SecretCache(AWSSecretsManagerClientBuilder builder)
```

Costruisce una nuova cache utilizzando un client di Secrets Manager creato utilizzando l'`AWSSecretsManagerClientBuilder` fornito. Utilizza questo costruttore per personalizzare il client di Secrets Manager, ad esempio per utilizzare una regione o un endpoint specifico.

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

Costruisce una nuova cache del segreto utilizzando l'`AWSSecretsManagerClient` fornito. Utilizza questo costruttore per personalizzare il client di Secrets Manager, ad esempio per utilizzare una regione o un endpoint specifico.

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

Costruisce una nuova cache del segreto utilizzando il `the section called "SecretCacheConfiguration"` fornito.

Metodi

getSecretString

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

Recupera un segreto stringa da Secrets Manager. Restituisce una `String`.

getSecretBinary

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

Recupera un segreto binario da Secrets Manager. Restituisce un `ByteBuffer`.

refreshNow

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

Forza l'aggiornamento della cache. Restituisce `true` se l'aggiornamento è stato completato senza errori, altrimenti `false`.

close

```
public void close()
```

Chiude la cache.

SecretCacheConfiguration

Opzioni di configurazione della cache per una the section called “SecretCache”, ad esempio dimensione massima della cache e durata (TTL) per i segreti memorizzati nella cache.

Costruttore

```
public SecretCacheConfiguration
```

Costruttore di default per un oggetto `SecretCacheConfiguration`.

Metodi

getClient

```
public AWSSecretsManager getClient()
```

Restituisce l'`AWSSecretsManagerClient` da cui la cache recupera segreti.

setClient

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

Restituisce il client `AWSSecretsManagerClient` da cui la cache recupera segreti.

getCacheHook

```
public SecretCacheHook getCacheHook()
```

Restituisce l'interfaccia the section called “SecretCacheHook” utilizzata per collegarsi agli aggiornamenti della cache.

setCacheHook

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

Imposta l'interfaccia the section called “SecretCacheHook” utilizzata per collegarsi agli aggiornamenti della cache.

getMaxCacheSize

```
public int getMaxCacheSize()
```

Restituisce la dimensione massima della cache. Il valore di default è 1024 segreti.

setMaxCacheSize

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

Imposta la dimensione massima della cache. Il valore di default è 1024 segreti.

getCacheItemTTL

```
public long getCacheItemTTL()
```

Restituisce il TTL in millisecondi per gli elementi memorizzati nella cache. Quando un segreto memorizzato nella cache supera questo TTL, la cache recupera una nuova copia del segreto dal `AWSSecretsManagerClient`. Il valore predefinito è 1 ora in millisecondi.

La cache aggiorna il segreto in modo sincrono quando viene richiesto il segreto dopo il TTL. Se l'aggiornamento sincrono ha esito negativo, la cache restituisce il segreto non aggiornato.

setCacheItemTTL

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

Restituisce il TTL in millisecondi per gli elementi memorizzati nella cache. Quando un segreto memorizzato nella cache supera questo TTL, la cache recupera una nuova copia del segreto dal `AWSSecretsManagerClient`. Il valore predefinito è 1 ora in millisecondi.

getVersionStage

```
public String getVersionStage()
```

Restituisce la versione dei segreti che si desidera memorizzare nella cache. Per ulteriori informazioni, consulta Versioni del segreto. Il valore predefinito è `"AWSCURRENT"`.

setVersionStage

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

Imposta la versione dei segreti che si desidera memorizzare nella cache. Per ulteriori informazioni, consulta Versioni del segreto. Il valore predefinito è `"AWSCURRENT"`.

SecretCacheConfiguration withClient

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

Imposta il `AWSSecretsManagerClient` da cui recuperare segreti. Restituisce l'oggetto `SecretCacheConfiguration` aggiornato con la nuova impostazione.

SecretCacheConfiguration withCacheHook

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

Imposta l'interfaccia utilizzata per collegare la cache in memoria. Restituisce l'oggetto `SecretCacheConfiguration` aggiornato con la nuova impostazione.

SecretCacheConfiguration withMaxCacheSize

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

Imposta la dimensione massima della cache. Restituisce l'oggetto `SecretCacheConfiguration` aggiornato con la nuova impostazione.

SecretCacheConfiguration withCacheItemTTL

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

Restituisce il TTL in millisecondi per gli elementi memorizzati nella cache. Quando un segreto memorizzato nella cache supera questo TTL, la cache recupera una nuova copia del segreto dal `AWSSecretsManagerClient`. Il valore predefinito è 1 ora in millisecondi. Restituisce l'oggetto `SecretCacheConfiguration` aggiornato con la nuova impostazione.

SecretCacheConfiguration withVersionStage

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

Imposta la versione dei segreti che si desidera memorizzare nella cache. Per ulteriori informazioni, consulta Versioni del segreto. Restituisce l'oggetto `SecretCacheConfiguration` aggiornato con la nuova impostazione.

SecretCacheHook

Un'interfaccia per collegarsi a una the section called “SecretCache” per eseguire operazioni sui segreti memorizzati al suo interno.

put

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

Prepara l'oggetto per la memorizzazione nella cache.

Restituisce l'oggetto da memorizzare nella cache.

get

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

Deriva l'oggetto dall'oggetto memorizzato nella cache.

Restituisce l'oggetto da restituire dalla cache

Recupera segreti AWS Secrets Manager nelle applicazioni Python

Quando si recupera un segreto, è possibile utilizzare il componente di caching basato su Python di Secrets Manager per memorizzarlo nella cache per un uso futuro. Il recupero di un segreto memorizzato nella cache è più veloce rispetto al recupero da Secrets Manager. Poiché è previsto un costo per chiamare le API di Secrets Manager, l'utilizzo di una cache può ridurre i costi. Per tutti i modi in cui puoi recuperare i segreti, vedi Recuperare segreti.

La policy della cache è Least Recently Used (LRU), quindi quando la cache deve eliminare un segreto, elimina il segreto usato meno di recente. Di default, la cache aggiorna i segreti ogni ora. È possibile configurare la frequenza con cui il segreto viene aggiornato nella cache ed è possibile collegarsi al recupero del segreto per aggiungere altre funzionalità.

La cache non impone la rimozione di oggetti inutili (garbage collection) una volta liberati i riferimenti alla cache. L'implementazione della cache non include l'invalidazione della cache. L'implementazione della cache è incentrata sulla cache stessa e non è rafforzata o focalizzata sulla sicurezza. Se hai bisogno di un livello di sicurezza aggiuntivo, come la crittografia degli elementi nella cache, usa le interfacce e i metodi astratti forniti.

Per usare il componente, devi disporre dei seguenti elementi:

- Python 3.6 o versioni successive.
- botocore 1.12 o versioni successive. Consulta AWSSDK for Python e Botocore.
- setuptools_scm 3.2 o versioni successive. Consulta https://pypi.org/project/setuptools-scm/.

Per scaricare il codice sorgente, consulta Componente di caching basato su Python di Secrets Manager su GitHub.

Per installare il componente, utilizza il comando seguente.

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

Autorizzazioni richieste:

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

Per ulteriori informazioni, consulta Riferimento per le autorizzazioni.

Riferimento

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

Example Recupero di un segreto

Gli esempi seguenti mostrano come ottenere il valore di un segreto per un segreto denominato *`mysecret`*.

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

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

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

SecretCache

Una cache in memoria per i segreti recuperati da Secrets Manager. Si usa the section called "get_secret_string" o the section called "get_secret_binary" per recuperare un segreto dalla cache. È possibile configurare le impostazioni della cache specificando un oggetto the section called "SecretCacheConfig" nel costruttore.

Per ulteriori informazioni, inclusi esempi, consulta the section called "Applicazioni Python".

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

Questi sono i metodi disponibili:

- get_secret_string
- get_secret_binary

get_secret_string

Recupera il valore della stringa del segreto.

Sintassi della richiesta

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

Parametri

- `secret_id` (string) -- [Obbligatorio] Il nome o l'ARN del segreto.
- `version_stage` (string) -- La versione dei segreti da recuperare. Per ulteriori informazioni, consulta Versioni del segreto. Il valore predefinito è "AWSCURRENT".

Tipo restituito

string

get_secret_binary

Recupera il valore binario del segreto.

Sintassi della richiesta

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

Parametri

- `secret_id` (string) -- [Obbligatorio] Il nome o l'ARN del segreto.
- `version_stage` (string) -- La versione dei segreti da recuperare. Per ulteriori informazioni, consulta Versioni del segreto. Il valore predefinito è "AWSCURRENT".

Tipo restituito

Stringa con codifica Base64

SecretCacheConfig

Opzioni di configurazione della cache per una the section called “SecretCache”, ad esempio dimensione massima della cache e durata (TTL) per i segreti memorizzati nella cache.

Parametri

`max_cache_size` (int)

La dimensione massima della cache. Il valore predefinito è `1024` segreti.

`exception_retry_delay_base` (int)

Il numero di secondi di attesa dopo aver riscontrato un'eccezione prima di riprovare la richiesta. Il valore predefinito è `1`.

`exception_retry_growth_factor` (int)pur

Il fattore di crescita da utilizzare per calcolare il tempo di attesa tra i tentativi di richieste non riuscite. Il valore predefinito è `2`.

`exception_retry_delay_max` (int)

Il numero massimo di secondi di attesa tra le richieste non riuscite. Il valore predefinito è `3600`.

`default_version_stage` (str)

Imposta la versione dei segreti che si desidera memorizzare nella cache. Per ulteriori informazioni, consulta Versioni del segreto. Il valore predefinito è `'AWSCURRENT'`.

`secret_refresh_interval` (int)

Il numero di secondi da attendere tra gli aggiornamenti delle informazioni del segreto memorizzate nella cache. Il valore predefinito è `3600`.

`secret_cache_hook` (SecretCacheHook)

Un'implementazione della classe astratta di `SecretCacheHook`. Il valore di default è `None`.

SecretCacheHook

Un'interfaccia per collegarsi a una the section called “SecretCache” per eseguire operazioni sui segreti memorizzati al suo interno.

Questi sono i metodi disponibili:

- put
- get

put

Prepara l'oggetto per la memorizzazione nella cache.

Sintassi della richiesta

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

```
)
```

Parametri

- `obj` (object) -- [Obbligatorio] Il segreto o l'oggetto che contiene il segreto.

Tipo restituito

oggetto

get

Deriva l'oggetto dall'oggetto memorizzato nella cache.

Sintassi della richiesta

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

Parametri

- `obj` (object) -- [Obbligatorio] Il segreto o l'oggetto che contiene il segreto.

Tipo restituito

oggetto

@InjectSecretString

Questo decoratore prevede una stringa identificativa del segreto e the section called “SecretCache” come primo e secondo argomento. Il decoratore restituisce il valore della stringa del segreto. Il segreto deve contenere una stringa.

```
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

Questo decoratore prevede una stringa identificativa del segreto e the section called “SecretCache” come primo e secondo argomento. Gli argomenti rimanenti mappano i parametri dalla funzione wrapping alle chiavi JSON nel segreto. Il segreto deve contenere una stringa nella struttura JSON.

Per un segreto che contiene questo JSON:

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

Gli esempi seguenti mostrano come estrarre i valori JSON per `username` e `password` dal segreto.

```
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')
```

Recuperare segreti AWS Secrets Manager nelle applicazioni .NET

Quando si recupera un segreto, è possibile utilizzare il componente di caching basato su .NET di Secrets Manager per memorizzarlo nella cache per un uso futuro. Il recupero di un segreto memorizzato nella cache è più veloce rispetto al recupero da Secrets Manager. Poiché è previsto un costo per chiamare le API di Secrets Manager, l'utilizzo di una cache può ridurre i costi. Per tutti i modi in cui puoi recuperare i segreti, vedi Recuperare segreti.

La policy della cache è Least Recently Used (LRU), quindi quando la cache deve eliminare un segreto, elimina il segreto usato meno di recente. Di default, la cache aggiorna i segreti ogni ora. È possibile configurare la frequenza con cui il segreto viene aggiornato nella cache ed è possibile collegarsi al recupero del segreto per aggiungere altre funzionalità.

La cache non impone la rimozione di oggetti inutili (garbage collection) una volta liberati i riferimenti alla cache. L'implementazione della cache non include l'invalidazione della cache. L'implementazione della cache è incentrata sulla cache stessa e non è rafforzata o focalizzata sulla sicurezza. Se hai bisogno di un livello di sicurezza aggiuntivo, come la crittografia degli elementi nella cache, usa le interfacce e i metodi astratti forniti.

Per usare il componente, devi disporre dei seguenti elementi:

- .NET Framework 4.6.2 o versioni successive o .NET Standard 2.0 o versioni successive. Consulta Download .NET sul sito Web di Microsoft .NET
- SDK AWS per .NET. Per informazioni, consultare the section called “AWS SDK”.

Per scaricare il codice sorgente, consulta Client di caching per .NET su GitHub.

Per utilizzare la cache, creane prima un'istanza, quindi recupera il segreto usando `GetSecretString` o `GetSecretBinary`. Nei recuperi successivi, la cache restituisce la copia del segreto memorizzata nella cache.

Recupero del pacchetto di caching

- Completa una delle seguenti operazioni:
  - Nella directory del progetto, esegui il seguente comando della CLI .NET.

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

  - Aggiungi il seguente riferimento al pacchetto al tuo file `.csproj`.

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

Autorizzazioni richieste:

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

Per ulteriori informazioni, consulta Riferimento per le autorizzazioni.

Riferimento

- SecretsManagerCache
- SecretCacheConfiguration
- ISecretCacheHook

Example Recupero di un segreto

L'esempio di codice riportato di seguito mostra un metodo che recupera un segreto denominato *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 Configurazione della durata dell'aggiornamento della cache Time To Live (TTL)

L'esempio di codice riportato di seguito mostra un metodo che recupera un segreto denominato *MySecret* e imposta la durata dell'aggiornamento della cache TTL su 24 ore.

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

namespace LambdaExample
{
```

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

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

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

SecretsManagerCache

Una cache in memoria per i segreti richiesti da Secrets Manager. Si usa the section called "GetSecretString" o the section called "GetSecretBinary" per recuperare un segreto dalla cache. È possibile configurare le impostazioni della cache specificando un oggetto the section called "SecretCacheConfiguration" nel costruttore.

Per ulteriori informazioni, inclusi esempi, consulta the section called "Applicazioni .NET".

Costruttori

`public SecretsManagerCache()`

Costruttore di default per un oggetto `SecretsManagerCache`.

`public SecretsManagerCache(IAmazonSecretsManager secretsManager)`

Costruisce una nuova cache utilizzando un client di Secrets Manager creato utilizzando l'AmazonSecretsManagerClient fornito. Utilizza questo costruttore per personalizzare il client di Secrets Manager, ad esempio per utilizzare una regione o un endpoint specifico.

Parametri

secretsManager

L'AmazonSecretsManagerClient da cui recuperare segreti.

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

Costruisce una nuova cache del segreto utilizzando il the section called "SecretCacheConfiguration" fornito. Utilizza questo costruttore per configurare la cache, ad esempio il numero di segreti da inserire nella cache e la frequenza di aggiornamento.

Parametri

config

Un the section called "SecretCacheConfiguration" che contiene informazioni di configurazione per la cache.

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

Costruisce una nuova cache utilizzando un client di Secrets Manager creato utilizzando l'AmazonSecretsManagerClient fornito e una the section called "SecretCacheConfiguration". Utilizza questo costruttore per personalizzare il client di Secrets Manager, ad esempio per utilizzare una regione o un endpoint specifici e configurare la cache, ad esempio il numero di segreti da inserire nella cache e la frequenza di aggiornamento.

Parametri

secretsManager

L'AmazonSecretsManagerClient da cui recuperare segreti.

config

Un the section called "SecretCacheConfiguration" che contiene informazioni di configurazione per la cache.

Metodi

GetSecretString

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

Recupera un segreto stringa da Secrets Manager.

Parametri

secretId

L'ARN o il nome del segreto da recuperare.

GetSecretBinary

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

Recupera un segreto binario da Secrets Manager.

Parametri

secretId

L'ARN o il nome del segreto da recuperare.

RefreshNowAsync

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

Richiede il valore del segreto da Secrets Manager e aggiorna la cache con eventuali modifiche. Se non esiste una voce di cache esistente, ne crea una nuova. Restituisce `true` se l'aggiornamento ha esito positivo.

Parametri

secretId

L'ARN o il nome del segreto da recuperare.

GetCachedSecret

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

Restituisce la voce di cache per il segreto specificato se presente nella cache. In caso contrario, recupera il segreto da Secrets Manager e crea una nuova voce di cache.

Parametri

secretId

L'ARN o il nome del segreto da recuperare.

SecretCacheConfiguration

Opzioni di configurazione della cache per un the section called “SecretsManagerCache”, come dimensione massima della cache e durata (TTL) per i segreti memorizzati nella cache.

Proprietà

CacheItemTTL

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

Il TTL di un elemento della cache in millisecondi. Il valore predefinito è `3600000` ms o 1 ora. Il massimo è `4294967295` ms, vale a dire circa 49,7 giorni.

MaxCacheSize

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

La dimensione massima della cache. Il valore di default è 1024 segreti. Il numero massimo è pari a 65.535.

VersionStage

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

Imposta la versione dei segreti che si desidera memorizzare nella cache. Per ulteriori informazioni, consulta Versioni del segreto. Il valore predefinito è `"AWSCURRENT"`.

Client

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

L'AmazonSecretsManagerClient da cui recuperare segreti. Se è `null`, la cache crea un'istanza di un nuovo client. Il valore predefinito è `null`.

CacheHook

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

Un the section called “ISecretCacheHook”.

ISecretCacheHook

Un'interfaccia per collegarsi a una the section called “SecretsManagerCache” per eseguire operazioni sui segreti memorizzati al suo interno.

Metodi

Put

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

Prepara l'oggetto per la memorizzazione nella cache.

Restituisce l'oggetto da memorizzare nella cache.

Get

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

Deriva l'oggetto dall'oggetto memorizzato nella cache.

Restituisce l'oggetto da restituire dalla cache

Recuperare segreti AWS Secrets Manager nelle applicazioni Go

Quando si recupera un segreto, è possibile utilizzare il componente di caching basato su Go di Secrets Manager per memorizzarlo nella cache per un uso futuro. Il recupero di un segreto memorizzato nella cache è più veloce rispetto al recupero da Secrets Manager. Poiché è previsto un costo per chiamare le API di Secrets Manager, l'utilizzo di una cache può ridurre i costi. Per tutti i modi in cui puoi recuperare i segreti, vedi Recuperare segreti.

La policy della cache è Least Recently Used (LRU), quindi quando la cache deve eliminare un segreto, elimina il segreto usato meno di recente. Di default, la cache aggiorna i segreti ogni ora. È possibile configurare la frequenza con cui il segreto viene aggiornato nella cache ed è possibile collegarsi al recupero del segreto per aggiungere altre funzionalità.

La cache non impone la rimozione di oggetti inutili (garbage collection) una volta liberati i riferimenti alla cache. L'implementazione della cache non include l'invalidazione della cache. L'implementazione della cache è incentrata sulla cache stessa e non è rafforzata o focalizzata sulla sicurezza. Se hai bisogno di un livello di sicurezza aggiuntivo, come la crittografia degli elementi nella cache, usa le interfacce e i metodi astratti forniti.

Per usare il componente, devi disporre dei seguenti elementi:

- AWS SDK per Go. Per informazioni, consultare the section called “AWS SDK”.

Per scaricare il codice sorgente, consulta Client di caching Go di Secrets Manager su GitHub.

Per configurare un ambiente di sviluppo Go, consulta Introduzione a Golang sul sito Web del linguaggio di programmazione Go.

Autorizzazioni richieste:

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

Per ulteriori informazioni, consulta Riferimento per le autorizzazioni.

Riferimento

- type Cache
- type CacheConfig
- type CacheHook

Example Recupero di un segreto

L'esempio di codice riportato di seguito mostra una funzione Lambda che recupera un segreto.

```
package main

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

var (
```

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

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

  // Use the secret, return success
}

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

type Cache

Una cache in memoria per i segreti richiesti da Secrets Manager. Si usa the section called "GetSecretString" o the section called "GetSecretBinary" per recuperare un segreto dalla cache.

Nell'esempio seguente viene illustrato come configurare le impostazioni della cache.

```
// 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 },
)
```

Per ulteriori informazioni, inclusi esempi, consulta the section called "Applicazioni Go".

Metodi

Novità

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

New costruisce una cache del segreto utilizzando opzioni funzionali, altrimenti usa i valori predefiniti. Inizializza un client SecretsManager da una nuova sessione. Inizializza CacheConfig ai valori predefiniti. Inizializza la cache LRU con una dimensione massima predefinita.

GetSecretString

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

GetSecretString ottiene il valore della stringa del segreto dalla cache per un determinato ID segreto. Restituisce la stringa del segreto e un errore in caso di errore dell'operazione.

GetSecretStringWithStage

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

GetSecretStringWithStage ottiene il valore della stringa del segreto dalla cache per determinati ID segreto e fase della versione. Restituisce la stringa del segreto e un errore in caso di errore dell'operazione.

GetSecretBinary

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

GetSecretBinary ottiene il valore binario della stringa del segreto dalla cache per un determinato ID segreto. Restituisce il numero binario del segreto e un errore in caso di errore dell'operazione.

GetSecretBinaryWithStage

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

GetSecretBinaryWithStage ottiene il valore binario del segreto dalla cache per determinati ID segreto e fase della versione. Restituisce il numero binario del segreto e un errore in caso di errore dell'operazione.

type CacheConfig

Opzioni di configurazione della cache per una Cache, ad esempio dimensione massima della cache, fase della versione di default e durata (TTL) per i segreti memorizzati nella 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
}
```

type CacheHook

Un'interfaccia per collegarsi a una Cache per eseguire operazioni sul segreto memorizzato al suo interno.

Metodi

Put

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

Prepara l'oggetto per la memorizzazione nella cache.

Get

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

Deriva l'oggetto dall'oggetto memorizzato nella cache.

Utilizzare i segreti di AWS Secrets Manager in AWS Batch

AWS Batch ti aiuta a eseguire carichi di lavoro di elaborazione in batch in Cloud AWS. Con AWS Batch, puoi inserire i dati sensibili nei processi archiviandoli nei segreti di AWS Secrets Manager e

quindi facendo riferimento a essi nella definizione del processo. Per ulteriori informazioni, consulta Specifica di dati sensibili con Secrets Manager.

Recuperare un segreto AWS Secrets Manager in una risorsa AWS CloudFormation

Con AWS CloudFormation, puoi recuperare un segreto da utilizzare in un altra risorsa AWS CloudFormation. Uno scenario comune consiste nel creare prima un segreto con una password generata da Secrets Manager e quindi recuperare il nome utente e la password dal segreto da utilizzare come credenziali per un nuovo database. Per ulteriori informazioni sulla creazione di un segreto con AWS CloudFormation, consulta AWS CloudFormation.

Per recuperare un segreto in un modello AWS CloudFormation, utilizza un riferimento dinamico. Quando crei lo stack, il riferimento dinamico inserisce il valore segreto nella risorsa AWS CloudFormation, quindi non devi codificare le informazioni segrete. Invece, riferisciti al segreto per nome o tramite l'ARN. È possibile utilizzare un riferimento dinamico per un segreto in qualsiasi proprietà della risorsa. Non è possibile utilizzare un riferimento dinamico per un segreto nei metadati delle risorse, ad esempio `AWS::CloudFormation::Init`, perché ciò renderebbe visibile il valore del segreto nella console.

Un riferimento dinamico per un segreto ha il seguente modello:

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

secret-id

Il nome o l'ARN del segreto. Per accedere a un segreto nell'account AWS, puoi utilizzare il nome del segreto. Per accedere a un segreto in un altro account AWS, utilizza l'ARN del segreto.

json-key (Facoltativo)

Il nome della chiave della coppia chiave-valore di cui intendi recuperare il valore. Se non si specifica `json-key`, AWS CloudFormation recupera tutto il testo del segreto. Questo segmento può non includere il carattere di due punti (`:`).

version-stage (Facoltativo)

La versione del segreto da utilizzare. Secrets Manager utilizza le etichette temporanee per tenere traccia delle differenti versioni durante il processo di rotazione. Se utilizzi `version-stage`, non

specificare `version-id`. Se non specifichi né `version-stage` né `version-id`, la versione di default sarà la `AWSCURRENT`. Questo segmento può non includere il carattere di due punti (`:`).

version-id (Facoltativo)

L'identificatore univoco della versione del segreto da utilizzare. Se specifichi `version-id`, non è necessario specificare anche `version-stage`. Se non specifichi né `version-stage` né `version-id`, la versione di default sarà la `AWSCURRENT`. Questo segmento può non includere il carattere di due punti (`:`).

Per ulteriori informazioni, consulta Utilizzo di riferimenti dinamici per specificare i segreti di Secrets Manager.

> **Note**
>
> Non creare un riferimento dinamico che abbia una barra rovesciata (`\`) come valore finale. AWS CloudFormation non è in grado di risolvere tali riferimenti, il che comporta l'errore della risorsa.

Utilizzare i segreti di AWS Secrets Manager in Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) è un servizio di orchestrazione di container completamente gestito che facilita l'implementazione, la gestione e il dimensionamento delle applicazioni containerizzate. Puoi inserire dati sensibili nei tuoi container facendo riferimento ai segreti di Secrets Manager. Per ulteriori informazioni, consulta le seguenti pagine nella Guida per gli sviluppatori di Amazon Elastic Container Service:

- Tutorial: specifica di dati sensibili utilizzando segreti Secrets Manager
- Recupero programmatico dei segreti attraverso l'applicazione
- Recupero di segreti attraverso variabili di ambiente
- Recupero di segreti per la configurazione di registrazione

Usa AWS Secrets Manager i segreti in Amazon Elastic Kubernetes Service

Per mostrare i segreti di Secrets Manager come file montati nei pod Amazon EKS, puoi utilizzare AWS Secrets and Configuration Provider (ASCP) per il driver CSI Kubernetes Secrets Store. L'ASCP funziona con Amazon Elastic Kubernetes Service (Amazon EKS) 1.17+ che esegue un gruppo di nodi Amazon EC2. AWS Fargate i gruppi di nodi non sono supportati. Con ASCP, puoi archiviare e gestire i segreti in Secrets Manager e recuperarli tramite i carichi di lavoro in esecuzione su Amazon EKS. Se il tuo segreto contiene più coppie chiave/valore in formato JSON, puoi scegliere quali montare in Amazon EKS. L'ASCP utilizza Sintassi JMESPath per interrogare le coppie chiave/valore nel tuo segreto. L'ASCP funziona anche con i parametri dell'archivio parametri.

Puoi utilizzare i ruoli e le policy IAM per concedere l'accesso ai tuoi segreti a specifici pod Amazon EKS in un cluster.

Per descrivere quali file creare nel pod Amazon EKS e quali segreti inserire in essi, crea un file the section called “`SecretProviderClass`” YAML. La `SecretProviderClass` deve trovarsi nello stesso spazio dei nomi del pod Amazon EKS a cui fa riferimento.

Se utilizzi un cluster Amazon EKS privato, assicurati che il VPC in cui si trova il cluster disponga di un endpoint Secrets Manager. Il driver CSI di Secrets Store utilizza l'endpoint per effettuare chiamate a Secrets Manager. Per informazioni su come creare un endpoint in un VPC, consulta la sezione Endpoint VPC.

Se si utilizza la rotazione automatica di Secrets Manager per i propri segreti, è anche possibile utilizzare la funzione di riconciliazione rotazione Secrets Store CSI Driver per assicurarsi di recuperare il segreto più recente da Secrets Manager. Per ulteriori informazioni, consulta Rotazione automatica dei contenuti montati e sincronizzati Kubernetes Secrets.

Per un tutorial su come utilizzare ASCP, consulta the section called “Tutorial”.

Installazione di ASCP

L'ASCP è disponibile GitHub nel repository secrets-store-csi-provider-aws. Il repository contiene anche file YAML di esempio per la creazione e il montaggio di un segreto.

Per installare ASCP

- Per installare il driver CSI di Secrets Store e ASCP tramite Helm, usa i comandi riportati di seguito. Per assicurarti che il repository punti al grafico più recente, utilizza `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
```

  In alternativa, per eseguire l'installazione tramite il file YAML nella directory di implementazione, utilizza i comandi riportati di seguito.

```
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
```

Fase 1: Configurazione del controllo degli accessi

Per concedere al tuo pod Amazon EKS l'accesso ai segreti in Secrets Manager, è necessario innanzitutto creare una policy che concede a `secretsmanager:GetSecretValue` e `secretsmanager:DescribeSecret` l'accesso ai segreti a cui il pod deve accedere. Per esempi di policy, consulta Esempi di policy di autorizzazione.

Quindi si crea un Ruolo IAM per l'account del servizio e collegare la policy ad esso. Per ulteriori informazioni, consulta la sezione Ruolo IAM per l'account del servizio.

L'ASCP recupera l'identità del pod e la scambia con il ruolo IAM. ASCP assume il ruolo IAM del pod, che gli dà accesso ai segreti autorizzati. Altri container non possono accedere ai segreti a meno che non vengano associati anche al ruolo IAM.

Se utilizzi un cluster Amazon EKS privato, assicurati che il VPC in cui si trova il cluster abbia un AWS STS endpoint. Per informazioni sulla creazione di un endpoint, consulta Interface VPC endpoints nellaAWS Identity and Access Management Guida per l'utente.

Fase 2: Identificazione dei segreti da montare

Per determinare quali segreti l'ASCP monta in Amazon EKS come file sul file system, è necessario creare un file `SecretProviderClass` YAML. Il file `SecretProviderClass` YAML elenca i segreti da montare e il nome del file con cui montarli. La `SecretProviderClass` deve trovarsi nello stesso spazio dei nomi del pod Amazon EKS a cui fa riferimento.

Gli esempi seguenti mostrano come usare `SecretProviderClass` per descrivere i segreti che desideri montare e come denominare i file montati nel pod Amazon EKS. Per ulteriori informazioni, consulta the section called “`SecretProviderClass`”.

Esempi:

- Esempio: montaggio di segreti per nome o ARN
- Esempio: montaggio di coppie chiave/valore da un segreto
- Esempio: definizione di una Regione di failover per un segreto multiregione
- Esempio: scelta di un segreto di failover da montare

Esempio: montaggio di segreti per nome o ARN

Il seguente esempio mostra un `SecretProviderClass` che monta tre file in Amazon EKS:

1. Un segreto specificato dall'ARN completo.
2. Un segreto specificato dal nome.
3. Una versione specifica di un segreto.

```
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"
```

Esempio: montaggio di coppie chiave/valore da un segreto

Il seguente esempio mostra un `SecretProviderClass` che monta tre file in Amazon EKS:

1. Un segreto specificato dall'ARN completo.
2. La `username` coppia chiave/valore dallo stesso segreto.
3. La `password` coppia chiave/valore dallo stesso segreto.

```
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
```

Esempio: definizione di una Regione di failover per un segreto multiregione

Per garantire la disponibilità durante le interruzioni della connettività o per le configurazioni del ripristino di emergenza, ASCP supporta una funzionalità di failover automatizzato per recuperare i segreti da una Regione secondaria.

L'esempio seguente mostra una `SecretProviderClass` che recupera un segreto replicato in più Regioni. In questo esempio, l'ASCP tenta di recuperare il segreto sia da `us-east-1` che

da `us-east-2`. Se una delle Regioni restituisce un errore 4xx, ad esempio per un problema di autenticazione, l'ASCP non installa nessuno dei due segreti. Se il segreto viene recuperato correttamente da `us-east-1`, l'ASCP monta tale valore del segreto. Se il segreto non viene recuperato correttamente da `us-east-1`, ma viene correttamente recuperato da `us-east-2`, l'ASCP monta tale valore del segreto.

```
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"
```

Esempio: scelta di un segreto di failover da montare

L'esempio seguente mostra una `SecretProviderClass` che specifica quale segreto montare in caso di failover. Il segreto di failover non è una replica. In questo esempio, l'ASCP tenta di recuperare i due segreti specificati da `objectName`. Se uno dei due restituisce un errore 4xx, ad esempio per un problema di autenticazione, l'ASCP non installa nessuno dei due segreti. Se il segreto viene recuperato correttamente da `us-east-1`, l'ASCP monta tale valore del segreto. Se il segreto non viene recuperato correttamente da `us-east-1`, ma viene correttamente recuperato da `us-east-2`, l'ASCP monta tale valore del segreto. Il file montato in Amazon EKS è denominato `MyMountedSecret`.

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

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

Risoluzione dei problemi

È possibile visualizzare la maggior parte degli errori descrivendo l'implementazione del pod.

Per visualizzare i messaggi di errore per il container

1. È possibile ottenere un elenco di nomi di pod con il comando seguente. Se non si sta utilizzando lo spazio dei nomi predefinito, utilizzare `-n <NAMESPACE>`.

   ```
   kubectl get pods
   ```

2. Per descrivere il pod, nel comando seguente, per *<PODID>* utilizzare l'ID pod dai pod trovati nel passaggio precedente. Se non si sta utilizzando lo spazio dei nomi predefinito, utilizzare `-n <NAMESPACE>`.

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

Come visualizzare gli errori per l'ASCP

- Per trovare maggiori informazioni nei log del provider, nel comando seguente, per utilizzare l'ID del pod -aws. csi-secrets-store-provider <PODID>

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

Tutorial: crea e AWS Secrets Manager monta un segreto in un pod Amazon EKS

In questo tutorial si crea un segreto di esempio in Secrets Manager, quindi si monta il segreto in un pod Amazon EKS e lo si distribuisce.

Prima di iniziare, installa ASCP: the section called “Installazione di ASCP”.

Come creare e montare un segreto

1. Imposta il nome Regione AWS e il nome del cluster come variabili di shell in modo da poterle utilizzare nei comandi bash. Per<REGION>, inserisci il Regione AWS luogo in cui viene eseguito il tuo cluster Amazon EKS. Per *<CLUSTERNAME>* inserisci un nome per il tuo cluster.

```
REGION=<REGION>
CLUSTERNAME=<CLUSTERNAME>
```

2. Crea un segreto di prova. Per ulteriori informazioni, consulta Creazione e gestione di segreti.

```
aws --region "$REGION" secretsmanager  create-secret --name MySecret --secret-
string '{"username":"lijuan", "password":"hunter2"}'
```

3. Crea una policy di risorse per il pod che ne limita l'accesso al segreto creato nella fase precedente. Per *<SECRETARN>*, usa l'ARN del segreto. Salva l'ARN della policy in una variabile di 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. Crea un provider OIDC IAM per il cluster se non ne è già presente uno. Per ulteriori informazioni, consulta Creare un provider OIDC IAM per il cluster.

```
eksctl utils associate-iam-oidc-provider --region="$REGION" --
cluster="$CLUSTERNAME" --approve # Only run this once
```

5. Creare l'account di servizio utilizzato dal pod e associare la policy delle risorse creata nel passaggio 3 a tale account di servizio. Per questo tutorial, per il nome dell'account di servizio che utilizzi nginx-deployment-sa. Per ulteriori informazioni, consulta Creare di un ruolo IAM per l'account di servizio.

```
eksctl create iamserviceaccount --name nginx-deployment-sa --region="$REGION" --
cluster "$CLUSTERNAME" --attach-policy-arn "$POLICY_ARN" --approve --override-
existing-serviceaccounts
```

6. Creazione dell' `SecretProviderClass` per specificare quale segreto montare nel pod. Il comando seguente utilizza `ExampleSecretProviderClass.yaml` nella directory ASCP GitHub Repo examples il mount del segreto creato nel passaggio 2. Per informazioni su come creare il tuo `SecretProviderClass`, consulta the section called "`SecretProviderClass`".

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

7. Distribuzione del pod Il comando seguente utilizza `ExampleDeployment.yaml` la directory ASCP degli esempi di GitHub repo per montare il segreto nel `/mnt/secrets-store` pod.

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

8. Per assicurarsi che il segreto sia stato montato correttamente, utilizza il seguente comando e verifica che venga visualizzato il valore del segreto.

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

   Viene visualizzato il valore del segreto.

```
{"username":"lijuan", "password":"hunter2"}
```

`SecretProviderClass`

Puoi utilizzare YAML per descrivere quali segreti montare in Amazon EKS utilizzando l'ASCP. Per alcuni esempi, consulta Identificazione dei segreti montare.

```
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
   name: <NAME>
spec:
```

```
  provider: aws
  parameters:
    region:
    failoverRegion:
    pathTranslation:
    objects:
```

Il campo `parameters` contiene i dettagli della richiesta di montaggio:

Regione

(Facoltativo) Il Regione AWS segreto. Se non utilizzi questo campo, l'ASCP cerca la regione dall'annotazione sul nodo. Questa ricerca aggiunge un sovraccarico alle richieste di montaggio, quindi consigliamo di fornire la regione per i cluster che utilizzano un numero elevato di pod.

Se specifichi anche la `failoverRegion`, l'ASCP tenta di recuperare il segreto da entrambe le Regioni. Se una delle Regioni restituisce un errore 4xx, ad esempio per un problema di autenticazione, l'ASCP non installa nessuno dei due segreti. Se il segreto viene recuperato correttamente da `region`, l'ASCP monta tale valore del segreto. Se il segreto non viene recuperato correttamente da `region`, ma viene correttamente recuperato da `failoverRegion`, l'ASCP monta tale valore del segreto.

failoverRegion

(Facoltativo) Se si include questo campo, l'ASCP tenta di recuperare il segreto dalle Regioni definite in`region` e in questo campo. Se una delle Regioni restituisce un errore 4xx, ad esempio per un problema di autenticazione, l'ASCP non installa nessuno dei due segreti. Se il segreto viene recuperato correttamente da `region`, l'ASCP monta tale valore del segreto. Se il segreto non viene recuperato correttamente da `region`, ma viene correttamente recuperato da `failoverRegion`, l'ASCP monta tale valore del segreto. Per un esempio su come utilizzare questo campo, consulta la sezione Definizione di una Regione di failover per un segreto multiregione.

pathTranslation

(Facoltativo) Un singolo carattere di sostituzione da utilizzare se il nome del file in Amazon EKS conterrà il carattere separatore di percorso, come barra (/) su Linux. L'ASCP non è in grado di creare un file montato che contiene un carattere separatore di percorso. Invece, l'ASCP sostituisce il carattere separatore di percorso con un carattere diverso. Se non utilizzi questo campo, il carattere sostitutivo è il carattere di sottolineatura (_), quindi ad esempio, `My/Path/Secret` monta come `My_Path_Secret`.

Per impedire la sostituzione dei caratteri, immettere la stringa `False`.

objects

Una stringa contenente una dichiarazione YAML dei segreti da montare. Si consiglia di utilizzare una stringa multiriga YAML o un carattere pipe (|).

objectName

Il nome o l'ARN completo del segreto. Se usi l'ARN, puoi omettere `objectType`. Questo campo diventa il nome file del segreto nel pod Amazon EKS, a meno che tu non specifichi `objectAlias`. Se utilizzi un ARN, la Regione nell'ARN deve corrispondere al campo `region`. Se includi un `failoverRegion`, questo campo rappresenta l'`objectName` primario.

objectType

Obbligatorio se non si utilizza un ARN di Secrets Manager per `objectName`. Può essere `secretsmanager` o `ssmparameter`.

objectAlias

(Facoltativo) Il nome file del segreto nel pod Amazon EKS. Se non indichi questo campo, `objectName` viene visualizzato come nome del file.

objectVersion

(Facoltativo) L'ID di versione del segreto. Non è consigliato perché è necessario aggiornare l'ID di versione ogni volta che si aggiorna il segreto. Per impostazione predefinita viene utilizzata la versione più recente. Se includi un `failoverRegion`, questo campo rappresenta l'`objectVersion` primario.

objectVersionLabel

(Facoltativo) L'alias per la versione. L'impostazione predefinita è la versione più recente AWSCURRENT. Per ulteriori informazioni, consulta the section called “Versione”. Se includi un `failoverRegion`, questo campo rappresenta l'`objectVersionLabel` primario.

jmesPath

(Facoltativo) Una mappa delle chiavi nel segreto per i file da montare in Amazon EKS. Per utilizzare questo campo, il valore del segreto deve essere in formato JSON. Se si utilizza questo campo, è necessario includere i sottocampi `path` e `objectAlias`.

path

Una chiave di una coppia chiave/valore nel JSON del valore del segreto. Se il campo contiene un trattino, usa le virgolette singole per evitarlo, ad esempio: `path: '"hyphenated-path"'`

objectAlias

Il nome del file da montare nel pod Amazon EKS. Se il campo contiene un trattino, usa le virgolette singole per evitarlo, ad esempio: `objectAlias: '"hyphenated-alias"'`

failoverObject

(Facoltativo) Se specifichi questo campo, l'ASCP tenta di recuperare sia il segreto specificato nel campo primario `objectName` che il segreto specificato nel sottocampo `failoverObject` `objectName`. Se uno dei due restituisce un errore 4xx, ad esempio per un problema di autenticazione, l'ASCP non installa nessuno dei due segreti. Se il segreto viene recuperato con successo dall'`objectName` primario, l'ASCP monta tale valore del segreto. Se il segreto non viene recuperato correttamente dall'`objectName` primario, ma viene recuperato con successo dall'`objectName` di failover, l'ASCP installa tale valore del segreto. Se si include questo campo, è necessario includere il campo `objectAlias`. Per un esempio su come utilizzare questo campo, consulta la sezione Scelta di un segreto di failover da montare.

In genere, si utilizza questo campo quando il segreto di failover non è una replica. Per un esempio su come specificare una replica, consulta la sezione Definizione di una Regione di failover per un segreto multiregione.

objectName

Il nome o l'ARN completo del segreto di failover. Se utilizzi un ARN, la Regione nell'ARN deve corrispondere al campo `failoverRegion`.

objectVersion

(Facoltativo) L'ID di versione del segreto. Deve corrispondere all'`objectVersion` primaria. Non è consigliato perché è necessario aggiornare l'ID di versione ogni volta che si aggiorna il segreto. Per impostazione predefinita viene utilizzata la versione più recente.

objectVersionLabel

(Facoltativo) L'alias per la versione. L'impostazione predefinita è la versione più recente AWSCURRENT. Per ulteriori informazioni, consulta the section called “Versione”.

Usa AWS Secrets Manager i segreti nei GitHub lavori

Per utilizzare un segreto in un GitHub lavoro, puoi utilizzare un' GitHub azione per recuperare i segreti AWS Secrets Manager e aggiungerli come variabili di ambiente mascherate nel tuo GitHub flusso di lavoro. Per ulteriori informazioni sulle GitHub azioni, consulta Understanding GitHub Actions in the GitHub Docs.

Quando aggiungi un segreto al tuo GitHub ambiente, questo è disponibile per tutte le altre fasi del tuo GitHub lavoro. Segui le indicazioni contenute in Security Hardening for GitHub Actions per evitare che i segreti presenti nel tuo ambiente vengano utilizzati in modo improprio.

È possibile impostare l'intera stringa nel valore segreto come valore della variabile di ambiente oppure, se la stringa è JSON, è possibile analizzare il JSON per impostare variabili di ambiente individuali per ogni coppia chiave-valore JSON. Se il valore segreto è binario, l'operazione lo converte in una stringa.

Per visualizzare le variabili di ambiente create dai tuoi segreti, attiva la registrazione del debug. Per ulteriori informazioni, consulta Abilitazione della registrazione di debug nei Documenti. GitHub

Per utilizzare le variabili di ambiente create dai tuoi segreti, consulta Variabili di ambiente nei Documenti. GitHub

Prerequisiti

Per utilizzare questa azione, devi prima configurare AWS le credenziali e impostarle Regione AWS nel tuo GitHub ambiente utilizzando la `configure-aws-credentials` procedura. Segui le istruzioni riportate nella sezione Configurazione AWS delle credenziali Action for GitHub Actions to Assume il ruolo direttamente utilizzando il provider GitHub OIDC. Ciò consente di utilizzare credenziali di breve durata ed evitare di memorizzare chiavi di accesso aggiuntive al di fuori di Gestione dei segreti.

Il ruolo IAM assunto dall'operazione deve disporre delle seguenti autorizzazioni:

- `GetSecretValue` sui segreti che desideri recuperare.
- `ListSecrets` su tutti i segreti.
- (Facoltativo) `Decrypt` KMS key se i segreti sono crittografati con un. chiave gestita dal cliente

Per ulteriori informazioni, consulta Autenticazione e controllo degli accessi.

Utilizzo

Per utilizzare l'operazione, aggiungi un passaggio al flusso di lavoro che utilizza la seguente sintassi.

```
- 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
```

Parametri

`secret-ids`

ARN segreti, nomi e prefissi dei nomi.

Per impostazione predefinita, la fase crea ogni nome di variabile di ambiente dal nome del segreto, trasformato in modo da includere solo lettere maiuscole, numeri e caratteri di sottolineatura e in modo che non inizi con un numero.

Per impostare il nome della variabile di ambiente, inseriscilo prima dell'ID del segreto, seguito da una virgola. Ad esempio, `ENV_VAR_1, secretId` crea una variabile di ambiente denominata ENV_VAR_1 dal `secretId` del segreto. Il nome della variabile di ambiente può contenere lettere maiuscole, numeri e il carattere di sottolineatura.

Per utilizzare un prefisso, inserisci almeno tre caratteri seguiti da un asterisco. Ad esempio, `dev*` abbina tutti i segreti con un nome che inizia in dev. Possono essere recuperati fino a 100 segreti corrispondenti. Se imposti il nome della variabile e il prefisso trova corrispondenze con più segreti, l'operazione ha esito negativo.

`parse-json-secrets`

(Facoltativo) Per impostazione predefinita, l'operazione imposta il valore della variabile di ambiente sull'intera stringa JSON nel valore del segreto. Imposta `parse-json-secrets` su `true` per creare variabili di ambiente per ogni coppia chiave-valore nel JSON.

Ricorda che se il JSON utilizza chiavi con distinzione tra maiuscole e minuscole (come "nome" e "Nome"), l'operazione risconterà conflitti di nomi duplicati. In questo caso, imposta `parse-json-secrets` su `false` e analizza il valore segreto JSON separatamente.

Denominazione delle variabili di ambiente

Le variabili di ambiente create dall'azione hanno lo stesso nome dei segreti da cui provengono. Le variabili di ambiente hanno requisiti di denominazione più rigorosi rispetto ai segreti, quindi l'azione trasforma i nomi segreti per soddisfare tali requisiti. Ad esempio, l'operazione trasforma le lettere minuscole in lettere maiuscole. Se si analizza il codice JSON del segreto, il nome della variabile di ambiente include sia il nome segreto che il nome della chiave JSON, ad esempio. `MYSECRET_KEYNAME`

Se due variabili di ambiente finiscono con lo stesso nome, l'azione ha esito negativo. In questo caso, è necessario specificare i nomi che si desidera utilizzare per le variabili di ambiente come alias.

Esempi di casi in cui i nomi potrebbero essere in conflitto:

- Un segreto chiamato "MySecret" e un segreto chiamato «mysecret» diventerebbero entrambi variabili di ambiente denominate «MYSECRET».
- Un segreto denominato «SECRET_keyname» e un segreto analizzato in JSON denominato «Secret» con una chiave denominata «keyname» diventerebbero entrambi variabili di ambiente denominate «SECRET_KEYNAME».

È possibile impostare il nome della variabile di ambiente specificando un alias, come mostrato nell'esempio seguente, che crea una variabile denominata. `ENV_VAR_NAME`

```
secret-ids: |
  ENV_VAR_NAME, secretId2
```

Alias vuoti

- Se imposti `parse-json-secrets: true` e inserisci un alias vuoto, seguito da una virgola e quindi dall'ID segreto, l'azione assegna alla variabile di ambiente lo stesso nome delle chiavi JSON analizzate. I nomi delle variabili non includono il nome segreto.

  Se il segreto non contiene un codice JSON valido, l'azione crea una variabile di ambiente e la nomina con lo stesso nome del segreto.
- Se imposti `parse-json-secrets: false` e inserisci un alias vuoto, seguito da una virgola e dall'ID segreto, l'azione nomina le variabili di ambiente come se non avessi specificato un alias.

L'esempio seguente mostra un alias vuoto.

```
,secret2
```

Esempi

Example 1 Recupera segreti per nome e per ARN

L'esempio seguente crea variabili di ambiente per i segreti identificati dal nome e dall'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
```

Variabili di ambiente create:

```
EXAMPLESECRETNAME: secretValue1
TEST1: secretValue2
_0_TEST_SECRET: secretValue3
_PROD_EXAMPLE_SECRET: secretValue4
SECRET_ALIAS_1: secretValue5
SECRET_ALIAS_2: secretValue6
SECRET2: secretValue7
```

Example 2 Recupera tutti i segreti che iniziano con un prefisso

L'esempio di seguito crea variabili di ambiente per tutti i segreti con nomi che iniziano con *beta*.

```
- name: Get Secret Names by Prefix
  uses: 2
  with:
    secret-ids: |
      beta*    # Retrieves all secrets that start with 'beta'
```

Variabili di ambiente create:

```
BETASECRETNAME: secretValue1
BETATEST: secretValue2
BETA_NEWSECRET: secretValue3
```

Example 3 Analizza il JSON nel segreto

L'esempio di seguito crea variabili di ambiente analizzando il JSON nel segreto.

```
- 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
```

Il segreto `test/secret` ha il seguente valore segreto.

```
{
  "api_user": "user",
  "api_key": "key",
  "config": {
    "active": "true"
  }
}
```

Il segreto `secret2` ha il seguente valore segreto.

```
{
  "myusername": "alejandro_rosalez",
  "mypassword": "EXAMPLE_PASSWORD"
}
```

Variabili di ambiente create:

```
TEST_SECRET_API_USER: "user"
TEST_SECRET_API_KEY: "key"
TEST_SECRET_CONFIG_ACTIVE: "true"
MYUSERNAME: "alejandro_rosalez"
MYPASSWORD: "EXAMPLE_PASSWORD"
```

Utilizzare i segreti di AWS Secrets Manager in AWS IoT Greengrass

AWS IoT Greengrass è un software che estende le funzionalità del cloud ai dispositivi locali. Consente ai dispositivi di raccogliere e analizzare i dati più vicini all'origine delle informazioni, reagire autonomamente a eventi locali e comunicare in modo sicuro tra di loro sulle reti locali.

AWS IoT Greengrass ti consente di eseguire l'autenticazione con i servizi e le applicazioni dai dispositivi Greengrass senza impostare come hard-coded le password, i token o altri segreti. Si può usare AWS Secrets Manager per archiviare e gestire in modo sicuro i segreti nel cloud. AWS IoT Greengrass estende Secrets Manager ai dispositivi core Greengrass in modo che i connettori e le funzioni Lambda possano utilizzare i segreti locali per interagire con i servizi e le applicazioni.

Per integrare un segreto in un gruppo Greengrass, è necessario creare una risorsa di gruppo che faccia riferimento al segreto Secrets Manager. Questa risorsa segreta fa riferimento al segreto cloud utilizzando l'ARN associato. Per informazioni su come creare, gestire e utilizzare risorse segrete, vedere Utilizzo delle risorse segrete nella AWS IoTGuida per lo sviluppatore.

Per distribuire i segreti nel AWS IoT Greengrass Core, consulta Distribuzione dei segreti nel AWS IoT Greengrass Core.

Usa AWS Secrets Manager i segreti nelle AWS Lambda funzioni

Puoi utilizzare l'estensione Lambda AWS Parameters and Secrets per recuperare e memorizzare AWS Secrets Manager nella cache i segreti nelle funzioni Lambda senza utilizzare un SDK. Il recupero di un segreto memorizzato nella cache è più veloce rispetto al recupero da Secrets Manager. Poiché è previsto un costo per chiamare le API di Secrets Manager, l'utilizzo di una cache può ridurre i costi. L'estensione può recuperare sia i segreti di Gestione dei segreti che i parametri di Archivio dei parametri Per informazioni su Archivio parametri, consulta Integrazione di Parameter Store con le estensioni Lambda nella Guida per l'utente AWS Systems Manager .

Un'estensione Lambda è un processo complementare che si aggiunge alle funzionalità di una funzione Lambda. Per ulteriori informazioni, consulta Estensioni di Lambda nella Guida per gli sviluppatori di Lambda. Per informazioni sull'utilizzo dell'estensione in un'immagine di container, consulta Operazioni con i livelli e le estensioni Lambda nelle immagini di container . Lambda registra le informazioni di esecuzione sull'estensione insieme alla funzione utilizzando Amazon Logs. CloudWatch Per impostazione predefinita, l'estensione registra una quantità minima di

informazioni su. CloudWatch Per registrare ulteriori dettagli, imposta la variabile di ambiente `PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL` su `debug`.

Per fornire la cache in memoria per parametri e segreti, l'estensione espone un endpoint HTTP locale, la porta localhost 2773, all'ambiente Lambda. È possibile configurare la porta impostando la variabile di ambiente `PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`.

Lambda crea istanze separate corrispondenti al livello di simultaneità richiesto dalla funzione. Ogni istanza è isolata e mantiene la propria cache locale dei dati di configurazione. Per ulteriori informazioni sulle istanze Lambda e sulla concorrenza, consulta Gestione della simultaneità per una funzione Lambda nella Guida per gli sviluppatori di Lambda.

Per aggiungere l'estensione per ARM, devi usare l'architettura `arm64` per la funzione Lambda. er ulteriori informazioni, consulta Architetture del set di istruzioni Lambda nella Guida per gli sviluppatori di Lambda. L'estensione supporta ARM nelle seguenti regioni: Asia Pacifico (Mumbai), Stati Uniti orientali (Ohio), Europa (Irlanda), Europa (Francoforte), Europa (Zurigo), Stati Uniti orientali (Virginia settentrionale), Europa (Londra), Europa (Spagna), Asia Pacifico (Tokyo), Stati Uniti occidentali (Oregon), Asia Pacifico (Singapore) e Asia Pacifico (Sydney).

L'estensione utilizza un AWS client. Per informazioni sulla configurazione del AWS client, consulta il riferimento alle impostazioni nella Guida di riferimento dell'AWS SDK e degli strumenti. Se la tua funzione Lambda viene eseguita in un VPC, devi creare un endpoint VPC in modo che l'estensione possa effettuare chiamate a Secrets Manager. Per ulteriori informazioni, consulta Endpoint VPC.

Autorizzazioni richieste:

- Il ruolo di esecuzione Lambda deve disporre `secretsmanager:GetSecretValue` dell'autorizzazione al segreto.
- Se il segreto è crittografato con una chiave gestita dal cliente anziché con Chiave gestita da AWS `aws/secretsmanager`, anche il ruolo di esecuzione necessita dell'`kms:Decrypt`autorizzazione per la chiave KMS.

Per utilizzare l'estensione Lambda AWS Parameters and Secrets

1. Aggiungi il livello alla tua funzione in uno dei seguenti modi:
   - Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
     a. Scegli la tua funzione, seleziona Layers (Livelli) e poi Add a layer (Aggiungi un livello).

b. Nella pagina Add layer (Aggiungi un livello), per AWS layers (Livelli), scegli AWS Parameters and Secrets Lambda Extension (Estensione Parameters and Secrets di Lambda) e poi Add (Aggiungi).

- Usa il AWS CLI comando seguente con l'ARN appropriato per la tua regione. Per un elenco di tutti gli ARN, consulta AWS Parameters and Secrets Lambda Extension ARNs (ARN estensione Parameters and Secrets di Lambda) nella Guida per l'utente di AWS Systems Manager .

```
aws lambda update-function-configuration \
    --function-name my-function \
    --layers LayerARN
```

2. Concedi le autorizzazioni al ruolo di esecuzione di Lambda per poter accedere ai segreti:
   - Autorizzazione `secretsmanager:GetSecretValue` per il segreto. Per informazioni, consulta the section called “Esempio: Autorizzazione per recuperare valori segreti individuali”.
   - (Facoltativo) Se il segreto è crittografato con una chiave gestita dal cliente anziché con Chiave gestita da AWS `aws/secretsmanager`, il ruolo di esecuzione richiede anche l'`kms:Decrypt`autorizzazione per la chiave KMS.
   - Puoi utilizzare il controllo degli accessi basato su attributi (ABAC) con il ruolo Lambda per consentire un accesso più granulare ai segreti nell'account. Per ulteriori informazioni, consultare the section called “Esempio: Controllare l'accesso ai segreti utilizzando i tag” e the section called “Esempio: Limitare l'accesso alle identità con tag che corrispondono ai tag dei segreti”.
3. Configura la cache con le variabili di ambiente di Lambda.
4. Per recuperare i segreti dalla cache delle estensioni, innanzitutto è necessario aggiungere il parametro `X-AWS-Parameters-Secrets-Token` all'intestazione della richiesta. Imposta il token su `AWS_SESSION_TOKEN`, fornito da Lambda per tutte le funzioni in esecuzione. L'utilizzo di questa intestazione indica che chi effettua la chiamata si trova all'interno dell'ambiente Lambda.

   Il seguente esempio di Python mostra come aggiungere l'intestazione.

```
import os
headers = {"X-Aws-Parameters-Secrets-Token": os.environ.get('AWS_SESSION_TOKEN')}
```

5. Per recuperare un segreto all'interno della funzione Lambda, utilizza una delle seguenti richieste HTTP GET:

   - Per recuperare un segreto, per `secretId` utilizza l'ARN o il nome del segreto.

     ```
     GET: /secretsmanager/get?secretId=secretId
     ```

   - Per recuperare il valore del segreto precedente o una versione specifica tramite etichetta temporanea, per `secretId` usa l'ARN o il nome del segreto, mentre per `versionStage` usa l'etichetta temporanea.

     ```
     GET: /secretsmanager/get?secretId=secretId&versionStage=AWSPREVIOUS
     ```

   - Per recuperare una versione segreta specifica in base all'ID, per `secretId` usa l'ARN o il nome del segreto, mentre per `versionId` usa l'ID della versione.

     ```
     GET: /secretsmanager/get?secretId=secretId&versionId=versionId
     ```

   Example Recupero di un segreto (Python)

   Il seguente esempio di Python mostra come recuperare un segreto e analizzare il risultato usando `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 Parametri e segreti: variabili di ambiente Lambda Extension

Puoi configurare l'estensione con le seguenti variabili di ambiente.

Per ulteriori informazioni sull'utilizzo delle variabili di ambiente, consulta Utilizzo delle variabili di ambiente di Lambda nella Guida per gli sviluppatori di Lambda.

`PARAMETERS_SECRETS_EXTENSION_CACHE_ENABLED`

Imposta su "true" per memorizzare nella cache parametri e segreti. Imposta su "false" per non memorizzare nella cache questi elementi. Il valore predefinito è true.

`PARAMETERS_SECRETS_EXTENSION_CACHE_SIZE`

Il numero massimo di segreti e parametri da memorizzare nella cache. Il valore deve essere compreso tra 0 e 1000. Il valore 0 indica l'assenza di caching. Questa variabile viene ignorata se sia `SSM_PARAMETER_STORE _TTL` sia `SECRETS_MANAGER_TTL` sono impostati su 0. Il valore predefinito è 1000.

`PARAMETERS_SECRETS_EXTENSION_HTTP_PORT`

La porta per il server HTTP locale. Il valore predefinito è 2773.

`PARAMETERS_SECRETS_EXTENSION_LOG_LEVEL`

Il livello di registrazione fornito dall'estensione: `debug`, `info`, `warn`, `error` o `none`. Imposta su `debug` per visualizzare la configurazione della cache. Il valore predefinito è `info`.

`PARAMETERS_SECRETS_EXTENSION_MAX_CONNECTIONS`

Numero massimo di connessioni per i client HTTP utilizzati dall'estensione per effettuare richieste ad Archivio dei parametri o Gestione dei segreti. Questa è una configurazione specifica per client. Il valore predefinito è 3.

`SECRETS_MANAGER_TIMEOUT_MILLIS`

Timeout per le richieste a Gestione dei segreti in millisecondi. Il valore 0 indica l'assenza di timeout. Il valore predefinito è 0.

`SECRETS_MANAGER_TTL`

TTL di un segreto nella cache in secondi. Il valore 0 indica l'assenza di caching. Il massimo è 300 secondi. Questa variabile viene ignorata se `PARAMETERS_SECRETS_CACHE_SIZE` è 0. Il valore predefinito è 300 secondi.

`SSM_PARAMETER_STORE_TIMEOUT_MILLIS`

Timeout per le richieste ad Archivio dei parametri in millisecondi. Il valore 0 indica l'assenza di timeout. Il valore predefinito è 0.

`SSM_PARAMETER_STORE_TTL`

TTL di un parametro nella cache in secondi. Il valore 0 indica l'assenza di caching. Il massimo è 300 secondi. Questa variabile viene ignorata se `PARAMETERS_SECRETS_CACHE_SIZE` è 0. Il valore predefinito è 300 secondi.

Utilizzare i segreti di AWS Secrets Manager in Parameter Store

AWS Systems Manager Parameter Store fornisce uno storage sicuro e gerarchico per la gestione dei dati di configurazione e dei segreti. È possibile archiviare dati, ad esempio le password, le stringhe di database e i codici di licenza, come valori dei parametri. Tuttavia, Archivio parametri non fornisce servizi di rotazione automatica per i segreti archiviati. Al contrario, Archivio parametri consente di archiviare il segreto in Secrets Manager e quindi fare riferimento al segreto come a un parametro di Parameter Store.

Quando si configura l'archivio parametri con Secrets Manager, l'archivio parametri `secret-id` richiede una barra (/) prima della stringa del nome.

Per ulteriori informazioni, consulta Riferimento AWS Secrets Manager Segreti dei parametri Parameter Store nella AWS Systems Manager Guida dell'utente.

Rotazione dei segreti AWS Secrets Manager

La rotazione è il processo di aggiornamento periodico di un segreto. Quando si ruota un segreto, vengono aggiornati sia il segreto in Secrets Manager che le credenziali del database o del servizio. In Secrets Manager, è possibile impostare la rotazione automatica per i segreti.

Argomenti

- Come funziona la rotazione
- Rotazione gestita per i segreti AWS Secrets Manager
- Impostazione della rotazione automatica per un segreto Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB utilizzando la console
- Imposta la rotazione automatica per i segreti AWS Secrets Manager utilizzando la console
- Impostazione della rotazione automatica per i segreti AWS Secrets Manager utilizzando la AWS CLI
- Rotazione immediata di un segreto AWS Secrets Manager
- AWS Secrets Manager modelli di funzioni di rotazione
- Espressioni di pianificazione nella rotazione di Secrets Manager
- Risolvi i problemi AWS Secrets Manager di rotazione

Come funziona la rotazione

> **Tip**
>
> Per alcuni Segreti gestiti da altri servizi, si utilizza la rotazione gestita. Per utilizzare Rotazione gestita, è necessario dapprima creare il segreto tramite il servizio di gestione.

La rotazione di Secrets Manager utilizza una AWS Lambda funzione per aggiornare il segreto e il database o il servizio. Per ulteriori informazioni sui costi di utilizzo della funzione Lambda, consulta la sezione Prezzi.

Per ruotare un segreto, Secrets Manager chiama una funzione Lambda in base alla pianificazione configurata. Puoi impostare una pianificazione in modo che ruoti dopo un certo periodo di tempo, ad esempio ogni 30 giorni, oppure puoi creare un'espressione cron. Per informazioni, consulta

Espressioni di pianificazione. Se aggiorni anche manualmente il valore segreto mentre è impostata la rotazione automatica, Secrets Manager la considera una rotazione valida quando calcola la data di rotazione successiva.

Per motivi di sicurezza, Secrets Manager consente solo a una funzione di rotazione Lambda di ruotare direttamente il segreto. La funzione di rotazione non può richiamare una seconda funzione Lambda per ruotare il segreto.

Secrets Manager utilizza etichette di gestione temporanea per etichettare le versioni dei segreti durante la rotazione. Durante la rotazione, Secrets Manager chiama la stessa funzione diverse volte, ogni volta con parametri diversi. Secrets Manager richiama la funzione con la struttura di parametri di richiesta JSON seguenti:

```
{
    "Step" : "request.type",
    "SecretId" : "string",
    "ClientRequestToken" : "string"
}
```

La funzione di rotazione fa il lavoro di rotazione del segreto. Tale rotazione avviene in quattro passaggi, che corrispondono a quattro fasi nella funzione di rotazione Lambda:

1. Creazione di una nuova versione del segreto (**`createSecret`**)

   Il primo passo del processo di rotazione consiste nel creare una nuova versione del segreto. Nei modelli di rotazione del database forniti da Secrets Manager, la funzione di rotazione Lambda genera una password di 32 caratteri per la nuova versione. La nuova versione potrà contenere una nuova password, un nuovo nome utente e password o altre informazioni del segreto. La funzione di rotazione Lambda etichetta la nuova versione `AWSPENDING`.
2. Modifica delle credenziali nel database o nel servizio (**`setSecret`**)

   Successivamente, la funzione di rotazione Lambda modifica le credenziali nel database o nel servizio in modo che corrispondano alle nuove credenziali nella versione `AWSPENDING` del segreto. A seconda della strategia di rotazione scelta, questa fase può creare un nuovo utente con le stesse autorizzazioni dell'utente esistente.

   Le funzioni di rotazione per Amazon RDS (tranne Oracle e Db2) e Amazon DocumentDB usano automaticamente Secure Socket Layer (SSL) o Transport Layer Security (TLS) per connettersi al database, se disponibile. Altrimenti usano una connessione non crittografata.

> **Note**
>
> Se hai impostato la rotazione automatica del segreto prima del 20 dicembre 2021, la funzione di rotazione potrebbe essere basata su un modello precedente che non supporta SSL/TLS. Consulta l'articolo Determinare quando è stata creata la funzione di rotazione. Se è stata creata prima del 20 dicembre 2021, per supportare le connessioni che utilizzano SSL/TLS è necessario ricreare la funzione di rotazione.

3. Test della nuova versione del segreto (**`testSecret`**)

   Successivamente, la funzione di rotazione Lambda esegue il test della versione `AWSPENDING` del segreto utilizzando questa versione per accedere al database o al servizio. Le funzioni di rotazione basate su Modelli di funzione di rotazione verificano il nuovo segreto utilizzando un accesso in lettura. A seconda del tipo di accesso necessario alle applicazioni, è possibile aggiornare la funzione in modo da includere altri accessi, ad esempio quello in scrittura.

4. Completamento della rotazione (**`finishSecret`**)

   Infine, la funzione di rotazione Lambda sposta l'etichetta `AWSCURRENT` dalla precedente versione segreta a questa versione, che rimuove anche l'etichetta `AWSPENDING` nella stessa chiamata API. Non dovresti rimuovere `AWSPENDING` prima di questo passaggio e comunque non farlo utilizzando una chiamata API separata, perché ciò può indicare a Secrets Manager che la rotazione non è stata completata correttamente. Secrets Manager aggiunge l'etichetta di gestione temporanea `AWSPREVIOUS` alla versione precedente, in modo da conservare l'ultima versione nota del segreto.

Durante la rotazione, Secrets Manager registra gli eventi che indicano lo stato della rotazione. Per ulteriori informazioni, consulta the section called “Accedi conAWS CloudTrail ”.

Se una qualsiasi fase di rotazione fallisce, Secrets Manager riprova l'intero processo di rotazione più volte.

Quando la rotazione ha esito positivo, l'etichetta di gestione temporanea `AWSPENDING` potrebbe essere collegata alla stessa versione come la versione `AWSCURRENT`, oppure potrebbe non essere collegata a nessuna versione. Se l'etichetta di gestione temporanea `AWSPENDING` è presente ma non è collegata alla stessa versione come `AWSCURRENT`, qualsiasi successiva chiamata di rotazione presuppone che una precedente richiesta di rotazione sia ancora in corso e viene segnalato un errore. Quando la rotazione non ha esito positivo, l'etichetta di gestione temporanea `AWSPENDING`

potrebbe essere collegata a una versione di un segreto vuota. Per ulteriori informazioni, consulta Risoluzione dei problemi della rotazione.

Una volta completata la rotazione, le applicazioni che Recuperare segreti da AWS Secrets Manager da Secrets Manager avranno automaticamente le credenziali aggiornate. Per maggiori dettagli su come funziona ciascuna fase di rotazione, consulta la sezione the section called “Modelli di funzione di rotazione”.

Rotazione gestita per i segreti AWS Secrets Manager

Alcuni servizi offrono la rotazione gestita, in cui il servizio configura e gestisce la rotazione per tuo conto. Con la rotazione gestita, non si utilizza una funzione AWS Lambda per aggiornare il segreto e le credenziali nel database. I seguenti servizi offrono una rotazione gestita:

- Amazon ECS Service Connect offre una rotazione gestita per i certificati AWS Private Certificate Authority TLS. Per ulteriori informazioni, consulta TLS with Service Connect nella Amazon Elastic Container Service Developer Guide.
- Amazon RDS offre una rotazione gestita per le credenziali degli utenti master. Per ulteriori informazioni, consulta la sezione Gestione delle password con Amazon RDS e AWS Secrets Manager nella Guida per l'utente di Amazon RDS.
- Amazon Aurora offre una rotazione gestita per le credenziali degli utenti master. Per ulteriori informazioni, consulta la sezione Gestione delle password con Amazon Aurora e AWS Secrets Manager nella Guida per l'utente di Amazon Aurora.
- Amazon Redshift offre una rotazione gestita per le password di amministratore. Per ulteriori informazioni, consulta Gestione delle password di amministratore Amazon Redshift utilizzando AWS Secrets Manager nella Guida alla gestione di Amazon Redshift.

Per altri tipi di segreti, consulta la sezione Rotazione dei segreti .

La rotazione per i segreti gestiti in genere viene completata entro un minuto. Durante la rotazione, le nuove connessioni che recuperano il segreto potrebbero ottenere la versione precedente delle credenziali. Nelle applicazioni, si consiglia di seguire la best practice di utilizzare un utente del database creato con i privilegi minimi richiesti per l'applicazione, piuttosto che utilizzare l'utente master. Per gli utenti dell'applicazione, ai fini della massima disponibilità, è possibile utilizzare la strategia di rotazione a utenti alternati.

Per modificare la pianificazione della rotazione gestita (tramite console)

1. Apri il segreto gestito nella console di Secrets Manager. Puoi seguire un link dal servizio di gestione o cercare il segreto nella console di Secrets Manager.
2. In Rotation schedule (Pianificazione della rotazione), inserisci la tua pianificazione seguendo il fuso orario UTC nel Schedule expression builder (Generatore di espressioni di pianificazione) o come Schedule expression (Espressione di pianificazione). Gestione dei segreti memorizza la tua pianificazione come un'espressione `rate()` o `cron()`. La finestra di rotazione inizia automaticamente a mezzanotte, a meno che non si specifichi un'ora di inizio. Puoi ruotare un segreto anche ogni quattro ore. Per ulteriori informazioni, consulta Espressioni di pianificazione.
3. (Facoltativo) Per Window duration (Durata della finestra), scegli la lunghezza della finestra durante la quale vuoi che Secrets Manager ruoti il tuo segreto, ad esempio **3h** per una finestra di tre ore. La finestra non deve continuare nella finestra di rotazione successiva. Se non si specifica la durata della finestra, per un programma di rotazione espresso in ore, la finestra si chiude automaticamente dopo un'ora. Per un programma di rotazione espresso in giorni, la finestra si chiude automaticamente alla fine della giornata.
4. Selezionare Salva.

Per modificare la pianificazione della rotazione gestita (AWS CLI)

- Chiama `rotate-secret`. L'esempio seguente mostra come ruotare il segreto tra le 16:00 e le 18:00 UTC del 1° e del 15° giorno del mese. Per ulteriori informazioni, consulta Espressioni di pianificazione.

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

Impostazione della rotazione automatica per un segreto Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB utilizzando la console

La rotazione è il processo di aggiornamento periodico di un segreto. Quando si ruota un segreto, vengono aggiornate le credenziali sia nel segreto che nel database. In Gestione dei segreti, puoi impostare la rotazione automatica per i segreti del tuo database.

Gestione dei segreti utilizza funzioni Lambda per ruotare i segreti. Per una panoramica, consulta the section called “Come funziona la rotazione”.

> **Tip**
>
> Per alcuni Segreti gestiti da altri servizi, si utilizza la rotazione gestita. Per utilizzare Rotazione gestita, è necessario dapprima creare il segreto tramite il servizio di gestione.

Per impostare la rotazione utilizzando la console, prima è necessario scegliere una strategia di rotazione. Poi è possibile configurare il segreto per la rotazione, creando una funzione di rotazione Lambda se non è già presente. La console imposta anche le autorizzazioni per il ruolo di esecuzione della funzione Lambda. L'ultimo passaggio consiste nel garantire che la funzione di rotazione Lambda possa accedere sia a Gestione dei segreti che al database tramite la rete.

Per attivare la rotazione automatica, è necessario disporre dell'autorizzazione per creare il ruolo di esecuzione IAM e collegarvi una policy di autorizzazione. Sono necessarie entrambe le autorizzazioni `iam:CreateRole` e `iam:AttachRolePolicy`.

> **Warning**
>
> Concedendo un'identità a entrambe le autorizzazioni `iam:CreateRole` e `iam:AttachRolePolicy` è possibile concedere qualsiasi autorizzazione all'identità.

Fasi:

- Fase 1: Scelta di una strategia di rotazione e (facoltativamente) creazione di un segreto del superutente
- Fase 2: Configurazione della rotazione e creazione di una funzione di rotazione

- Passaggio 3: (facoltativo) impostazione di condizioni di autorizzazione aggiuntive sulla funzione di rotazione
- Fase 4: Configurazione dell'accesso alla rete per la funzione di rotazione
- Fase 5: (facoltativo) personalizzazione della funzione di rotazione
- Fasi successive

Fase 1: Scelta di una strategia di rotazione e (facoltativamente) creazione di un segreto del superutente

Per Amazon RDS, Amazon Redshift e Amazon DocumentDB, Gestione dei segreti offre due strategie di rotazione:

Strategia di rotazione a utente singolo

Questa strategia aggiorna le credenziali per un utente in un unico segreto. Poiché gli utenti non possono modificare le proprie password, per le istanze Amazon RDS Db2 è necessario fornire le credenziali di amministratore in un segreto separato. Questa è la strategia di rotazione più semplice ed è adatta alla maggior parte dei casi d'uso. In particolare, ti consigliamo di utilizzare questa strategia per le credenziali per utenti occasionali (ad hoc) o interattivi.

Quando il segreto ruota, le connessioni aperte al database non vengono eliminate. Mentre si verifica la rotazione, tra la modifica della password nel database e l'aggiornamento del segreto corrispondente passa un breve periodo di tempo. Durante questo periodo di tempo, c'è un basso rischio che il database neghi le chiamate che utilizzano le credenziali ruotate. È possibile mitigare questo rischio con una strategia per nuovi tentativi appropriata. Dopo la rotazione, le nuove connessioni utilizzano le nuove credenziali.

Strategia di rotazione a utenti alternati

Questa strategia aggiorna le credenziali per due utenti in un unico segreto. Viene creato il primo utente e, durante la prima rotazione, la funzione di rotazione lo clona creando il secondo. Ogni volta che il segreto ruota, la funzione di rotazione alterna quale password dell'utente viene aggiornata. Poiché la maggior parte degli utenti non dispone dell'autorizzazione per clonarsi, è necessario fornire le credenziali relative a un `superuser` in un altro segreto. Si consiglia di utilizzare la strategia di rotazione per singolo utente quando gli utenti clonati nel database non hanno le stesse autorizzazioni dell'utente originale e per le credenziali di utenti occasionali (ad hoc) o interattivi.

Questa strategia è adatta per i database con modelli di autorizzazione in cui un ruolo possiede le tabelle del database e un secondo ruolo ha l'autorizzazione per accedervi. È adatta anche all'uso in applicazioni che richiedono una disponibilità elevata. Se un'applicazione recupera il segreto durante la rotazione, ottiene comunque un set di credenziali valido. Dopo la rotazione, entrambe le credenziali `user` e `user_clone` sono valide. Ci sono anche meno possibilità che le applicazioni ottengano un rifiuto durante questo tipo di rotazione rispetto alla rotazione a utente singolo. Se il database è ospitato in una server farm dove la modifica della password richiede tempo per propagarsi a tutti i server, esiste il rischio che il database rifiuti le chiamate che utilizzano le nuove credenziali. È possibile mitigare questo rischio con una strategia per nuovi tentativi appropriata.

Secrets Manager crea l'utente clonato con le stesse autorizzazioni dell'utente originale. Se modifichi le autorizzazioni dell'utente originale dopo la creazione del clone, devi modificare anche le autorizzazioni dell'utente clonato.

> **Important**
>
> Se scegli la strategia a utenti alternati, è necessario Creazione di un segreto del database e memorizzare al suo interno le credenziali del superutente del database. Con le credenziali di superutente è necessario un segreto perché la rotazione clona il primo utente e la maggior parte degli utenti non dispone di tale autorizzazione.

Fase 2: Configurazione della rotazione e creazione di una funzione di rotazione

Le funzioni di rotazione per Amazon RDS (tranne Oracle e Db2) e Amazon DocumentDB usano automaticamente Secure Socket Layer (SSL) o Transport Layer Security (TLS) per connettersi al database, se disponibile. Altrimenti usano una connessione non crittografata.

Per attivare la rotazione per un segreto Amazon RDS, Amazon DocumentDB o Amazon Redshift

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nella pagina dell'elenco Secrets (Segreti) scegli il segreto.
3. Nella pagina Secret details (Dettagli del segreto), nella sezione Rotation configuration (Configurazione rotazione) scegli Edit rotation (Modifica rotazione).

4. Nella finestra di dialogo Edit rotation configuration (modifica configurazione rotazione), procedi come indicato di seguito:

   a. Attiva Automatic rotation (Rotazione automatica).

   b. In Rotation schedule (Pianificazione della rotazione), inserisci la tua pianificazione seguendo il fuso orario UTC nel Schedule expression builder (Generatore di espressioni di pianificazione) o come Schedule expression (Espressione di pianificazione). Gestione dei segreti memorizza la tua pianificazione come un'espressione `rate()` o `cron()`. La finestra di rotazione inizia automaticamente a mezzanotte, a meno che non si specifichi un'ora di inizio. Puoi ruotare un segreto anche ogni quattro ore. Per ulteriori informazioni, consulta Espressioni di pianificazione.

   c. (Facoltativo) Per Window duration (Durata della finestra), scegli la lunghezza della finestra durante la quale vuoi che Secrets Manager ruoti il tuo segreto, ad esempio **3h** per una finestra di tre ore. La finestra non deve continuare nella finestra di rotazione successiva. Se non si specifica la durata della finestra, per un programma di rotazione espresso in ore, la finestra si chiude automaticamente dopo un'ora. Per un programma di rotazione espresso in giorni, la finestra si chiude automaticamente alla fine della giornata.

   d. (Facoltativo) Scegli Rotate immediately when the secret is stored (Ruota immediatamente quando viene memorizzato il segreto) per ruotare il segreto al salvataggio delle modifiche. Deselezionando la casella di controllo, la prima rotazione inizierà nella pianificazione impostata.

      Se la rotazione non avviene, ad esempio perché le fasi 3 e 4 non sono ancora state completate, Gestione dei segreti riprova il processo di rotazione più volte.

   e. In Rotation function (Funzione di rotazione), esegui una delle seguenti operazioni:

      - Scegli Crea una nuova funzione Lambda e immetti un nome per la nuova funzione. Gestione dei segreti aggiunge "`SecretsManager`" all'inizio del nome della funzione. Gestione dei segreti crea la funzione in base al modello appropriato e imposta le autorizzazioni necessarie per il ruolo di esecuzione di Lambda.
      - Scegli Use an existing Lambda function (Usa una funzione Lambda esistente) per riutilizzare una funzione di rotazione Lambda esistente per un altro segreto. Le funzioni di rotazione elencate in Recommended VPC configurations (Configurazioni VPC consigliate) dispongono dello stesso VPC e gruppo di sicurezza del database, e questo aiuta la funzione ad accedere al database.

f. Per Strategia di rotazione scegli la strategia Utente singolo o Utenti alternati. Per ulteriori informazioni, consulta the section called “Fase 1: Scelta di una strategia di rotazione e (facoltativamente) creazione di un segreto del superutente”.

5. Seleziona Salva.

Passaggio 3: (facoltativo) impostazione di condizioni di autorizzazione aggiuntive sulla funzione di rotazione

Nella policy delle risorse per la funzione di rotazione, si consiglia di includere la chiave di contesto `aws:SourceAccount` per prevenire che Lambda venga usato come confused deputy. Per alcuni servizi AWS, per evitare lo scenario confused deputy, AWS consiglia di utilizzare entrambe le chiavi di condizione globali `aws:SourceArn` e `aws:SourceAccount`. Tuttavia, se includi la condizione `aws:SourceArn` nella tua policy della funzione di rotazione, la funzione di rotazione può essere utilizzata solo per ruotare il segreto specificato da tale ARN. Ti consigliamo di includere solo la chiave di contesto `aws:SourceAccount` in modo da poter utilizzare la funzione di rotazione per più segreti.

Per aggiornare la policy delle risorse della funzione di rotazione

1. Nella console Gestione dei segreti, scegli il tuo segreto e quindi nella pagina dei dettagli, nella sezione Rotation configuration (Configurazione della rotazione), scegli la funzione di rotazione Lambda. Si apre la console Lambda.
2. Segui le istruzioni in Utilizzo di politiche basate sulle risorse per Lambda per aggiungere una condizione `aws:sourceAccount`.

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

Se il segreto è crittografato con una chiave KMS diversa da Chiave gestita da AWS `aws/secretsmanager`, Secrets Manager concede al ruolo di esecuzione Lambda l'autorizzazione a utilizzare la chiave. È possibile utilizzare il contesto di crittografia SecretARN per limitare l'uso della funzione di decrittografia, in modo che il ruolo della funzione di rotazione sia autorizzato ad accedere soltanto per decrittografare il segreto della cui rotazione è responsabile.

Aggiornamento del ruolo di esecuzione della funzione di rotazione

1. Dalla funzione di rotazione Lambda, scegli Configurazione, quindi in Ruolo di esecuzione scegli Nome del ruolo.
2. Segui le istruzioni riportate nella sezione Modifica di una policy sulle autorizzazioni dei ruoli per aggiungere una condizione `kms:EncryptionContext:SecretARN`.

```
"Condition": {
    "StringEquals": {
        "kms:EncryptionContext:SecretARN": "SecretARN"
    }
},
```

Fase 4: Configurazione dell'accesso alla rete per la funzione di rotazione

Per poter ruotare un segreto, la funzione di rotazione Lambda deve essere in grado di accedere sia al segreto che al database o al servizio.

Per accedere a un segreto

La funzione di rotazione Lambda deve essere in grado di accedere a un endpoint di Secrets Manager. Se la funzione Lambda può accedere a Internet, è possibile utilizzare un endpoint pubblico. Per trovare un endpoint, consulta the section called “Endpoint di Secrets Manager”.

Se la funzione Lambda viene eseguita in un VPC che non dispone di accesso a Internet, si consiglia di configurare gli endpoint privati del servizio Secrets Manager all'interno del VPC. Il tuo VPC può quindi intercettare le richieste indirizzate all'endpoint regionale pubblico e reindirizzarle all'endpoint privato. Per ulteriori informazioni, consulta Endpoint VPC.

In alternativa, puoi abilitare la funzione Lambda per accedere a un endpoint pubblico di Gestione dei segreti aggiungendo un gateway NAT o un gateway Internet al VPC che consente al traffico dal tuo VPC di raggiungere l'endpoint pubblico Ciò espone il tuo VPC a un livello di rischio perché vi è un indirizzo IP (per il gateway) che può essere soggetto ad attacchi dalla rete Internet pubblica.

Per accedere al database o al servizio

Se il database o il servizio è in esecuzione su un'istanza Amazon EC2 in un VPC, è consigliabile configurare la funzione Lambda in modo che sia eseguita nello stesso VPC. Quindi la funzione

di rotazione può comunicare direttamente con il servizio. Per ulteriori informazioni, consulta Configurazione dell'accesso VPC.

Per consentire alla funzione Lambda di accedere al database o al servizio, è necessario assicurarsi che i gruppi di sicurezza collegati alla funzione di rotazione Lambda consentano connessioni in uscita al database o al servizio. Inoltre, è necessario accertarsi che i gruppi di sicurezza collegati al database o al servizio consentano le connessioni in entrata dalla funzione di rotazione Lambda.

Per la rotazione alternata degli utenti in cui il segreto del superutente è gestito da un altro servizio AWS, la funzione di rotazione Lambda deve essere in grado di chiamare l'endpoint per ottenere le informazioni sulla connessione al database. Si consiglia di configurare un endpoint VPC per il servizio del database. Per ulteriori informazioni, consulta:

- Endpoint VPC dell'interfaccia e API Amazon RDS nella Guida per l'utente di Amazon RDS.
- Utilizzo degli endpoint VPC nella Guida di gestione di Amazon Redshift.

Fase 5: (facoltativo) personalizzazione della funzione di rotazione

In rari casi, potrebbe essere necessario personalizzare la funzione di rotazione. Ad esempio, con la rotazione alternata degli utenti, Secrets Manager crea l'utente clonato copiando i parametri di configurazione di runtime del primo utente. Se desideri includere più attributi o modificare quelli concessi all'utente clonato, devi aggiornare il codice nella funzione `set_secret`.

Un altro esempio, per Amazon RDS MySQL, Secrets Manager crea un utente clonato con un nome non più lungo di 16 caratteri nella rotazione alternata degli utenti. Puoi modificare la funzione di rotazione per consentire nomi utente più lunghi. La versione 5.7 e successive di MySQL supportano nomi utente fino a 32 caratteri, tuttavia Secrets Manager aggiunge "_clone" (sei caratteri) alla fine del nome utente, quindi è necessario mantenere il nome utente a un massimo di 26 caratteri.

Per aprire la funzione di rotazione Lambda per modificarla

1. Nella console di Gestione dei segreti, scegli il tuo segreto.
2. Nella sezione Rotation configuration (Configurazione rotazione), sotto Lambda rotation function (Funzione di rotazione Lambda), scegli la tua funzione di rotazione.

   Si apre la console Lambda.

   - Per modificare il codice nella funzione, scorri verso il basso fino alla sezione Codice sorgente.

- Per MySQL versione 5.7 e successive, in Variabili di ambiente, modifica `USERNAME_CHARACTER_LIMIT` per alternare la rotazione degli utenti e per modificare la lunghezza massima del nome utente.

Fasi successive

Per informazioni, consultare the section called “Risoluzione dei problemi della rotazione”.

Imposta la rotazione automatica per i segreti AWS Secrets Manager utilizzando la console

La rotazione è il processo di aggiornamento periodico di un segreto. Quando ruoti un segreto, aggiorni le credenziali sia nel segreto che nel database o nel servizio a cui si riferisce il segreto.

Gestione dei segreti utilizza funzioni Lambda per ruotare i segreti. Per una panoramica, consulta the section called “Come funziona la rotazione”.

Puoi anche utilizzare la AWS CLI per impostare la rotazione. Per ulteriori informazioni, consulta Rotazione automatica (AWS CLI).

Per impostare la rotazione utilizzando la console, è necessario innanzitutto configurare il segreto per la rotazione. Durante questa fase, crei anche una funzione di rotazione Lambda vuota. A seguire, imposti le autorizzazioni per la funzione di rotazione e per il ruolo di esecuzione di Lambda. A questo punto, scrivi il codice della funzione di rotazione. L'ultimo passaggio consiste nel garantire che la funzione di rotazione Lambda possa accedere sia a Gestione dei segreti che al database o al servizio tramite la rete.

Per informazioni segrete sul database, consulta the section called “Rotazione automatica per i segreti del database (console)”.

Per attivare la rotazione automatica, è necessario disporre dell'autorizzazione per creare il ruolo di esecuzione IAM e collegarvi una policy di autorizzazione. Sono necessarie entrambe le autorizzazioni `iam:CreateRole` e `iam:AttachRolePolicy`.

> **Warning**
>
> Concedendo un'identità a entrambe le autorizzazioni `iam:CreateRole` e `iam:AttachRolePolicy` è possibile concedere qualsiasi autorizzazione all'identità.

Fasi:

- Fase 1: Configurare il segreto per la rotazione
- Fase 2: Impostare le autorizzazioni per la funzione di rotazione
- Fase 3: (Facoltativa) Impostazione di una condizione di autorizzazione aggiuntiva sulla funzione di rotazione
- Fase 4: Configurazione dell'accesso alla rete per la funzione di rotazione
- Fase 5: Scrittura del codice della funzione di rotazione
- Passaggi successivi

Fase 1: Configurare il segreto per la rotazione

In questa fase, imposti una pianificazione di rotazione per il tuo segreto e crei una funzione di rotazione vuota. Il tuo segreto non verrà ruotato finché non avrai finito di scrivere la funzione di rotazione. Se pianifichi la rotazione prima che la funzione di rotazione venga scritta, o se non viene eseguita per un motivo qualsiasi, Gestione dei segreti riproverà più volte la funzione di rotazione.

Per configurare la rotazione e creare una funzione di rotazione vuota

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Nella pagina dell'elenco Secrets (Segreti) scegli il segreto.
3. Nella pagina Secret details (Dettagli del segreto), nella sezione Rotation configuration (Configurazione rotazione) scegli Edit rotation (Modifica rotazione). Nella finestra di dialogo Edit rotation configuration (modifica configurazione rotazione), procedi come indicato di seguito:

    a. Attiva Automatic rotation (Rotazione automatica).

    b. In Rotation schedule (Pianificazione della rotazione), inserisci la tua pianificazione seguendo il fuso orario UTC nel Schedule expression builder (Generatore di espressioni di pianificazione) o come Schedule expression (Espressione di pianificazione). Gestione dei segreti memorizza la tua pianificazione come un'espressione `rate()` o `cron()`. La finestra di rotazione inizia automaticamente a mezzanotte, a meno che non si specifichi un'ora di inizio. Puoi ruotare un segreto anche ogni quattro ore. Per ulteriori informazioni, consulta Espressioni di pianificazione.

    c. (Facoltativo) Per Window duration (Durata della finestra), scegli la lunghezza della finestra durante la quale vuoi che Secrets Manager ruoti il tuo segreto, ad esempio **3h** per una

finestra di tre ore. La finestra non deve continuare nella finestra di rotazione successiva. Se non si specifica la durata della finestra, per un programma di rotazione espresso in ore, la finestra si chiude automaticamente dopo un'ora. Per un programma di rotazione espresso in giorni, la finestra si chiude automaticamente alla fine della giornata.

d. (Facoltativo) Scegli Rotate immediately when the secret is stored (Ruota immediatamente quando viene memorizzato il segreto) per ruotare il segreto al salvataggio delle modifiche. Deselezionando la casella di controllo, la prima rotazione inizierà nella pianificazione impostata.

e. In Rotation function (Funzione di rotazione), scegli Create function (Crea funzione). La console Lambda si apre in una nuova finestra.

   - Nella console Lambda, nella pagina Create function (Crea funzione), completa una delle seguenti operazioni:
     - Se vedi Browse serverless app repository (Cerca nel repository di app serverless), selezionalo.

       A. In Applicazioni pubbliche, nella casella di ricerca, immettere SecretsManagerRotationTemplate.

       B. Scegli Show apps that create custom IAM roles or resource policies (Visualizza le app che creano ruoli IAM personalizzati o policy delle risorse).

       C. Scegli il SecretsManagerRotationTemplateriquadro.

       D. Nella pagina Review, configure and deploy (Revisione, configurazione e distribuzione), nel riquadro Application settings (Impostazioni dell'applicazione), compila i campi richiesti, quindi scegli Deploy (Distribuisci). Per un elenco di endpoint, consulta the section called “Endpoint di Secrets Manager”.

     - Se non vedi Browse serverless app repository (Cerca nel repository di app serverless), la tua Regione AWS potrebbe non supportare il AWS Serverless Application Repository. Scegli Author from scratch (Crea da zero).

       A. In Function name (Nome funzione), inserisci un nome per la funzione di rotazione.

       B. In Runtime, scegli Python 3.9.

       C. Quando si apre la nuova funzione Lambda, scorri verso il basso per scegliere Configuration (Configurazione), quindi a sinistra scegli Permissions (Autorizzazioni).

D. Scorri verso il basso fino a Policy basata sulle risorse e scegli Aggiungi autorizzazioni per concedere a Gestione dei segreti l'autorizzazione a richiamare la funzione. Per collegare una policy sulle risorse a una funzione Lambda, consulta Utilizzo delle policy basate su risorse per Lambda.

La policy seguente mostra come consentire a Gestione dei segreti di invocare la funzione Lambda.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "secretsmanager.amazonaws.com"
            },
        "Action": "lambda:InvokeFunction",
        "Resource": "LambdaRotationFunctionARN"
    }
    ]
}
```

f. Torna alla console di Gestione dei segreti per collegare la nuova funzione di rotazione al segreto.

g. In Lambda rotation function (Funzione di rotazione Lambda), seleziona il pulsante di aggiornamento. Quindi, nell'elenco di funzioni, scegli la tua nuova funzione.

h. Selezionare Salva.

Fase 2: Impostare le autorizzazioni per la funzione di rotazione

La funzione di rotazione Lambda richiede l'autorizzazione per accedere al segreto in Gestione dei segreti e necessita dell'autorizzazione per accedere al database o al servizio. In questo passaggio, concedi queste autorizzazioni al ruolo di esecuzione di Lambda. Se il segreto è crittografato con una chiave KMS diversa da Chiave gestita da AWS `aws/secretsmanager`, allora è necessario concedere l'autorizzazione per utilizzare la chiave al ruolo di esecuzione Lambda. È possibile utilizzare il contesto di crittografia SecretARN per limitare l'uso della funzione di decrittografia, in modo che il ruolo della funzione di rotazione sia autorizzato ad accedere soltanto per decrittografare

il segreto della cui rotazione è responsabile. Per esempi di policy, consulta Autorizzazioni per la rotazione.

Per istruzioni, consulta Ruolo di esecuzione di Lambda nella Guida per gli sviluppatori di AWS Lambda.

Fase 3: (Facoltativa) Impostazione di una condizione di autorizzazione aggiuntiva sulla funzione di rotazione

Nella policy delle risorse per la funzione di rotazione, si consiglia di includere la chiave di contesto `aws:SourceAccount` per prevenire che Lambda venga usato come confused deputy. Per alcuni servizi AWS, per evitare lo scenario confused deputy, AWS consiglia di utilizzare entrambe le chiavi di condizione globali `aws:SourceArn` e `aws:SourceAccount`. Tuttavia, se includi la condizione `aws:SourceArn` nella tua policy della funzione di rotazione, la funzione di rotazione può essere utilizzata solo per ruotare il segreto specificato da tale ARN. Ti consigliamo di includere solo la chiave di contesto `aws:SourceAccount` in modo da poter utilizzare la funzione di rotazione per più segreti.

Per aggiornare la policy delle risorse della funzione di rotazione

1. Nella console Gestione dei segreti, scegli il tuo segreto e quindi nella pagina dei dettagli, nella sezione Rotation configuration (Configurazione della rotazione), scegli la funzione di rotazione Lambda. Si apre la console Lambda.
2. Segui le istruzioni in Utilizzo di politiche basate sulle risorse per Lambda per aggiungere una condizione `aws:sourceAccount`.

```
"Condition": {
    "StringEquals": {
        "AWS:SourceAccount": "123456789012"
    }
},
```

Fase 4: Configurazione dell'accesso alla rete per la funzione di rotazione

Per poter ruotare un segreto, la funzione di rotazione Lambda deve essere in grado di accedere al segreto. Se il segreto contiene credenziali, la funzione Lambda deve anche potere accedere all'origine di tali credenziali, ad esempio un database o un servizio.

Per accedere a un segreto

La funzione di rotazione Lambda deve essere in grado di accedere a un endpoint di Secrets Manager. Se la funzione Lambda può accedere a Internet, è possibile utilizzare un endpoint pubblico. Per trovare un endpoint, consulta the section called “Endpoint di Secrets Manager”.

Se la funzione Lambda viene eseguita in un VPC che non dispone di accesso a Internet, si consiglia di configurare gli endpoint privati del servizio Secrets Manager all'interno del VPC. Il tuo VPC può quindi intercettare le richieste indirizzate all'endpoint regionale pubblico e reindirizzarle all'endpoint privato. Per ulteriori informazioni, consulta Endpoint VPC.

In alternativa, puoi abilitare la funzione Lambda per accedere a un endpoint pubblico di Gestione dei segreti aggiungendo un gateway NAT o un gateway Internet al VPC che consente al traffico dal tuo VPC di raggiungere l'endpoint pubblico Ciò espone il tuo VPC a un livello di rischio perché vi è un indirizzo IP (per il gateway) che può essere soggetto ad attacchi dalla rete Internet pubblica.

(Facoltativo) Per accedere al database o al servizio

Per segreti come le chiavi API, non è necessario aggiornare il database o il servizio di origine insieme al segreto.

Se il database o il servizio è in esecuzione su un'istanza Amazon EC2 in un VPC, è consigliabile configurare la funzione Lambda in modo che sia eseguita nello stesso VPC. Quindi la funzione di rotazione può comunicare direttamente con il servizio. Per ulteriori informazioni, consulta Configurazione dell'accesso VPC.

Per consentire alla funzione Lambda di accedere al database o al servizio, è necessario assicurarsi che i gruppi di sicurezza collegati alla funzione di rotazione Lambda consentano connessioni in uscita al database o al servizio. Inoltre, è necessario accertarsi che i gruppi di sicurezza collegati al database o al servizio consentano le connessioni in entrata dalla funzione di rotazione Lambda.

Fase 5: Scrittura del codice della funzione di rotazione

La funzione di rotazione creata nella Fase 1 è il punto di partenza della funzione. Scrivi il codice per il tuo caso d'uso specifico. Per una funzione in grado di ruotare un ElastiCache segreto Amazon, puoi copiare il codice dal modello appropriato fornito da Secrets Manager.

Quando scrivi una funzione, fai attenzione a includere istruzioni per la registrazione o il debug. Queste istruzioni possono far sì che le informazioni relative alla tua funzione vengano scritte su Amazon CloudWatch, quindi devi assicurarti che il log non includa informazioni sensibili raccolte durante lo sviluppo.

Per motivi di sicurezza, Secrets Manager consente solo a una funzione di rotazione Lambda di ruotare direttamente il segreto. La funzione di rotazione non può richiamare una seconda funzione Lambda per ruotare il segreto.

Per esempi di istruzioni di log, consulta il codice sorgente the section called “Modelli di funzione di rotazione”.

Se utilizzi binari e librerie esterne, ad esempio per connetterti a una risorsa, devi gestire l'applicazione di patch e la loro conservazione. up-to-date

Per suggerimenti di debug, consulta Esecuzione di test e debug di applicazioni serverless.

Per aprire la funzione di rotazione Lambda per modificarla

1. Nella console di Gestione dei segreti, scegli il tuo segreto.
2. Nella sezione Rotation configuration (Configurazione rotazione), sotto Lambda rotation function (Funzione di rotazione Lambda), scegli la tua funzione di rotazione.

   Si apre la console Lambda.

   - Per modificare il codice nella funzione, scorri verso il basso fino alla sezione Codice sorgente.
   - Per MySQL versione 5.7 e successive, in Variabili di ambiente, modifica `USERNAME_CHARACTER_LIMIT` per alternare la rotazione degli utenti e per modificare la lunghezza massima del nome utente.

Se la tua funzione non lo dispone già, copia il codice da. SecretsManagerRotationTemplate

La rotazione di un segreto consiste in quattro passaggi, che corrispondono ai seguenti quattro metodi di una funzione di rotazione Lambda.

Metodi

- create_secret
- set_secret
- test_secret

- finish_secret

`create_secret`

In `create_secret`, per prima cosa è necessario controllare se esiste un segreto chiamando `get_secret_value` con il `ClientRequestToken` passato. Se non esiste un segreto, crei un nuovo segreto con `create_secret` e il token come `VersionId`. Quindi puoi generare un nuovo valore del segreto con `get_random_password`. È necessario assicurarsi che il nuovo valore del segreto includa solo caratteri validi per il database o il servizio. Escludi i caratteri utilizzando il parametro `ExcludeCharacters`. Chiama `put_secret_value` per archiviarlo con l'etichetta temporanea `AWSPENDING`. La memorizzazione del nuovo valore del segreto in `AWSPENDING` aiuta a garantire l'idempotenza. Se per un motivo qualsiasi la rotazione non viene eseguita, puoi fare riferimento a quel valore del segreto nelle chiamate successive. Consulta Come posso rendere idempotente la mia funzione Lambda.

Mentre testi la tua funzione, usa la AWS CLI per vedere le fasi della versione: chiama `describe-secret` e osserva `VersionIdsToStages`.

`set_secret`

In `set_secret`, modifichi la credenziale nel database o nel servizio in modo che corrisponda al nuovo valore del segreto nella versione `AWSPENDING` del segreto.

Se passi istruzioni a un servizio che interpreta le istruzioni, come un database, utilizza la parametrizzazione delle query. Per ulteriori informazioni, consulta il Cheat Sheet sulla parametrizzazione delle query sul sito Web OWASP.

La funzione di rotazione è un "privileged deputy" che ha l'autorizzazione ad accedere e modificare le credenziali del cliente sia nel segreto di Gestione dei segreti che nella risorsa di destinazione. Per evitare un potenziale attacco confused deputy, devi assicurarti che un utente malintenzionato non possa utilizzare la funzione per accedere ad altre risorse. Prima di aggiornare le credenziali:

- Verifica che la credenziale nella versione `AWSCURRENT` del segreto sia valida. Se la credenziale `AWSCURRENT` non è valida, abbandona il tentativo di rotazione.
- Verifica che i valori dei segreti `AWSCURRENT` e `AWSPENDING` si riferiscano alla stessa risorsa. Per il nome utente e la password, verifica che i nomi utente `AWSCURRENT` e `AWSPENDING` siano uguali.
- Verifica che la risorsa del servizio di destinazione sia la stessa. Per un database, verifica che i nomi degli host `AWSCURRENT` e `AWSPENDING` siano uguali.

`test_secret`

In `test_secret`, esegui il test della versione `AWSPENDING` del segreto utilizzando questa versione per accedere al database o al servizio.

`finish_secret`

In `finish_secret`, utilizzi `update_secret_version_stage` per spostare l'etichetta temporanea `AWSCURRENT` dalla versione segreta precedente alla nuova versione segreta. Gestione dei segreti aggiunge automaticamente l'etichetta di gestione temporanea `AWSPREVIOUS` alla versione precedente, in modo da conservare l'ultima versione nota del segreto.

Passaggi successivi

Per informazioni, consulta the section called “Risoluzione dei problemi della rotazione”.

Impostazione della rotazione automatica per i segreti AWS Secrets Manager utilizzando la AWS CLI

La rotazione è il processo di aggiornamento periodico di un segreto. Quando ruoti un segreto, aggiorni le credenziali sia nel segreto che nel database o nel servizio a cui si riferisce il segreto.

Gestione dei segreti utilizza funzioni Lambda per ruotare i segreti. Per una panoramica, consulta the section called “Come funziona la rotazione”.

Puoi anche utilizzare la console per impostare la rotazione. Per ulteriori informazioni, consulta Rotazione automatica (console).

Per impostare la rotazione utilizzando la AWS CLI, se stai ruotando un segreto Amazon RDS, Amazon Redshift o Amazon DocumentDB, innanzitutto bisogna scegliere una the section called “Strategia di rotazione”. Se scegli la strategia a utenti alternati, è necessario archiviare un segreto separato con credenziali per un superutente del database. A questo punto, puoi scrivere il codice della funzione di rotazione. Gestione dei segreti fornisce modelli su cui puoi basare la funzione. Quindi, crei una funzione Lambda con il tuo codice e imposti le autorizzazioni sia per la funzione Lambda che per il ruolo di esecuzione Lambda. Il passaggio successivo consiste nel garantire che la funzione di rotazione Lambda possa accedere sia a Gestione dei segreti che al database o al servizio tramite la rete. Infine, puoi configurare il segreto per la rotazione.

Per attivare la rotazione automatica, è necessario disporre dell'autorizzazione per creare il ruolo di esecuzione IAM e collegarvi una policy di autorizzazione. Sono necessarie entrambe le autorizzazioni `iam:CreateRole` e `iam:AttachRolePolicy`.

> **Warning**
>
> Concedendo un'identità a entrambe le autorizzazioni `iam:CreateRole` e `iam:AttachRolePolicy` è possibile concedere qualsiasi autorizzazione all'identità.

Fasi:

- (Facoltativa) Fase 1: Creazione di un segreto del superutente
- Fase 2: Scrittura del codice della funzione di rotazione
- Fase 3: Creare la funzione Lambda e il ruolo di esecuzione
- Fase 4: Configurazione dell'accesso alla rete
- Fase 5: Configurare il segreto per la rotazione
- Passaggi successivi

(Facoltativa) Fase 1: Creazione di un segreto del superutente

Per Amazon RDS, Amazon Redshift e Amazon DocumentDB, Gestione dei segreti offre due strategie di rotazione:

Strategia di rotazione a utente singolo

Questa strategia aggiorna le credenziali per un utente in un unico segreto. Poiché gli utenti non possono modificare le proprie password, per le istanze Amazon RDS Db2 è necessario fornire le credenziali di amministratore in un segreto separato. Questa è la strategia di rotazione più semplice ed è adatta alla maggior parte dei casi d'uso. In particolare, ti consigliamo di utilizzare questa strategia per le credenziali per utenti occasionali (ad hoc) o interattivi.

Quando il segreto ruota, le connessioni aperte al database non vengono eliminate. Mentre si verifica la rotazione, tra la modifica della password nel database e l'aggiornamento del segreto corrispondente passa un breve periodo di tempo. Durante questo periodo di tempo, c'è un basso rischio che il database neghi le chiamate che utilizzano le credenziali ruotate. È possibile mitigare questo rischio con una strategia per nuovi tentativi appropriata. Dopo la rotazione, le nuove connessioni utilizzano le nuove credenziali.

Strategia di rotazione a utenti alternati

Questa strategia aggiorna le credenziali per due utenti in un unico segreto. Viene creato il primo utente e, durante la prima rotazione, la funzione di rotazione lo clona creando il secondo. Ogni volta che il segreto ruota, la funzione di rotazione alterna quale password dell'utente viene aggiornata. Poiché la maggior parte degli utenti non dispone dell'autorizzazione per clonarsi, è necessario fornire le credenziali relative a un `superuser` in un altro segreto. Si consiglia di utilizzare la strategia di rotazione per singolo utente quando gli utenti clonati nel database non hanno le stesse autorizzazioni dell'utente originale e per le credenziali di utenti occasionali (ad hoc) o interattivi.

Questa strategia è adatta per i database con modelli di autorizzazione in cui un ruolo possiede le tabelle del database e un secondo ruolo ha l'autorizzazione per accedervi. È adatta anche all'uso in applicazioni che richiedono una disponibilità elevata. Se un'applicazione recupera il segreto durante la rotazione, ottiene comunque un set di credenziali valido. Dopo la rotazione, entrambe le credenziali `user` e `user_clone` sono valide. Ci sono anche meno possibilità che le applicazioni ottengano un rifiuto durante questo tipo di rotazione rispetto alla rotazione a utente singolo. Se il database è ospitato in una server farm dove la modifica della password richiede tempo per propagarsi a tutti i server, esiste il rischio che il database rifiuti le chiamate che utilizzano le nuove credenziali. È possibile mitigare questo rischio con una strategia per nuovi tentativi appropriata.

Secrets Manager crea l'utente clonato con le stesse autorizzazioni dell'utente originale. Se modifichi le autorizzazioni dell'utente originale dopo la creazione del clone, devi modificare anche le autorizzazioni dell'utente clonato.

> **Important**
>
> Se scegli la strategia a utenti alternati, è necessario Creazione di un segreto del database e memorizzare al suo interno le credenziali del superutente del database. Con le credenziali di superutente è necessario un segreto, perché la rotazione clona il primo utente e la maggior parte degli utenti non dispone di tale autorizzazione.

Fase 2: Scrittura del codice della funzione di rotazione

Per ruotare un segreto, hai bisogno di una funzione di rotazione. Una funzione di rotazione è una funzione Lambda chiamata da Gestione dei segreti per ruotare il tuo segreto.

Per una funzione in grado di ruotare un Amazon RDS, Amazon Aurora, Amazon Redshift, Amazon DocumentDB ElastiCache o Amazon secret, puoi copiare il codice dal modello appropriato fornito da Secrets Manager.

Per tutti gli altri tipi di segreti, usa il modello di rotazione generico come punto di partenza per scrivere la tua funzione di rotazione.

Salva la funzione di rotazione in un file ZIP *`my-function.zip`* insieme alle dipendenze richieste.

Quando scrivi una funzione, fai attenzione a includere istruzioni per la registrazione o il debug. Queste istruzioni possono far sì che le informazioni relative alla tua funzione vengano scritte su Amazon CloudWatch, quindi devi assicurarti che il log non includa informazioni sensibili raccolte durante lo sviluppo.

Per motivi di sicurezza, Secrets Manager consente solo a una funzione di rotazione Lambda di ruotare direttamente il segreto. La funzione di rotazione non può richiamare una seconda funzione Lambda per ruotare il segreto.

Per esempi di istruzioni di log, consulta il codice sorgente the section called “Modelli di funzione di rotazione”.

Se utilizzi binari e librerie esterne, ad esempio per connetterti a una risorsa, devi gestire l'applicazione di patch e la loro conservazione. up-to-date

Per suggerimenti di debug, consulta Esecuzione di test e debug di applicazioni serverless.

Per aprire la funzione di rotazione Lambda per modificarla

1. Nella console di Gestione dei segreti, scegli il tuo segreto.
2. Nella sezione Rotation configuration (Configurazione rotazione), sotto Lambda rotation function (Funzione di rotazione Lambda), scegli la tua funzione di rotazione.

   Si apre la console Lambda.

   - Per modificare il codice nella funzione, scorri verso il basso fino alla sezione Codice sorgente.
   - Per MySQL versione 5.7 e successive, in Variabili di ambiente, modifica `USERNAME_CHARACTER_LIMIT` per alternare la rotazione degli utenti e per modificare la lunghezza massima del nome utente.

Se la tua funzione non lo dispone già, copia il codice da. SecretsManagerRotationTemplate

La rotazione di un segreto consiste in quattro passaggi, che corrispondono ai seguenti quattro metodi di una funzione di rotazione Lambda.

Metodi

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

In `create_secret`, per prima cosa è necessario controllare se esiste un segreto chiamando `get_secret_value` con il `ClientRequestToken` passato. Se non esiste un segreto, crei un nuovo segreto con `create_secret` e il token come `VersionId`. Quindi puoi generare un nuovo valore del segreto con `get_random_password`. È necessario assicurarsi che il nuovo valore del segreto includa solo caratteri validi per il database o il servizio. Escludi i caratteri utilizzando il parametro `ExcludeCharacters`. Chiama `put_secret_value` per archiviarlo con l'etichetta temporanea `AWSPENDING`. La memorizzazione del nuovo valore del segreto in `AWSPENDING` aiuta a garantire l'idempotenza. Se per un motivo qualsiasi la rotazione non viene eseguita, puoi fare riferimento a quel valore del segreto nelle chiamate successive. Consulta Come posso rendere idempotente la mia funzione Lambda.

Mentre testi la tua funzione, usa la AWS CLI per vedere le fasi della versione: chiama `describe-secret` e osserva `VersionIdsToStages`.

`set_secret`

In `set_secret`, modifichi la credenziale nel database o nel servizio in modo che corrisponda al nuovo valore del segreto nella versione `AWSPENDING` del segreto.

Se passi istruzioni a un servizio che interpreta le istruzioni, come un database, utilizza la parametrizzazione delle query. Per ulteriori informazioni, consulta il Cheat Sheet sulla parametrizzazione delle query sul sito Web OWASP.

La funzione di rotazione è un "privileged deputy" che ha l'autorizzazione ad accedere e modificare le credenziali del cliente sia nel segreto di Gestione dei segreti che nella risorsa di destinazione. Per evitare un potenziale attacco confused deputy, devi assicurarti che un utente malintenzionato non possa utilizzare la funzione per accedere ad altre risorse. Prima di aggiornare le credenziali:

- Verifica che la credenziale nella versione `AWSCURRENT` del segreto sia valida. Se la credenziale `AWSCURRENT` non è valida, abbandona il tentativo di rotazione.
- Verifica che i valori dei segreti `AWSCURRENT` e `AWSPENDING` si riferiscano alla stessa risorsa. Per il nome utente e la password, verifica che i nomi utente `AWSCURRENT` e `AWSPENDING` siano uguali.
- Verifica che la risorsa del servizio di destinazione sia la stessa. Per un database, verifica che i nomi degli host `AWSCURRENT` e `AWSPENDING` siano uguali.

`test_secret`

In `test_secret`, esegui il test della versione `AWSPENDING` del segreto utilizzando questa versione per accedere al database o al servizio.

`finish_secret`

In `finish_secret`, utilizzi `update_secret_version_stage` per spostare l'etichetta temporanea `AWSCURRENT` dalla versione segreta precedente alla nuova versione segreta. Gestione dei segreti aggiunge automaticamente l'etichetta di gestione temporanea `AWSPREVIOUS` alla versione precedente, in modo da conservare l'ultima versione nota del segreto.

Fase 3: Creare la funzione Lambda e il ruolo di esecuzione

Un ruolo di esecuzione Lambda è un ruolo che Lambda assume quando viene richiamata la funzione.

Creazione di una funzione di rotazione Lambda e di un ruolo di esecuzione

1. Crea una policy di attendibilità per il ruolo di esecuzione Lambda e salvalo come file JSON. Per alcuni esempi, consulta Autorizzazioni per la rotazione. La policy deve:
   - Consentire al ruolo di chiamare le operazioni di Gestione dei segreti sul segreto.
   - Consentire al ruolo di utilizzare la chiave KMS se il segreto è crittografato con una chiave diversa da `aws/secretsmanager`.
   - Consentire al ruolo di chiamare il servizio a cui si riferisce il segreto.
2. Crea il ruolo di esecuzione Lambda e applica la policy di attendibilità chiamando `iam create-role`.

```
aws iam create-role \
    --role-name rotation-lambda-role \
```

```
    --assume-role-policy-document file://trust-policy.json
```

3. (Facoltativo) Per un segreto che contiene credenziali Amazon RDS o Aurora, se utilizzi la strategia degli utenti alternati e il segreto del superutente è gestito da Amazon RDS, devi consentire alla funzione di rotazione di chiamare le API di sola lettura su Amazon RDS in modo che possa ottenere le informazioni di connessione per il database. A tale scopo, collega la policy AWS gestita AmazonRDS ReadOnlyAccess al ruolo di esecuzione della funzione Lambda chiamando. `iam attach-role-policy`

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/AmazonRDSReadOnlyAccess \
    --role-name rotation-lambda-role
```

4. Crea la funzione Lambda dal file ZIP chiamando `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. Imposta una policy delle risorse sulla funzione Lambda per consentire a Gestione dei segreti di richiamarla effettuando una chiamata `lambda add-permission`. Il comando di esempio include `source-account` per evitare che Lambda venga usato come confused deputy.

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

Fase 4: Configurazione dell'accesso alla rete

Per poter ruotare un segreto, la funzione di rotazione Lambda deve essere in grado di accedere sia al segreto che al database o al servizio.

Per accedere a un segreto

La funzione di rotazione Lambda deve essere in grado di accedere a un endpoint di Secrets Manager. Se la funzione Lambda può accedere a Internet, è possibile utilizzare un endpoint pubblico. Per trovare un endpoint, consulta the section called “Endpoint di Secrets Manager”.

Se la funzione Lambda viene eseguita in un VPC che non dispone di accesso a Internet, si consiglia di configurare gli endpoint privati del servizio Secrets Manager all'interno del VPC. Il tuo VPC può quindi intercettare le richieste indirizzate all'endpoint regionale pubblico e reindirizzarle all'endpoint privato. Per ulteriori informazioni, consulta Endpoint VPC.

In alternativa, puoi abilitare la funzione Lambda per accedere a un endpoint pubblico di Gestione dei segreti aggiungendo un gateway NAT o un gateway Internet al VPC che consente al traffico dal tuo VPC di raggiungere l'endpoint pubblico Ciò espone il tuo VPC a un livello di rischio perché vi è un indirizzo IP (per il gateway) che può essere soggetto ad attacchi dalla rete Internet pubblica.

Per accedere al database o al servizio

Se il database o il servizio è in esecuzione su un'istanza Amazon EC2 in un VPC, è consigliabile configurare la funzione Lambda in modo che sia eseguita nello stesso VPC. Quindi la funzione di rotazione può comunicare direttamente con il servizio. Per ulteriori informazioni, consulta Configurazione dell'accesso VPC.

Per consentire alla funzione Lambda di accedere al database o al servizio, è necessario assicurarsi che i gruppi di sicurezza collegati alla funzione di rotazione Lambda consentano connessioni in uscita al database o al servizio. Inoltre, è necessario accertarsi che i gruppi di sicurezza collegati al database o al servizio consentano le connessioni in entrata dalla funzione di rotazione Lambda.

Per la rotazione alternata degli utenti in cui il segreto del superutente è gestito da un altro servizio AWS, la funzione di rotazione Lambda deve essere in grado di chiamare l'endpoint per ottenere le informazioni sulla connessione al database. Si consiglia di configurare un endpoint VPC per il servizio del database. Per ulteriori informazioni, consultare:

- Endpoint VPC dell'interfaccia e API Amazon RDS nella Guida per l'utente di Amazon RDS.
- Utilizzo degli endpoint VPC nella Guida di gestione di Amazon Redshift.

Fase 5: Configurare il segreto per la rotazione

Per attivare la rotazione automatica del segreto, chiama `rotate-secret`. Puoi impostare una pianificazione di rotazione con un'espressione di pianificazione `cron()` o `rate()` e impostare una durata della finestra di rotazione. Puoi ruotare un segreto anche ogni quattro ore. Per ulteriori informazioni, consulta Espressioni di pianificazione.

```
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\"}"
```

Passaggi successivi

Per informazioni, consulta the section called “Risoluzione dei problemi della rotazione”.

Rotazione immediata di un segreto AWS Secrets Manager

È possibile ruotare solo un segreto che ha la rotazione configurata. Per determinare se un segreto è stato configurato per la rotazione, nella console visualizza il segreto e scorri verso il basso fino alla sezione Configurazione della rotazione. Se lo Stato di rotazione è Abilitato, il segreto è configurato per la rotazione. Oppure nella AWS CLI, chiama `describe-secret`. Se la risposta ha un `RotationLambdaARN` e `RotationRules`, il segreto viene configurato per la rotazione. In caso contrario, è possibile impostare la rotazione automatica:

- Rotazione automatica per i segreti del database (console)
- Rotazione automatica (console)
- Rotazione automatica (AWS CLI)

Come ruotare immediatamente un segreto (console)

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli il tuo segreto.
3. Nella pagina dei dettagli del segreto, in Configurazione rotazione, scegli Ruota immediatamente il segreto.

4. Nella finestra di dialogo Ruota segreto, scegli Ruota.

AWS CLI

Example Rotazione immediata di un segreto

L'esempio di `rotate-secret` seguente mostra come avviare una rotazione immediata. L'output mostra la VersionId della nuova versione del segreto creata dalla rotazione. Il segreto deve avere già la rotazione configurata.

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

AWS Secrets Manager modelli di funzioni di rotazione

Secrets Manager fornisce modelli di funzioni di rotazione per:

- Amazon RDS e Amazon Aurora
- Amazon DocumentDB (compatibile con MongoDB)
- Amazon Redshift
- Amazon ElastiCache
- Other type of secret (Altro tipo di segreto)

Per utilizzare il modello, consulta:

- Effettua la rotazione delle credenziali per Amazon RDS, Amazon Aurora, Amazon Redshift o Amazon DocumentDB
- Altri tipi di credenziali (istruzioni della console)
- Altri tipi di credenziali (AWS CLI istruzioni)

I modelli supportano Python 3.9.

Per scrivere una funzione di rotazione personalizzata, consulta Scrivere una funzione di rotazione.

Amazon RDS e Amazon Aurora

Argomenti

- Amazon RDS Db2 per utente singolo
- Amazon RDS Db2 utenti alternati
- Utente singolo di Amazon RDS MariaDB
- Utenti alternativi di Amazon RDS MariaDB
- Utente singolo di Amazon RDS e Amazon Aurora MySQL
- Utenti alternati di Amazon RDS e Amazon Aurora MySQL
- Utente singolo per Amazon RDS Oracle
- Utenti alternati Amazon RDS Oracle
- Utente singolo di Amazon RDS e Amazon Aurora PostgreSQL
- Utenti alternati di Amazon RDS e Amazon Aurora PostgreSQL
- Utente singolo per Amazon RDS Microsoft SQLServer
- Utenti alternativi Amazon RDS Microsoft SQLServer

Amazon RDS Db2 per utente singolo

- Nome del modello: SecretsManager RDSdb2 RotationSingleUser
- Strategia di rotazione: Strategia di rotazione a utente singolo.
- Struttura del **`SecretString`**: the section called “Struttura del segreto di Amazon RDS Db2”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ RDSDb2 /lambda_function.py SecretsManager RotationSingleUser
- Dipendenza: python-ibmdb

Amazon RDS Db2 utenti alternati

- Nome del SecretsManager modello: RDSdb2 RotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`**: the section called “Struttura del segreto di Amazon RDS Db2”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ RDSDb2 /lambda_function.py SecretsManager RotationMultiUser
- Dipendenza: python-ibmdb

Utente singolo di Amazon RDS MariaDB

- Nome del SecretsManager modello: RDSMariaDB RotationSingleUser
- Strategia di rotazione: Strategia di rotazione a utente singolo.
- Struttura del **`SecretString`**: the section called “Struttura del segreto di MariaDB di Amazon RDS”.
- Codice sorgente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSMariaDB / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationSingleUser
- PyMyDipendenza: SQL 1.0.2

Utenti alternativi di Amazon RDS MariaDB

- Nome del modello: RDSMariaDB SecretsManager RotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`**: the section called “Struttura del segreto di MariaDB di Amazon RDS”.
- Codice sorgente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSMariaDB / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationMultiUser
- PyMyDipendenza: SQL 1.0.2

Utente singolo di Amazon RDS e Amazon Aurora MySQL

- Nome del modello: RDSMySQL SecretsManager RotationSingleUser
- Strategia di rotazione: the section called “Utente singolo”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Amazon RDS e Amazon Aurora MySQL”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/master/ RDSMySQL /lambda_function.py SecretsManager RotationSingleUser
- PyMyDipendenza: SQL 1.0.2

Utenti alternati di Amazon RDS e Amazon Aurora MySQL

- Nome del modello: RDSMySQL SecretsManager RotationMultiUser

- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Amazon RDS e Amazon Aurora MySQL”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation -lambdas/tree/ master/ RDSMySQL /lambda_function.py SecretsManager RotationMultiUser
- PyMyDipendenza: SQL 1.0.2

Utente singolo per Amazon RDS Oracle

- Nome del modello: RDS SecretsManager OracleRotationSingleUser
- Strategia di rotazione: the section called “Utente singolo”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Oracle di Amazon RDS”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDS /lambda_function.py OracleRotationSingleUser
- Dipendenza: python-oracledb 2.0.1

Utenti alternati Amazon RDS Oracle

- Nome del modello: RDS SecretsManager OracleRotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Oracle di Amazon RDS”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDS /lambda_function.py OracleRotationMultiUser
- Dipendenza: python-oracledb 2.0.1

Utente singolo di Amazon RDS e Amazon Aurora PostgreSQL

- Nome del modello: RDSpostgreSQL SecretsManager RotationSingleUser
- Strategia di rotazione: Strategia di rotazione a utente singolo.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Amazon RDS e Amazon Aurora PostgreSQL”.

- Codice sorgente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSpostgreSQL / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationSingleUser
- PyGreDipendenza: SQL 5.0.7

Utenti alternati di Amazon RDS e Amazon Aurora PostgreSQL

- Nome del modello: RDSpostgreSQL SecretsManager RotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Amazon RDS e Amazon Aurora PostgreSQL”.
- Codice sorgente: https://github.com/aws-samples/ -lambdas/tree/master/ RDSpostgreSQL / lambda_function.py aws-secrets-manager-rotation SecretsManager RotationMultiUser
- PyGreDipendenza: SQL 5.0.7

Utente singolo per Amazon RDS Microsoft SQLServer

- Nome del modello: RDSSQL SecretsManager ServerRotationSingleUser
- Strategia di rotazione: the section called “Utente singolo”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Microsoft SQLServer di Amazon RDS”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDSSQL /lambda_function.py ServerRotationSingleUser
- Dipendenza: Pymssql 2.2.2

Utenti alternativi Amazon RDS Microsoft SQLServer

- Nome del SecretsManager modello: RDSSQL ServerRotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Microsoft SQLServer di Amazon RDS”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManager -lambdas/tree/master/ RDSSQL /lambda_function.py ServerRotationMultiUser
- Dipendenza: Pymssql 2.2.2

Amazon DocumentDB (compatibile con MongoDB)

Utente singolo Amazon DocumentDB

- Nome SecretsManagerMongo del modello: DB RotationSingleUser
- Strategia di rotazione: the section called “Utente singolo”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Amazon DocumentDB”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerMongo -lambdas/tree/master/ DB /lambda_function.py RotationSingleUser
- Dipendenza: Pymongo 3.2

Utenti alternativi Amazon DocumentDB

- Nome SecretsManagerMongo del modello: DB RotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- Struttura del **`SecretString`** prevista: the section called “Struttura del segreto di Amazon DocumentDB”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerMongo -lambdas/tree/master/ DB /lambda_function.py RotationMultiUser
- Dipendenza: Pymongo 3.2

Amazon Redshift

Amazon Redshift utente singolo

- Nome del modello: SecretsManagerRedshiftRotationSingleUser
- Strategia di rotazione: the section called “Utente singolo”.
- **`SecretString`**Struttura prevista: the section called “Struttura del segreto di Amazon Redshift” othe section called “Struttura segreta di Amazon Redshift Serverless”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerRedshiftRotationSingleUser -lambdas/tree/master/ /lambda_function.py
- PyGreDipendenza: SQL 5.0.7

Utenti alternati Amazon Redshift

- Nome del modello: SecretsManagerRedshiftRotationMultiUser
- Strategia di rotazione: the section called “Utenti alternati”.
- **`SecretString`**Struttura prevista: the section called “Struttura del segreto di Amazon Redshift” othe section called “Struttura segreta di Amazon Redshift Serverless”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerRedshiftRotationMultiUser -lambdas/tree/master/ /lambda_function.py
- PyGreDipendenza: SQL 5.0.7

Amazon ElastiCache

Per utilizzare questo modello, consulta Rotazione automatica delle password per gli utenti nella Amazon ElastiCache User Guide.

- Nome del modello: SecretsManagerElasticacheUserRotation
- Struttura del **`SecretString`** prevista: the section called “Struttura ElastiCache segreta di Amazon”.
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerElasticacheUserRotation -lambdas/tree/master/ /lambda_function.py

Other type of secret (Altro tipo di segreto)

Secrets Manager fornisce questo modello come punto di partenza per creare una funzione di rotazione per qualsiasi tipo di segreto.

- Nome del modello: SecretsManagerRotationTemplate
- Codice sorgente: https://github.com/aws-samples/ aws-secrets-manager-rotation SecretsManagerRotationTemplate -lambdas/tree/master/ /lambda_function.py

Quando scrivi una funzione, fai attenzione a includere istruzioni per la registrazione o il debug. Queste istruzioni possono far sì che le informazioni relative alla tua funzione vengano scritte su Amazon CloudWatch, quindi devi assicurarti che il log non includa informazioni sensibili raccolte durante lo sviluppo.

Per motivi di sicurezza, Secrets Manager consente solo a una funzione di rotazione Lambda di ruotare direttamente il segreto. La funzione di rotazione non può richiamare una seconda funzione Lambda per ruotare il segreto.

Per esempi di istruzioni di log, consulta il codice sorgente the section called “Modelli di funzione di rotazione”.

Se utilizzi binari e librerie esterne, ad esempio per connetterti a una risorsa, devi gestire l'applicazione di patch e la loro conservazione. up-to-date

Per suggerimenti di debug, consulta Esecuzione di test e debug di applicazioni serverless.

La rotazione di un segreto consiste in quattro passaggi, che corrispondono ai seguenti quattro metodi di una funzione di rotazione Lambda.

Metodi

- create_secret
- set_secret
- test_secret
- finish_secret

`create_secret`

In `create_secret`, per prima cosa è necessario controllare se esiste un segreto chiamando `get_secret_value` con il `ClientRequestToken` passato. Se non esiste un segreto, crei un nuovo segreto con `create_secret` e il token come `VersionId`. Quindi puoi generare un nuovo valore del segreto con `get_random_password`. È necessario assicurarsi che il nuovo valore del segreto includa solo caratteri validi per il database o il servizio. Escludi i caratteri utilizzando il parametro `ExcludeCharacters`. Chiama `put_secret_value` per archiviarlo con l'etichetta temporanea `AWSPENDING`. La memorizzazione del nuovo valore del segreto in `AWSPENDING` aiuta a garantire l'idempotenza. Se per un motivo qualsiasi la rotazione non viene eseguita, puoi fare riferimento a quel valore del segreto nelle chiamate successive. Consulta Come posso rendere idempotente la mia funzione Lambda.

Mentre testate la vostra funzione, utilizzate le fasi AWS CLI per vedere la versione: chiamate `describe-secret`e guardate. `VersionIdsToStages`

set_secret

In `set_secret`, modifichi la credenziale nel database o nel servizio in modo che corrisponda al nuovo valore del segreto nella versione `AWSPENDING` del segreto.

Se passi istruzioni a un servizio che interpreta le istruzioni, come un database, utilizza la parametrizzazione delle query. Per ulteriori informazioni, consulta il Cheat Sheet sulla parametrizzazione delle query sul sito Web OWASP.

La funzione di rotazione è un "privileged deputy" che ha l'autorizzazione ad accedere e modificare le credenziali del cliente sia nel segreto di Gestione dei segreti che nella risorsa di destinazione. Per evitare un potenziale attacco confused deputy, devi assicurarti che un utente malintenzionato non possa utilizzare la funzione per accedere ad altre risorse. Prima di aggiornare le credenziali:

- Verifica che la credenziale nella versione `AWSCURRENT` del segreto sia valida. Se la credenziale `AWSCURRENT` non è valida, abbandona il tentativo di rotazione.
- Verifica che i valori dei segreti `AWSCURRENT` e `AWSPENDING` si riferiscano alla stessa risorsa. Per il nome utente e la password, verifica che i nomi utente `AWSCURRENT` e `AWSPENDING` siano uguali.
- Verifica che la risorsa del servizio di destinazione sia la stessa. Per un database, verifica che i nomi degli host `AWSCURRENT` e `AWSPENDING` siano uguali.

test_secret

In `test_secret`, esegui il test della versione `AWSPENDING` del segreto utilizzando questa versione per accedere al database o al servizio.

finish_secret

In `finish_secret`, utilizzi `update_secret_version_stage` per spostare l'etichetta temporanea `AWSCURRENT` dalla versione segreta precedente alla nuova versione segreta. Gestione dei segreti aggiunge automaticamente l'etichetta di gestione temporanea `AWSPREVIOUS` alla versione precedente, in modo da conservare l'ultima versione nota del segreto.

Espressioni di pianificazione nella rotazione di Secrets Manager

Attivando la rotazione automatica, è possibile utilizzare un'espressione cron() o rate() per impostare la pianificazione per la rotazione del segreto. Con un'espressione di frequenza, è possibile creare

un programma di rotazione che si ripete in un intervallo di ore o giorni. Con un'espressione cron, è possibile creare programmi di rotazione più dettagliati di un intervallo di rotazione. I programmi di rotazione di Secrets Manager utilizzano il fuso orario UTC. Puoi ruotare un segreto frequentemente, anche ogni quattro ore. Secrets Manager ruota il tuo segreto in qualsiasi momento durante la finestra di rotazione.

Per attivare la rotazione, consulta:

- the section called “Rotazione automatica per i segreti del database (console)”
- the section called “Rotazione automatica (console)”
- the section called “Rotazione automatica (AWS CLI)”

Espressioni rate

Le espressioni rate di Secrets Manager presentano il formato seguente, dove *`Value`* (Valore) è un numero intero positivo e *`Unit`* (Unità) può essere `hour`, `hours`, `day` o `days`:

```
rate(Value Unit)
```

Puoi ruotare un segreto frequentemente, anche ogni quattro ore. Esempi:

- `rate(4 hours)` significa che il segreto viene ruotato ogni quattro ore.
- `rate(1 day)` significa che il segreto viene ruotato ogni giorno.
- `rate(10 days)` significa che il segreto viene ruotato ogni 10 giorni.

Per una frequenza espressa in ore, la finestra di rotazione predefinita inizia a mezzanotte e si chiude dopo un'ora. Puoi impostare una durata della finestra per modificare la finestra di rotazione. La finestra di rotazione non deve estendersi fino alla finestra di rotazione successiva. Un modo per verificare questo aspetto è controllare che la finestra di rotazione sia inferiore o uguale al numero di ore tra le rotazioni.

Per una frequenza espressa in giorni, la finestra di rotazione predefinita inizia a mezzanotte e si chiude alla fine della giornata. Puoi impostare una durata della finestra per modificare la finestra di rotazione. La finestra di rotazione non deve estendersi nel giorno successivo (in base al fuso UTC). Un modo per verificarlo è confermare che l'ora di inizio sommata alla durata della finestra sia inferiore o uguale a 24 ore.

Espressioni cron

Le espressioni Cron hanno il formato seguente:

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

Un'espressione cron che include incrementi di ore viene ripristinata ogni giorno. Ad esempio, `cron(0 4/12 * * ? *)` significa 04:00, 16:00 e poi il giorno successivo 04:00, 16:00. I programmi di rotazione di Secrets Manager utilizzano il fuso orario UTC.

Per una pianificazione in ore, la finestra di rotazione predefinita si chiude dopo un'ora. Puoi impostare una durata della finestra per modificare la finestra di rotazione. La finestra di rotazione non deve continuare nella finestra di rotazione successiva. Puoi ruotare un segreto anche ogni quattro ore.

Esempio di pianificazione	Espressione
Ogni otto ore a partire da mezzanotte.	`cron(0 /8 * * ? *)`
Ogni otto ore a partire dalle 08:00.	`cron(0 8/8 * * ? *)`
Ogni dieci ore, a partire dalle ore 02:00. Le finestre di rotazione inizieranno alle 02:00, alle 12:00 e alle 22:00 e poi il giorno successivo alle 02:00, alle 12:00 e alle 22:00.	`cron(0 2/10 * * ? *)`
Ogni giorno alle 10:00.	`cron(0 10 * * ? *)`
Ogni sabato alle 18:00.	`cron(0 18 ? * SAT *)`
Il primo giorno di ogni mese alle 8:00.	`cron(0 8 1 * ? *)`
Ogni tre mesi, la prima domenica all'1:00.	`cron(0 1 ? 1/3 SUN#1 *)`
L'ultimo giorno di ogni mese alle 17:00.	`cron(0 17 L * ? *)`
Dal lunedì al venerdì alle 8:00.	`cron(0 8 ? * MON-FRI *)`
Il primo e il quindicesimo giorno di ogni mese alle 16:00.	`cron(0 16 1,15 * ? *)`

Esempio di pianificazione	Espressione
La prima domenica di ogni mese a mezzanotte.	`cron(0 0 ? * SUN#1 *)`

Requisiti di espressione cron in Secrets Manager

Secrets Manager prevede alcune restrizioni su cosa può essere utilizzato per le espressioni cron. Un'espressione cron per Secrets Manager deve avere 0 nel campo dei minuti perché le finestre di rotazione di Secrets Manager iniziano ogni ora. Deve avere * nel campo dell'anno, perché Secrets Manager non supporta programmi di rotazione distanti più di un anno. Nella tabella seguente sono riportate le opzioni che puoi utilizzare.

Campi	Valori	Caratteri jolly
Minuti	Deve essere 0	Nessuno
Ore	0-23	Usa / (barra in avanti) per specificare gli incrementi. Ad esempio, `2/10` significa ogni 10 ore a partire dalle 02:00. Puoi ruotare un segreto anche ogni quattro ore.
Day-of-month (Giorno del mese)	1-31	Usa , (virgola) per includere valori aggiuntivi. Ad esempio, `1,15` indica il primo e il quindicesimo giorno del mese. Usa - (trattino) per specificare un intervallo. Ad esempio, con `1–15` si intendono i giorni dal 1° al 15 del mese. Usa * (asterisco) per includere tutti i valori nel campo. Ad esempio, * significa ogni giorno del mese.

Campi	Valori	Caratteri jolly
		Il carattere jolly ? (punto interrogativo) specifica un valore. Non puoi specificare i campi `Day-of-month` e `Day-of-week` nella stessa espressione cron. Se specifichi un valore in uno dei campi, devi usare un carattere ? nell'altro campo. Usa / (barra in avanti) per specificare gli incrementi. Ad esempio, `1/2` significa ogni due giorni a partire dal giorno 1 (in altre parole, il giorno 1, 3, 5 e così via). Usa L per specificare l'ultimo giorno del mese. Usa ***`DAY`***L per specificare l'ultimo giorno indicato del mese. Ad esempio, `SUNL` significa l'ultima domenica del mese.

Campi	Valori	Caratteri jolly
Mese	1-12 o JAN-DEC	Usa , (virgola) per includere valori aggiuntivi. Ad esempio, `JAN,APR,JUL,OCT` indica gennaio, aprile, luglio e ottobre. Usa - (trattino) per specifica re un intervallo. Ad esempio, `1-3` indica dal 1° al 3° mese dell'anno. Usa * (asterisco) per includere tutti i valori nel campo. Ad esempio, `*` significa ogni mese. Usa / (barra in avanti) per specificare gli incrementi. Ad esempio, `1/3` significa ogni terzo mese, a partire dal 1° mese (in altre parole i mesi 1, 4, 7 e 10).

Campi	Valori	Caratteri jolly
Day-of-week (Giorno della settimana)	1-7 o SUN-SAT	Usa # per specificare un giorno feriale all'interno del mese. Ad esempio, `TUE#3` indica il terzo martedì del mese. Usa , (virgola) per includere valori aggiuntivi. Ad esempio, `1,4` significa il primo e il quarto giorno della settimana. Usa - (trattino) per specificare un intervallo. Ad esempio, `1-4` indica i giorni dal 1° al 4° della settimana. Usa * (asterisco) per includere tutti i valori nel campo. Ad esempio, `*` significa tutti i giorni della settimana. Il carattere jolly ? (punto interrogativo) specifica un valore. Non puoi specificare i campi `Day-of-month` e `Day-of-week` nella stessa espressione cron. Se specifich i un valore in uno dei campi, devi usare un carattere ? nell'altro campo. Usa / (barra in avanti) per specificare gli incrementi. Ad esempio, `1/2` significa ogni secondo giorno della settimana, a partire dal primo

Campi	Valori	Caratteri jolly
		giorno, quindi i giorni 1, 3, 5 e 7. Usa L per specificare l'ultimo giorno della settimana.
Anno	Deve essere *	Nessuno

Risolvi i problemi AWS Secrets Manager di rotazione

Per molti servizi, Secrets Manager utilizza una funzione Lambda per ruotare i segreti. Per ulteriori informazioni, consulta the section called “Come funziona la rotazione”. La funzione di rotazione Lambda interagisce con il database o il servizio a cui appartiene il segreto e con Gestione dei segreti. Quando la rotazione non funziona come previsto, è necessario innanzitutto controllare i CloudWatch registri.

> **Note**
>
> Alcuni servizi possono gestire i segreti per te, tra cui la gestione della rotazione automatica. Per ulteriori informazioni, consulta the section called “Rotazione gestita”.

Per visualizzare i CloudWatch log della funzione Lambda

1. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli il tuo segreto e quindi nella pagina dei dettagli, nella sezione Rotation configuration (Configurazione della rotazione), scegli la funzione di rotazione Lambda. Si apre la console Lambda.
3. Nella scheda Monitor, scegli Registri, quindi scegli Visualizza registri. CloudWatch

   La CloudWatch console si apre e visualizza i registri relativi alla tua funzione.

Interpretazione dei registri

- Nessuna attività dopo "Found credentials in environment variables" (Trovate credenziali nelle variabili di ambiente)
- Nessuna attività dopo "createSecret"
- Errore: "Access to KMS is not allowed"
- Errore: "Key is missing from secret JSON" (Chiave non presente nella struttura JSON del segreto)
- Errore: "setSecret: Unable to log into database" (setSecret: impossibile accedere al database)
- Errore: "Unable to import module 'lambda_function'"
- Aggiornare una funzione di rotazione esistente da Python 3.7 a 3.9

Nessuna attività dopo "Found credentials in environment variables" (Trovate credenziali nelle variabili di ambiente)

Se non si verifica alcuna attività dopo la visualizzazione del messaggio "Found credentials in environment variables" (Trovate credenziali nelle variabili di ambiente) e la durata del processo è lunga (ad esempio, il timeout predefinito di Lambda è 30.000 ms), è possibile che il timeout della funzione Lambda si verifichi durante il tentativo di raggiungere l'endpoint di Gestione dei segreti.

La funzione di rotazione Lambda deve essere in grado di accedere a un endpoint di Secrets Manager. Se la funzione Lambda può accedere a Internet, è possibile utilizzare un endpoint pubblico. Per trovare un endpoint, consulta the section called “Endpoint di Secrets Manager”.

Se la funzione Lambda viene eseguita in un VPC che non dispone di accesso a Internet, si consiglia di configurare gli endpoint privati del servizio Secrets Manager all'interno del VPC. Il tuo VPC può quindi intercettare le richieste indirizzate all'endpoint regionale pubblico e reindirizzarle all'endpoint privato. Per ulteriori informazioni, consulta Endpoint VPC.

In alternativa, puoi abilitare la funzione Lambda per accedere a un endpoint pubblico di Gestione dei segreti aggiungendo un gateway NAT o un gateway Internet al VPC che consente al traffico dal tuo VPC di raggiungere l'endpoint pubblico Ciò espone il tuo VPC a un livello di rischio perché vi è un indirizzo IP (per il gateway) che può essere soggetto ad attacchi dalla rete Internet pubblica.

Nessuna attività dopo "createSecret"

Di seguito sono riportati i problemi che possono causare l'interruzione della rotazione dopo createSecret:

Le liste di controllo degli accessi (ACL) di rete del VPC non consentono il traffico HTTPS in ingresso e in uscita.

Per ulteriori informazioni, consulta Controllo del traffico verso le sottoreti utilizzando ACL di rete nella Guida per l'utente di Amazon VPC.

La configurazione del timeout della funzione Lambda è troppo breve per eseguire il processo.

Per ulteriori informazioni, consulta Configurazione delle opzioni della funzione Lambda nella Guida per gli sviluppatori di AWS Lambda .

L'endpoint VPC di Gestione dei segreti non consente l'instradamento interdominio senza classi (CIDR) del VPC in ingresso nei gruppi di sicurezza assegnati.

Per ulteriori informazioni, consulta Controllo del traffico verso le risorse utilizzando gruppi di sicurezza nella Guida per l'utente di Amazon VPC.

La policy degli endpoint VPC di Gestione dei segreti non consente a Lambda di utilizzare l'endpoint VPC.

Per ulteriori informazioni, consulta Endpoint VPC.

Il segreto utilizza la rotazione alternata degli utenti, il segreto superutente è gestito da Amazon RDS e la funzione Lambda non può accedere all'API RDS.

Per la rotazione alternata degli utenti in cui il segreto del superutente è gestito da un altro AWS servizio, la funzione di rotazione Lambda deve essere in grado di chiamare l'endpoint del servizio per ottenere le informazioni di connessione al database. Si consiglia di configurare un endpoint VPC per il servizio del database. Per ulteriori informazioni, consultare:

- Endpoint VPC dell'interfaccia e API Amazon RDS nella Guida per l'utente di Amazon RDS.
- Utilizzo degli endpoint VPC nella Guida di gestione di Amazon Redshift.

Errore: "Access to KMS is not allowed"

Se ricevi l'errore `ClientError: An error occurred (AccessDeniedException) when calling the GetSecretValue operation: Access to KMS is not allowed`, la funzione di rotazione non è autorizzata a decrittografare il segreto utilizzando la chiave KMS utilizzata per crittografare il segreto. La policy delle autorizzazioni potrebbe contenere una condizione che limita il contesto di crittografia a un segreto specifico. Per informazioni sulle autorizzazioni richieste, consulta la sezione the section called “Istruzione della policy per una chiave gestita dal cliente”.

Errore: "Key is missing from secret JSON" (Chiave non presente nella struttura JSON del segreto)

Una funzione di rotazione Lambda richiede che il valore segreto si trovi in una struttura JSON specifica. Se viene visualizzato questo errore, è possibile che la chiave a cui la funzione di rotazione ha cercato di accedere non sia presente nella struttura JSON. Per informazioni sulla struttura JSON per ogni tipo di segreto, consulta the section called “Struttura JSON di un segreto ”.

Errore: "setSecret: Unable to log into database" (setSecret: impossibile accedere al database)

Di seguito sono riportati i problemi che possono causare questo errore:

La funzione di rotazione non può accedere al database.

- Se la durata del processo è lunga, ad esempio oltre 5.000 ms, la funzione di rotazione Lambda potrebbe non essere in grado di accedere al database tramite la rete.

  Se il database o il servizio è in esecuzione su un'istanza Amazon EC2 in un VPC, è consigliabile configurare la funzione Lambda in modo che sia eseguita nello stesso VPC. Quindi la funzione di rotazione può comunicare direttamente con il servizio. Per ulteriori informazioni, consulta Configurazione dell'accesso VPC.

  Per consentire alla funzione Lambda di accedere al database o al servizio, è necessario assicurarsi che i gruppi di sicurezza collegati alla funzione di rotazione Lambda consentano connessioni in uscita al database o al servizio. Inoltre, è necessario accertarsi che i gruppi di sicurezza collegati al database o al servizio consentano le connessioni in entrata dalla funzione di rotazione Lambda.

Le credenziali nel segreto non sono corrette.

- Se la durata del processo è breve, la funzione di rotazione Lambda potrebbe non essere in grado di autenticarsi con le credenziali segrete. Controlla le credenziali accedendo manualmente con le informazioni contenute nelle `AWSPREVIOUS` versioni `AWSCURRENT` e nelle versioni del segreto utilizzando il comando. AWS CLI `get-secret-value`

Il database utilizza **`scram-sha-256`** per crittografare le password.

- Se il database è Aurora PostgreSQL versione 13 o successiva e questo utilizza `scram-sha-256` per crittografare le password, ma la funzione di rotazione utilizza `libpq` versione 9 o precedente

che non supporta `scram-sha-256`, in questo caso la funzione di rotazione non può connettersi al database.

Per determinare quali utenti del database utilizzano la crittografia **`scram-sha-256`**

- Consulta Checking for users with non-SCRAM passwords (Verifica della presenza di utenti con password non SCRAM) nel blog Autenticazione SCRAM in RDS per PostgreSQL 13.

Per determinare quale versione di **`libpq`** viene utilizzata dalla funzione di rotazione

1. In un computer basato su Linux, sulla console Lambda, accedi alla funzione di rotazione e scarica il pacchetto di implementazione. Decomprimi il file zip in una directory di lavoro.
2. Nella riga di comando, nella directory di lavoro, esegui:

   ```
   readelf -a libpq.so.5 | grep RUNPATH
   ```
3. Se vedi la stringa *`PostgreSQL-9.4.x`*, o qualsiasi versione principale inferiore a 10, la funzione di rotazione non supporta `scram-sha-256`.
   - Output per una funzione di rotazione che non supporta `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 per una funzione di rotazione che supporta `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**
>
> Se hai impostato la rotazione automatica del segreto prima del 30 dicembre 2021, la funzione di rotazione conteneva nel bundle una versione precedente di `libpq` che non supporta `scram-sha-256`. Per supportare `scram-sha-256`, è necessario ricreare la funzione di rotazione.

Il database richiede l'accesso SSL/TLS.

Se il database richiede una connessione SSL/TLS, ma la funzione di rotazione utilizza una connessione non crittografata, in questo caso la funzione di rotazione non può connettersi al database. Le funzioni di rotazione per Amazon RDS (tranne Oracle e Db2) e Amazon DocumentDB usano automaticamente Secure Socket Layer (SSL) o Transport Layer Security (TLS) per connettersi al database, se disponibile. Altrimenti usano una connessione non crittografata.

> **Note**
>
> Se hai impostato la rotazione automatica del segreto prima del 20 dicembre 2021, la funzione di rotazione potrebbe essere basata su un modello precedente che non supporta SSL/TLS. Per supportare le connessioni che utilizzano SSL/TLS è necessario ricreare la funzione di rotazione.

Per determinare quando è stata creata la funzione di rotazione

1. Apri il segreto nella console di Secrets Manager https://console.aws.amazon.com/secretsmanager/. Nella sezione Rotation configuration (Configurazione rotazione), sotto Lambda rotation function (Funzione di rotazione Lambda), viene visualizzato l'ARN della funzione Lambda, ad esempio, `arn:aws:lambda:`*`aws-region`*`:`*`123456789012`*`:function:`*`SecretsManagerMyRotationFunction`*. Copia il nome della funzione dalla fine dell'ARN, in questo esempio *`SecretsManagerMyRotationFunction`* .
2. Nella AWS Lambda console https://console.aws.amazon.com/lambda/, in Funzioni, incolla il nome della funzione Lambda nella casella di ricerca, scegli Invio, quindi scegli la funzione Lambda.

3. Nella pagina dei dettagli della funzione, nella scheda Configuration (Configurazione), in Tag, copia il valore accanto alla chiave aws:cloudformation:stack-name.
4. Nella AWS CloudFormation console https://console.aws.amazon.com/cloudformation, in Stacks, incolla il valore della chiave nella casella di ricerca, quindi scegli Invio.
5. L'elenco degli stack filtra in modo che venga visualizzato solo lo stack che ha creato la funzione di rotazione Lambda. Nella colonna Created date (Data di creazione), visualizza la data di creazione dello stack. Questa è la data di creazione della funzione di rotazione Lambda.

Errore: "Unable to import module 'lambda_function'"

Potresti ricevere questo errore se stai eseguendo una funzione Lambda precedente che è stata aggiornata automaticamente da Python 3.7 a una versione più recente di Python. Per risolvere l'errore, puoi modificare la versione della funzione Lambda in Python 3.7 e quindi the section called "Aggiornare una funzione di rotazione esistente da Python 3.7 a 3.9". Per ulteriori informazioni, consulta la sezione Perché la rotazione della mia funzione Lambda di Secrets Manager non è riuscita con un errore "pg module not found"? in AWS re:Post.

Aggiornare una funzione di rotazione esistente da Python 3.7 a 3.9

Alcune funzioni di rotazione create prima di novembre 2022 utilizzavano Python 3.7. L' AWS SDK per Python ha smesso di supportare Python 3.7 a dicembre 2023. Per ulteriori informazioni, consulta Aggiornamenti delle politiche di supporto di Python per AWS SDK e strumenti. Per passare a una nuova funzione di rotazione che utilizza Python 3.9, puoi aggiungere una proprietà runtime a una funzione di rotazione esistente o ricreare la funzione di rotazione.

Per scoprire quali funzioni di rotazione Lambda usano Python 3.7

1. Accedi AWS Management Console e apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Nell'elenco delle funzioni, filtra per **`SecretsManager`**.
3. Nell'elenco filtrato delle funzioni, in Runtime, cerca Python 3.7.

Per eseguire l'aggiornamento a Python 3.9:

- Opzione 1: ricrea la funzione di rotazione usando AWS CloudFormation

- Opzione 2: aggiorna il runtime per la funzione di rotazione esistente utilizzando AWS CloudFormation
- Opzione 3: per AWS CDK gli utenti, aggiornate la libreria CDK

Opzione 1: ricrea la funzione di rotazione usando AWS CloudFormation

Quando si utilizza la console Secrets Manager per attivare la rotazione, Secrets Manager crea AWS CloudFormation le risorse necessarie, inclusa la funzione di rotazione Lambda. Se hai utilizzato la console per attivare la rotazione o hai creato la funzione di rotazione utilizzando una AWS CloudFormation pila, puoi utilizzare lo stesso AWS CloudFormation stack per ricreare la funzione di rotazione con un nuovo nome. La nuova funzione utilizza la versione più recente di Python.

Per trovare lo AWS CloudFormation stack che ha creato la funzione di rotazione

- Nella pagina dei dettagli della funzione Lambda, nella sezione Configurazione scegli Tag. Visualizza l'ARN accanto a aws:cloudformation:stack-id.

  Il nome dello stack è incorporato nell'ARN, come illustrato nell'esempio seguente.

  - ARN: `arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`
  - Nome stack: **`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**

Per ricreare una funzione di rotazione (AWS CloudFormation)

1. In AWS CloudFormation, cerca lo stack per nome, quindi scegli Aggiorna.

   Se viene visualizzata una finestra di dialogo che consiglia di aggiornare lo stack principale, scegli Vai allo stack principale, quindi scegli Aggiorna.
2. Nella pagina Aggiorna stack, scegli Modifica modello in Designer quindi scegli Visualizza in Designer.
3. Nel designer, nel codice del modello, in `SecretRotationScheduleHostedRotationLambda`, sostituisci il valore per `"functionName": "SecretsManagerTestRotationRDS"` con un nuovo nome di funzione, ad esempio in JSON, **`"functionName": "SecretsManagerTestRotationRDSupdated"`**

4. Continua con il flusso di lavoro AWS CloudFormation dello stack, quindi scegli Invia.

Opzione 2: aggiorna il runtime per la funzione di rotazione esistente utilizzando AWS CloudFormation

Quando si utilizza la console Secrets Manager per attivare la rotazione, Secrets Manager crea AWS CloudFormation le risorse necessarie, inclusa la funzione di rotazione Lambda. Se hai utilizzato la console per attivare la rotazione o hai creato la funzione di rotazione utilizzando uno AWS CloudFormation stack, puoi utilizzare lo stesso AWS CloudFormation stack per aggiornare il runtime della funzione di rotazione.

Per trovare lo AWS CloudFormation stack che ha creato la funzione di rotazione

- Nella pagina dei dettagli della funzione Lambda, nella sezione Configurazione scegli Tag. Visualizza l'ARN accanto a aws:cloudformation:stack-id.

  Il nome dello stack è incorporato nell'ARN, come illustrato nell'esempio seguente.

  - ARN: `arn:aws:cloudformation:us-west-2:408736277230:stack/`**`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**`-3CUDHZMDMBO8/79fc9050-2eef-11ed-`
  - Nome stack: **`SecretsManagerRDSMySQLRotationSingleUser5c2-SecretRotationScheduleHostedRotationLambda`**

Per aggiornare il runtime per una funzione di rotazione (AWS CloudFormation)

1. In AWS CloudFormation, cerca lo stack per nome, quindi scegli Aggiorna.

   Se viene visualizzata una finestra di dialogo che consiglia di aggiornare lo stack principale, scegli Vai allo stack principale, quindi scegli Aggiorna.
2. Nella pagina Aggiorna stack, scegli Modifica modello in Designer quindi scegli Visualizza in Designer.
3. Nel designer, nel modello JSON, per, sotto`SecretRotationScheduleHostedRotationLambda`, sotto`Properties`, aggiungi `Parameters` **`"runtime": "python3.9"`**
4. Continua con il flusso di lavoro AWS CloudFormation dello stack, quindi scegli Invia.

Opzione 3: per AWS CDK gli utenti, aggiornate la libreria CDK

Se hai utilizzato la versione AWS CDK precedente alla v2.94.0 per impostare la rotazione per il tuo segreto, puoi aggiornare la funzione Lambda eseguendo l'aggiornamento alla versione 2.94.0 o successiva. Per ulteriori informazioni, consulta la Guida per gli sviluppatori v2 di AWS Cloud Development Kit (AWS CDK).

Segreti AWS Secrets Manager gestiti da altri servizi AWS

Molti servizi AWS archiviano e utilizzano segreti in AWS Secrets Manager. In alcuni casi, questi segreti sono segreti gestiti, pertanto il servizio che li ha creati aiuta anche a gestirli. Ad esempio, alcuni segreti gestiti includono la rotazione gestita, quindi non richiedono la configurazione della rotazione da parte dell'utente. Il servizio di gestione potrebbe implementare restrizioni che impediscono l'aggiornamento o l'eliminazione immediata dei segreti, al fine di garantire un periodo di recupero. Questa misura aiuta a prevenire interruzioni perché il servizio di gestione dipende dall'utilizzo corretto del segreto.

I segreti gestiti utilizzano una convenzione di denominazione che include l'ID del servizio di gestione per aiutarli a identificarli.

```
Secret name: ServiceID!MySecret
Secret ARN : arn:aws:us-east-1:ServiceID!MySecret-a1b2c3
```

ID per servizi che gestiscono segreti

- `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 “Marketplace AWS”
- `opsworks-cm` – the section called “AWS OpsWorks for Chef Automate”
- `rds` – the section called “Amazon RDS e Aurora”
- `redshift` – the section called “Amazon Redshift”
- `sqlworkbench` – the section called “Editor di query v2 di Amazon Redshift”

Per cercare segreti che vengono gestiti da altri servizi AWS, consulta la sezione Ricerca di segreti gestiti.

Per un elenco completo dei servizi che utilizzano segreti, vederethe section called “AWS servizi che utilizzano AWS Secrets Manager segreti”.

Amazon AppFlow

In Amazon AppFlow, quando configuri un'applicazione SaaS come origine o destinazione, crei una connessione. Ciò include le informazioni necessarie per la connessione alle applicazioni SaaS, come i token di autenticazione, i nomi utente e le password. Amazon AppFlow archivia i dati di connessione in un segreto gestito da Secrets Manager con il prefisso`appflow`. Il costo di archiviazione del segreto è incluso nell'addebito per Amazon AppFlow. Per ulteriori informazioni, consulta la sezione Protezione dei dati in Amazon AppFlow nella Amazon AppFlow User Guide.

AWS Glue DataBrew

AWS Glue DataBrew fornisce i passaggi della ricetta `DETERMINISTIC_DECRYPT`, `DETERMINISTIC_ENCRYPT` e `CRYPTOGRAPHIC_HASH` per eseguire le trasformazioni sulle Informazioni di identificazione personale (PII) in un set di dati, che utilizzano una chiave di crittografia archiviata in un segreto di Secrets Manager. Se utilizzi il segreto DataBrew predefinito per archiviare la chiave di crittografia, DataBrew crea un segreto gestito con il prefisso`databrew`. Il costo di archiviazione del segreto è incluso nel costo di utilizzo DataBrew.

AWS DataSync

Per raccogliere informazioni su un sistema di storage locale, AWS DataSync Discovery utilizza le credenziali per l'interfaccia di gestione del sistema di storage. DataSync memorizza tali credenziali in un segreto gestito da Secrets Manager con il prefisso`datasync`. Viene addebitato il costo per quel segreto. Per ulteriori informazioni, vedere Aggiungere il sistema di storage locale a DataSync Discovery nella Guida per l'AWS DataSyncutente.

AWS Direct Connect

AWS Direct Connect archivia un nome per la chiave dell'associazione di connettività e una coppia di chiavi dell'associazione di connettività (coppia CKN/CAK) in un segreto gestito con il prefisso `directconnect`. Il costo del segreto è incluso nell'addebito per AWS Direct Connect. Per aggiornare il segreto, è necessario utilizzare AWS Direct Connect piuttosto che Secrets Manager. Per ulteriori informazioni, consulta Associazione di una coppia CKN/CAK MacSec a un LAG nella Guida per l'utente di AWS Direct Connect.

Amazon Elastic Container Service

Quando usi Amazon ECS Service Connect, Amazon ECS utilizza i segreti di Secrets Manager per archiviare i certificati AWS Private Certificate Authority TLS. Il costo di archiviazione del segreto è incluso nei costi di Amazon ECS. Per aggiornare il segreto, devi utilizzare Amazon ECS anziché Secrets Manager. Per ulteriori informazioni, consulta TLS with Service Connect nella Amazon Elastic Container Service Developer Guide.

Amazon EventBridge

Quando crei una destinazione EventBridge API Amazon, EventBridge memorizza la relativa connessione in un segreto gestito da Secrets Manager con il prefisso`events`. Il costo di archiviazione del segreto è incluso nell'addebito per l'utilizzo di una destinazione API. Per aggiornare il segreto, è necessario utilizzare EventBridge piuttosto che Secrets Manager. Per ulteriori informazioni, consulta le destinazioni API nella Amazon EventBridge User Guide.

Marketplace AWS

Quando usi Marketplace AWS Quick Launch, Marketplace AWS distribuisce il software insieme alla chiave di licenza. Marketplace AWS archivia la chiave di licenza nel tuo account come segreto gestito da Secrets Manager. Il costo di archiviazione del segreto è incluso nell'addebito per Marketplace AWS. Per aggiornare il segreto, è necessario utilizzare Marketplace AWS piuttosto che Secrets Manager. Per ulteriori informazioni, consulta Configura Quick Launch nella Guida per i rivenditori Marketplace AWS.

AWS OpsWorks for Chef Automate

Quando si crea un nuovo server inAWS OpsWorks CM, OpsWorks CM archivia le informazioni relative al server in un segreto gestito da Secrets Manager con il prefisso`opsworks-cm`. Il costo del segreto è incluso nell'addebito per AWS OpsWorks. Per ulteriori informazioni, consulta Integrazione con AWS Secrets Manager nella Guida per l'utente di AWS OpsWorks.

Amazon RDS e Aurora

Per gestire le credenziali degli utenti principali per Amazon Relational Database Service (Amazon RDS), incluso Aurora, Amazon RDS può creare un segreto gestito appositamente per te. Viene

addebitato il costo per quel segreto. Amazon RDS gestisce anche la rotazione per queste credenziali. Per ulteriori informazioni, consulta le sezioni Password management with Amazon RDS and AWS Secrets Manager (Gestione delle password con Amazon RDS e) nella Guida per l'utente di Amazon RDS e Password management with Amazon Aurora and AWS Secrets Manager (Gestione delle password con Amazon Auroraе) nella Guida per l'utente di Amazon Aurora.

Per altre credenziali Amazon RDS, consulta the section called “Creazione di un segreto del database”.

Amazon Redshift

Al fine di gestire le credenziali di amministratore per Amazon Redshift, Amazon Redshift può creare per te un segreto gestito. Viene addebitato il costo per quel segreto. Amazon Redshift gestisce anche la rotazione per queste credenziali. Per ulteriori informazioni, consulta Gestione delle password di amministratore Amazon Redshift utilizzando AWS Secrets Manager nella Guida alla gestione di Amazon Redshift.

Per altre credenziali Amazon Redshift, consulta the section called “Creazione di un segreto del database”. Per utilizzare un segreto per le credenziali quando chiami l'API dati, consulta Utilizzo dell’API per i dati di Amazon Redshift. Per utilizzare di un segreto durante l'utilizzo dell'editor di query Amazon Redshift al fine di connetterti a un database, consulta Esecuzione di query in un database con l'editor della query nella Guida alla gestione di Amazon Redshift e the section called “Editor di query v2 di Amazon Redshift”.

Editor di query v2 di Amazon Redshift

Quando utilizzi l'editor di query Amazon Redshift v2 per connetterti a un database, Amazon Redshift può archiviare le credenziali in un segreto gestito di Secrets Manager con il prefisso `sqlworkbench`. Il costo di archiviazione del segreto è incluso nell'addebito per Amazon Redshift. Per aggiornare il segreto, è necessario utilizzare Amazon Redshift piuttosto che Secrets Manager. Per informazioni, consulta Utilizzo dell'editor di query v2 nella Guida alla gestione di Amazon Redshift.

Utilizzo di un endpoint VPC AWS Secrets Manager

Consigliamo di eseguire la maggior parte delle infrastrutture su reti private non accessibili da Internet pubblico. È possibile stabilire una connessione privata tra il VPC e Secrets Manager creando un endpoint VPC dell'interfaccia. Gli endpoint dell'interfaccia sono basati su AWS PrivateLink, una tecnologia che consente di accedere privatamente alle API di Secrets Manager senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione AWS Direct Connect. Le istanze presenti nel VPC non richiedono indirizzi IP pubblici per comunicare con le API di Secrets Manager. Il traffico tra il VPC e Secrets Manager non esce dalla rete AWS. Per ulteriori informazioni, consultare Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida per l'utente di Amazon VPC.

Quando Secrets Manager effettua una rotazione del segreto utilizzando una funzione di rotazione Lambda, ad esempio un segreto che contiene credenziali del database, la funzione Lambda effettua richieste sia al database che a Secrets Manager. Quando si attiva la rotazione automatica utilizzando la console, Secrets Manager crea la funzione Lambda nello stesso VPC del database. Suggeriamo di creare un endpoint di Secrets Manager nello stesso VPC in modo che le richieste dalla funzione di rotazione Lambda a Secrets Manager non escano dalla rete Amazon.

Se si abilita il DNS privato per l'endpoint, è possibile effettuare richieste API verso Secrets Manager utilizzando il nome DNS predefinito per la regione, ad esempio `secretsmanager.us-east-1.amazonaws.com`. Per ulteriori informazioni, consulta Accesso a un servizio tramite un endpoint dell'interfaccia in Guida per l'utente di Amazon VPC.

È possibile assicurarsi che le richieste a Secrets Manager provengano dall’accesso VPC includendo una condizione nelle policy di autorizzazione. Per ulteriori informazioni, consulta the section called “Esempio: autorizzazioni e VPC”.

È possibile utilizzare i log AWS CloudTrail di audit per l'utilizzo dei segreti tramite l'endpoint VPC.

Creare un endpoint VPC privato di Secrets Manager

1. Consulta Creazione di un endpoint di interfaccia nella Amazon VPC User Guide. Usa il nome di servizio: com.amazonaws.*region*.secretsmanager
2. Per controllare l'accesso all'endpoint, consulta Controllare l'accesso agli endpoint VPC utilizzando le policy degli endpoint.

Sottoreti condivise

Non puoi creare, descrivere, modificare o eliminare gli endpoint VPC nelle sottoreti condivise con te. Tuttavia, puoi utilizzare gli endpoint VPC in sottoreti condivise con te. Per informazioni sulla condivisione VPC, consulta Condivisione del VPC con altri account nella Guida per l'utente di Amazon Virtual Private Cloud.

Creazione di segreti AWS Secrets Manager in AWS CloudFormation

È possibile creare segreti in uno stack CloudFormation utilizzando la risorsa `AWS::SecretsManager::Secret` in un modello CloudFormation come mostrato in Creazione di un segreto.

Per creare un segreto di amministrazione per Amazon RDS o Aurora, ti consigliamo di utilizzare `ManageMasterUserPassword` in `AWS::RDS::DBCluster`. Quindi Amazon RDS crea il segreto e gestisce la rotazione per te. Per ulteriori informazioni, consulta Rotazione gestita.

Per le credenziali Amazon Redshift e Amazon DocumentDB, innanzitutto crea un segreto con una password generata da Secrets Manager e quindi utilizza un riferimento dinamico per recuperare il nome utente e la password dal segreto da utilizzare come credenziali per un nuovo database. Quindi, usa la risorsa `AWS::SecretsManager::SecretTargetAttachment` per aggiungere dettagli sul database al segreto necessari a Secrets Manager per ruotare il segreto. Infine, per attivare la rotazione automatica, utilizza la risorsa `AWS::SecretsManager::RotationSchedule` e fornisci una funzione di rotazione e una pianificazione. Fare riferimento agli esempi riportati di seguito:

- Crea un segreto con le credenziali Amazon Redshift
- Crea un segreto con le credenziali Amazon DocumentDB

Per allegare una policy delle risorse al segreto, usa la risorsa `AWS::SecretsManager::ResourcePolicy`.

Per informazioni su come creare risorse con AWS CloudFormation, consulta Informazioni sulle nozioni di base dei modelli nella Guida per l'utente di AWS CloudFormation. Puoi anche utilizzare l'AWS Cloud Development Kit (AWS CDK). Per ulteriori informazioni, consulta AWS Secrets Manager Libreria Construct.

Creazione di un segreto AWS Secrets Manager con AWS CloudFormation

Questo esempio crea un segreto denominato **CloudFormationCreatedSecret-*a1b2c3d4e5f6***. Il valore del segreto è il seguente JSON, con una password di 32 caratteri che viene generata al momento della creazione del segreto.

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

In questo esempio viene utilizzata la seguente risorsa CloudFormation:

- AWS::SecretsManager::Secret

Per informazioni su come creare risorse con AWS CloudFormation, consulta Informazioni sulle nozioni di base dei modelli nella Guida per l'utente di AWS CloudFormation.

JSON

```
{
    "Resources": {
        "CloudFormationCreatedSecret": {
            "Type": "AWS::SecretsManager::Secret",
            "Properties": {
                "Description": "Simple secret created by AWS CloudFormation.",
                "GenerateSecretString": {
                    "SecretStringTemplate": "{\"username\": \"saanvi\"}",
                    "GenerateStringKey": "password",
                    "PasswordLength": 32
                }
            }
        }
    }
}
```

YAML

```
Resources:
  CloudFormationCreatedSecret:
    Type: 'AWS::SecretsManager::Secret'
    Properties:
      Description: Simple secret created by AWS CloudFormation.
      GenerateSecretString:
        SecretStringTemplate: '{"username": "saanvi"}'
        GenerateStringKey: password
        PasswordLength: 32
```

Creare un segreto AWS Secrets Manager con rotazione automatica e un'istanza database MySQL di Amazon RDS con AWS CloudFormation

Per creare un segreto di amministrazione per Amazon RDS o Aurora, ti consigliamo di utilizzare `ManageMasterUserPassword`, come mostrato nell'esempio Crea un segreto Secrets Manager per una password principale in `AWS::RDS::DBCluster`. Quindi Amazon RDS crea il segreto e gestisce la rotazione per te. Per ulteriori informazioni, consulta Rotazione gestita.

Crea un cluster AWS Secrets Manager segreto e un cluster Amazon Redshift con AWS CloudFormation

Per creare un segreto amministrativo per Amazon Redshift, ti consigliamo di utilizzare gli esempi su `AWS::Redshift::Cluster`e. `AWS::RedshiftServerless::Namespace`

Crea un segreto AWS Secrets Manager e un'istanza Amazon DocumentDB con AWS CloudFormation

In questo esempio vengono creati un segreto e un'istanza Amazon DocumentDB che utilizzano le credenziali nel segreto come utente e password. Il segreto ha una policy basata sulle risorse collegata che definisce chi può accedere al segreto. Il modello, inoltre, crea una funzione di rotazione Lambda e configura il segreto da Modelli di funzione di rotazione affinché ruoti automaticamente tra le 8:00 e le 10:00 UTC il primo giorno di ogni mese. Come best practice di sicurezza, l'istanza si trova in un Amazon VPC.

In questo esempio vengono utilizzate le seguenti risorse CloudFormation per Secrets Manager:

- AWS::SecretsManager::Secret
- AWS::SecretsManager::SecretTargetAttachment
- AWS::SecretsManager::RotationSchedule

Per informazioni su come creare risorse con AWS CloudFormation, consulta Informazioni sulle nozioni di base dei modelli nella Guida per l'utente di AWS CloudFormation.

JSON

```
{
   "AWSTemplateFormatVersion":"2010-09-09",
   "Transform":"AWS::SecretsManager-2020-07-23",
   "Resources":{
      "TestVPC":{
         "Type":"AWS::EC2::VPC",
         "Properties":{
            "CidrBlock":"10.0.0.0/16",
            "EnableDnsHostnames":true,
            "EnableDnsSupport":true
         }
      },
      "TestSubnet01":{
         "Type":"AWS::EC2::Subnet",
         "Properties":{
            "CidrBlock":"10.0.96.0/19",
            "AvailabilityZone":{
               "Fn::Select":[
                  "0",
                  {
                     "Fn::GetAZs":{
                        "Ref":"AWS::Region"
                     }
                  }
               ]
            },
            "VpcId":{
               "Ref":"TestVPC"
            }
         }
```

```
      },
      "TestSubnet02":{
         "Type":"AWS::EC2::Subnet",
         "Properties":{
            "CidrBlock":"10.0.128.0/19",
            "AvailabilityZone":{
               "Fn::Select":[
                  "1",
                  {
                     "Fn::GetAZs":{
                        "Ref":"AWS::Region"
                     }
                  }
               ]
            },
            "VpcId":{
               "Ref":"TestVPC"
            }
         }
      },
      "SecretsManagerVPCEndpoint":{
         "Type":"AWS::EC2::VPCEndpoint",
         "Properties":{
            "SubnetIds":[
               {
                  "Ref":"TestSubnet01"
               },
               {
                  "Ref":"TestSubnet02"
               }
            ],
            "SecurityGroupIds":[
               {
                  "Fn::GetAtt":[
                     "TestVPC",
                     "DefaultSecurityGroup"
                  ]
               }
            ],
            "VpcEndpointType":"Interface",
            "ServiceName":{
               "Fn::Sub":"com.amazonaws.${AWS::Region}.secretsmanager"
            },
            "PrivateDnsEnabled":true,
```

```
            "VpcId":{
               "Ref":"TestVPC"
            }
         }
      },
      "MyDocDBClusterRotationSecret":{
         "Type":"AWS::SecretsManager::Secret",
         "Properties":{
            "GenerateSecretString":{
               "SecretStringTemplate":"{\"username\": \"someadmin\",\"ssl\": true}",
               "GenerateStringKey":"password",
               "PasswordLength":16,
               "ExcludeCharacters":"\"@/\\"
            },
            "Tags":[
               {
                  "Key":"AppName",
                  "Value":"MyApp"
               }
            ]
         }
      },
      "MyDocDBCluster":{
         "Type":"AWS::DocDB::DBCluster",
         "Properties":{
            "DBSubnetGroupName":{
               "Ref":"MyDBSubnetGroup"
            },
            "MasterUsername":{
               "Fn::Sub":"{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::username}}"
            },
            "MasterUserPassword":{
               "Fn::Sub":"{{resolve:secretsmanager:
${MyDocDBClusterRotationSecret}::password}}"
            },
            "VpcSecurityGroupIds":[
               {
                  "Fn::GetAtt":[
                     "TestVPC",
                     "DefaultSecurityGroup"
                  ]
               }
            ]
```

```
        }
      },
      "DocDBInstance":{
        "Type":"AWS::DocDB::DBInstance",
        "Properties":{
          "DBClusterIdentifier":{
            "Ref":"MyDocDBCluster"
          },
          "DBInstanceClass":"db.r5.large"
        }
      },
      "MyDBSubnetGroup":{
        "Type":"AWS::DocDB::DBSubnetGroup",
        "Properties":{
          "DBSubnetGroupDescription":"",
          "SubnetIds":[
            {
              "Ref":"TestSubnet01"
            },
            {
              "Ref":"TestSubnet02"
            }
          ]
        }
      },
      "SecretDocDBClusterAttachment":{
        "Type":"AWS::SecretsManager::SecretTargetAttachment",
        "Properties":{
          "SecretId":{
            "Ref":"MyDocDBClusterRotationSecret"
          },
          "TargetId":{
            "Ref":"MyDocDBCluster"
          },
          "TargetType":"AWS::DocDB::DBCluster"
        }
      },
      "MySecretRotationSchedule":{
        "Type":"AWS::SecretsManager::RotationSchedule",
        "DependsOn":"SecretDocDBClusterAttachment",
        "Properties":{
          "SecretId":{
            "Ref":"MyDocDBClusterRotationSecret"
          },
```

```
            "HostedRotationLambda":{
               "RotationType":"MongoDBSingleUser",
               "RotationLambdaName":"MongoDBSingleUser",
               "VpcSecurityGroupIds":{
                  "Fn::GetAtt":[
                     "TestVPC",
                     "DefaultSecurityGroup"
                  ]
               },
               "VpcSubnetIds":{
                  "Fn::Join":[
                     ",",
                     [
                        {
                           "Ref":"TestSubnet01"
                        },
                        {
                           "Ref":"TestSubnet02"
                        }
                     ]
                  ]
               }
            },
            "RotationRules":{
              "Duration": "2h",
              "ScheduleExpression": "cron(0 8 1 * ? *)"
            }
         }
      }
   }
}
```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::SecretsManager-2020-07-23
Resources:
  TestVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
```

```
  TestSubnet01:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.96.0/19
      AvailabilityZone:
        Fn::Select:
        - '0'
        - Fn::GetAZs:
            Ref: AWS::Region
      VpcId:
        Ref: TestVPC
  TestSubnet02:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.128.0/19
      AvailabilityZone:
        Fn::Select:
        - '1'
        - Fn::GetAZs:
            Ref: AWS::Region
      VpcId:
        Ref: TestVPC
  SecretsManagerVPCEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      SubnetIds:
      - Ref: TestSubnet01
      - Ref: TestSubnet02
      SecurityGroupIds:
      - Fn::GetAtt:
        - TestVPC
        - DefaultSecurityGroup
      VpcEndpointType: Interface
      ServiceName:
        Fn::Sub: com.amazonaws.${AWS::Region}.secretsmanager
      PrivateDnsEnabled: true
      VpcId:
        Ref: TestVPC
  MyDocDBClusterRotationSecret:
    Type: AWS::SecretsManager::Secret
    Properties:
      GenerateSecretString:
        SecretStringTemplate: '{\"username\": \"someadmin\",\"ssl\": true}'
        GenerateStringKey: password
```

```
        PasswordLength: 16
        ExcludeCharacters: "\"@/\\"
      Tags:
      - Key: AppName
        Value: MyApp
  MyDocDBCluster:
    Type: AWS::DocDB::DBCluster
    Properties:
      DBSubnetGroupName:
        Ref: MyDBSubnetGroup
      MasterUsername:
        Fn::Sub: "{{resolve:secretsmanager:${MyDocDBClusterRotationSecret}::username}}"
      MasterUserPassword:
        Fn::Sub: "{{resolve:secretsmanager:${MyDocDBClusterRotationSecret}::password}}"
      VpcSecurityGroupIds:
      - Fn::GetAtt:
        - TestVPC
        - DefaultSecurityGroup
  DocDBInstance:
    Type: AWS::DocDB::DBInstance
    Properties:
      DBClusterIdentifier:
        Ref: MyDocDBCluster
      DBInstanceClass: db.r5.large
  MyDBSubnetGroup:
    Type: AWS::DocDB::DBSubnetGroup
    Properties:
      DBSubnetGroupDescription: ''
      SubnetIds:
      - Ref: TestSubnet01
      - Ref: TestSubnet02
  SecretDocDBClusterAttachment:
    Type: AWS::SecretsManager::SecretTargetAttachment
    Properties:
      SecretId:
        Ref: MyDocDBClusterRotationSecret
      TargetId:
        Ref: MyDocDBCluster
      TargetType: AWS::DocDB::DBCluster
  MySecretRotationSchedule:
    Type: AWS::SecretsManager::RotationSchedule
    DependsOn: SecretDocDBClusterAttachment
    Properties:
      SecretId:
```

```
      Ref: MyDocDBClusterRotationSecret
  HostedRotationLambda:
    RotationType: MongoDBSingleUser
    RotationLambdaName: MongoDBSingleUser
    VpcSecurityGroupIds:
      Fn::GetAtt:
      - TestVPC
      - DefaultSecurityGroup
    VpcSubnetIds:
      Fn::Join:
      - ","
      - - Ref: TestSubnet01
        - Ref: TestSubnet02
  RotationRules:
    Duration: 2h
    ScheduleExpression: 'cron(0 8 1 * ? *)'
```

Come Secrets Manager utilizza AWS CloudFormation

Quando utilizzi la console per attivare la rotazione, Secrets Manager utilizza AWS CloudFormation per creare le risorse per la rotazione. Se crei una nuova funzione di rotazione durante quel processo, AWS CloudFormation crea un `AWS::Serverless::Function` basato sui Modelli di funzione di rotazione appropriati. Quindi AWS CloudFormation imposta la `RotationSchedule`, che a sua volta imposta la funzione di rotazione e i file di rotazione per il segreto. Puoi visualizzare lo stack AWS CloudFormation scegliendo View stack (Visualizza stack) nel banner dopo aver attivato la rotazione automatica.

Per informazioni sull'attivazione della rotazione automatica, consulta Rotazione dei segreti .

Creazione di segreti AWS Secrets Manager in AWS Cloud Development Kit (AWS CDK)

Per creare, gestire e recuperare segreti in un'app CDK, puoi utilizzare la Libreria dei costrutti di AWS Secrets Manager, che contiene costrutti `ResourcePolicy`, `RotationSchedule`, `Secret`, `SecretRotation` e `SecretTargetAttachment`.

Per degli esempi, consulta:

- Creazione di un segreto
- Importazione di un segreto
- Recupero di un segreto
- Concessione del permesso di utilizzare il segreto
- Rotazione di un segreto
- Rotazione di un segreto del database
- Replica un segreto in altre regioni

Per ulteriori informazioni sulla CDK, consulta la Guida per gli sviluppatori di AWS Cloud Development Kit (AWS CDK) v2.

Monitorare i segreti AWS Secrets Manager

AWS fornisce i seguenti strumenti di monitoraggio per controllare i segreti di Secret Manager, segnalare un problema e intervenire automaticamente quando necessario. Puoi usare il registro per analizzare qualsiasi modifica o utilizzo imprevisto e le modifiche indesiderate possono essere annullate. È inoltre possibile impostare dei controlli automatici per l'uso inappropriato dei segreti e per qualsiasi tentativo di eliminarli.

Argomenti

- Log di eventi di AWS Secrets Manager con AWS CloudTrail
- Abbina AWS Secrets Manager gli eventi con Amazon EventBridge
- Monitora AWS Secrets Manager con Amazon CloudWatch
- Monitoraggio dei segreti AWS Secrets Manager pianificati per l'eliminazione utilizzando Amazon CloudWatch

Log di eventi di AWS Secrets Manager con AWS CloudTrail

AWS CloudTrail registra tutte le chiamate API per Secrets Manager come eventi, incluse le chiamate dalla console Secrets Manager e diversi altri eventi per la rotazione e l'eliminazione delle versioni dei segreti. Per un elenco delle voci del file di log dei record di Secrets Manager, vedereCloudTrail voci.

È possibile utilizzare la CloudTrail console per visualizzare gli ultimi 90 giorni di eventi registrati. Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Secrets Manager, crea un percorso in modo che CloudTrail invii i file di registro a un bucket Amazon S3. Consultare Creazione di un percorso per il tuo account AWS. Puoi anche configurare la ricezione CloudTrail di file di CloudTrail registro da più Account AWS e. Regioni AWS

È possibile configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati raccolti nei CloudTrail log. Visualizza le integrazioni AWS dei servizi con CloudTrail i log. Puoi anche ricevere notifiche quando CloudTrail pubblica nuovi file di log nel tuo bucket Amazon S3. Consulta Configurazione delle notifiche Amazon SNS per. CloudTrail

Per recuperare gli eventi di Secrets Manager dai CloudTrail registri (console)

1. Apri la CloudTrail console all'indirizzo https://console.aws.amazon.com/cloudtrail/.

2. Assicurati che la console punti alla regione in cui si sono verificati gli eventi. La console mostra solo gli eventi che si sono verificati nella regione selezionata. Scegli la regione dall'elenco a discesa nell'angolo in alto a destra della console.
3. Nel riquadro di navigazione a sinistra, seleziona Event history (Cronologia eventi).
4. Scegli i criteri di Filter (Filtro) e/o un valore per Time range (Intervallo di tempo) per individuare l'evento che stai cercando. Ad esempio, per visualizzare tutti gli eventi Secrets Manager per Select attribute (Seleziona attributo), scegli Event source (Origine eventi). Quindi per Enter event source (Inserisci origine evento) scegli **`secretsmanager.amazonaws.com`**.
5. Per ulteriori dettagli, scegli la freccia per l'espansione accanto all'evento. Per visualizzare tutte le informazioni disponibili, scegli View event (Visualizza evento).

AWS CLI

Example Recupera gli eventi di Secrets Manager dai registri CloudTrail

L'esempio di `lookup-events` seguente mostra la ricerca degli eventi di Secrets Manager.

```
aws cloudtrail lookup-events \
    --region us-east-1 \
    --lookup-attributes
 AttributeKey=EventSource,AttributeValue=secretsmanager.amazonaws.com
```

Voci di AWS CloudTrail per il Secrets Manager

AWS Secrets Manager scrive le voci nel tuo file di log AWS CloudTrail per tutte le operazioni di Secrets Manager e per altri eventi relativi alla rotazione e alla cancellazione. Per informazioni su come intervenire su questi eventi, consulta Abbina gli eventi di Secrets Manager con EventBridge.

Tipi di voci del file di log

- Voce del file di log per le operazioni di Secrets Manager
- Registra le voci del file di log da eliminare
- Voci di log per la replica
- Registra le voci del file di log per la rotazione

Voce del file di log per le operazioni di Secrets Manager

Gli eventi generati dalle chiamate alle operazioni di Secrets Manager hanno `"detail-type": ["AWS API Call via CloudTrail"]`.

> **Note**
>
> Prima di febbraio 2024, alcune operazioni di Secrets Manager segnalavano eventi che contenevano «arN» anziché «arn» per l'ARN segreto. Per ulteriori informazioni, consulta AWS re:Post.

Di seguito sono riportate le CloudTrail voci generate quando l'utente o un servizio richiama le operazioni di Secrets Manager tramite API, SDK o CLI.

BatchGetSecretValue

Generato dall'operazione. BatchGetSecretValue Per ulteriori informazioni sul recupero dei segreti, consulta Recuperare segreti.

CancelRotateSecret

Generato dall'CancelRotateSecretoperazione. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

CreateSecret

Generato dall'CreateSecretoperazione. Per ulteriori informazioni sulla creazione di un segreto, consulta Creazione e gestione di segreti.

DeleteResourcePolicy

Generato dall'DeleteResourcePolicyoperazione. Per informazioni sulle autorizzazioni, consulta Autenticazione e controllo degli accessi.

DeleteSecret

Generato dall'DeleteSecretoperazione. Per informazioni sull'eliminazione dei segreti, consulta the section called “Eliminare un segreto”.

DescribeSecret

Generato dall'DescribeSecretoperazione.

GetRandomPassword

Generato dall'GetRandomPasswordoperazione.

GetResourcePolicy

Generato dall'GetResourcePolicyoperazione. Per informazioni sulle autorizzazioni, consulta Autenticazione e controllo degli accessi.

GetSecretValue

Generato dalle BatchGetSecretValueoperazioni GetSecretValuee. Per ulteriori informazioni sul recupero dei segreti, consulta Recuperare segreti.

ListSecrets

Generato dall'ListSecretsoperazione. Per ulteriori informazioni sulla visualizzazione dei segreti, consulta the section called “Scopri i segreti”.

ListSecretVersionIds

Generato dall'ListSecretVersionIdsoperazione.

PutResourcePolicy

Generato dall'PutResourcePolicyoperazione. Per informazioni sulle autorizzazioni, consulta Autenticazione e controllo degli accessi.

PutSecretValue

Generato dall'PutSecretValueoperazione. Per ulteriori informazioni sull'aggiornamento di un segreto, consulta the section called “Modificare un segreto”.

RemoveRegionsFromReplication

Generato dall'RemoveRegionsFromReplicationoperazione. Per ulteriori informazioni sulla replica di un segreto, consulta the section called “Replica un segreto ad altre Regioni”.

ReplicateSecretToRegions

Generato dall'ReplicateSecretToRegionsoperazione. Per ulteriori informazioni sulla replica di un segreto, consulta the section called “Replica un segreto ad altre Regioni”.

RestoreSecret

Generato dall'RestoreSecretoperazione. Per informazioni sul ripristino di un segreto eliminato, consulta the section called “Ripristino di un segreto”.

RotateSecret

Generato dall'RotateSecretoperazione. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

StopReplicationToReplica

Generato dall'StopReplicationToReplicaoperazione. Per ulteriori informazioni sulla replica di un segreto, consulta the section called “Replica un segreto ad altre Regioni”.

TagResource

Generato dall'TagResourceoperazione. Per informazioni su come aggiungere un tag a un segreto, consulta the section called “Tag segreti ”.

UntagResource

Generato dall'UntagResourceoperazione. Per informazioni su come eliminare il tag di un segreto, consulta the section called “Tag segreti ”.

UpdateSecret

Generato dall'UpdateSecretoperazione. Per ulteriori informazioni sull'aggiornamento di un segreto, consulta the section called “Modificare un segreto”.

UpdateSecretVersionStage

Generato dall'UpdateSecretVersionStageoperazione. Per ulteriori informazioni sulle fasi di una versione, consulta the section called “Versione”.

ValidateResourcePolicy

Generato dall'ValidateResourcePolicyoperazione. Per informazioni sulle autorizzazioni, consulta Autenticazione e controllo degli accessi.

Registra le voci del file di log da eliminare

Oltre agli eventi per le operazioni di Secrets Manager, Secrets Manager genera i seguenti eventi relativi alla cancellazione. Questi eventi hanno `"detail-type": ["AWS Service Event via CloudTrail"]`.

CancelSecretVersionDelete

Generato dal servizio Secrets Manager. Se chiami `DeleteSecret` su un segreto che ha delle versioni e successivamente chiami `RestoreSecret`, Secrets Manager registra nel file di log

questo evento per ogni versione del segreto ripristinata. Per informazioni sul ripristino di un segreto eliminato, consulta the section called “Ripristino di un segreto”.

EndSecretVersionDelete

Generato dal servizio Secrets Manager quando viene eliminata una versione del segreto. Per ulteriori informazioni, consulta the section called “Eliminare un segreto”.

StartSecretVersionDelete

Generato dal servizio Secrets Manager quando Secrets Manager avvia l'eliminazione di una versione del segreto. Per informazioni sull'eliminazione dei segreti, consulta the section called “Eliminare un segreto”.

SecretVersionDeletion

Generato dal servizio Secrets Manager quando Secrets Manager elimina una versione del segreto obsoleta. Per ulteriori informazioni, consulta Versioni del segreto.

Voci di log per la replica

Oltre agli eventi per le operazioni di Secrets Manager, Secrets Manager genera i seguenti eventi relativi alla replica. Questi eventi hanno `"detail-type": ["AWS Service Event via CloudTrail"]`.

ReplicationFailed

Generato dal servizio Secrets Manager quando la replica fallisce. Per ulteriori informazioni sulla replica di un segreto, consulta the section called “Replica un segreto ad altre Regioni”.

ReplicationStarted

Generato dal servizio Secrets Manager quando Secrets Manager inizia a replicare un segreto. Per ulteriori informazioni sulla replica di un segreto, consulta the section called “Replica un segreto ad altre Regioni”.

ReplicationSucceeded

Generato dal servizio Secrets Manager quando un segreto viene replicato correttamente. Per ulteriori informazioni sulla replica di un segreto, consulta the section called “Replica un segreto ad altre Regioni”.

Registra le voci del file di log per la rotazione

Oltre agli eventi per le operazioni di Secrets Manager, Secrets Manager genera i seguenti eventi relativi alla rotazione. Questi eventi hanno `"detail-type": ["AWS Service Event via CloudTrail"]`.

RotationStarted

Generato dal servizio Secrets Manager quando Secrets Manager inizia a ruotare un segreto. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

RotationAbandoned

Generato dal servizio Secrets Manager quando Secrets Manager abbandona un tentativo di rotazione e rimuove l'etichetta `AWSPENDING` da una versione esistente di un segreto. Secrets Manager abbandona la rotazione quando crei una nuova versione di un segreto durante la rotazione. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

RotationFailed

Generato dal servizio Secrets Manager quando la rotazione fallisce. Per ulteriori informazioni sulla rotazione, consulta the section called “Risoluzione dei problemi della rotazione”.

RotationSucceeded

Generato dal servizio Secrets Manager quando un segreto viene ruotato correttamente. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

TestRotationStarted

Generato dal servizio Secrets Manager quando Secrets Manager inizia il test della rotazione per un segreto che non è pianificato per la rotazione immediata. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

TestRotationSucceeded

Generato dal servizio Secrets Manager quando Secrets Manager esegue correttamente il test della rotazione per un segreto che non è pianificato per la rotazione immediata. Per ulteriori informazioni sulla rotazione, consulta Rotazione dei segreti .

TestRotationFailed

Generato dal servizio Secrets Manager quando Secrets Manager esegue il test della rotazione per un segreto che non è pianificato per la rotazione immediata e la rotazione non riesce. Per ulteriori informazioni sulla rotazione, consulta the section called “Risoluzione dei problemi della rotazione”.

Abbina AWS Secrets Manager gli eventi con Amazon EventBridge

In Amazon EventBridge, puoi abbinare gli eventi di Secrets Manager dalle voci di CloudTrail registro. Puoi configurare EventBridge regole che cercano questi eventi e quindi inviano i nuovi eventi generati a un target affinché agisca. Per un elenco delle CloudTrail voci registrate da Secrets Manager, vedereCloudTrail voci. Per istruzioni sulla configurazione EventBridge, consulta Guida introduttiva EventBridge nella Guida per l'EventBridge utente.

Associa tutte le modifiche a un segreto specificato

L'esempio seguente mostra uno schema di EventBridge eventi che corrisponde alle voci di registro relative alle modifiche a un segreto.

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

Abbina gli eventi quando un valore segreto ruota

L'esempio seguente mostra uno schema di EventBridge eventi che corrisponde alle voci di CloudTrail registro per le modifiche ai valori segreti che si verificano in seguito agli aggiornamenti manuali o alla rotazione automatica. Poiché alcuni di questi eventi derivano dalle operazioni di Secrets Manager e altri sono generati dal servizio Secrets Manager, è necessario includere il `detail-type` per entrambi.

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

Monitora AWS Secrets Manager con Amazon CloudWatch

Puoi monitorare AWS Secrets Manager utilizzando Amazon CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione Web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Per Secrets Manager, puoi CloudWatch utilizzarlo per avvisarti quando la frequenza di richieste di API o il numero di segreti nel tuo account raggiunge una soglia specifica. Puoi anche utilizzarlo CloudWatch per monitorare i costi stimati di Secrets Manager. Per ulteriori informazioni, consulta Creazione di un allarme di fatturazione per il monitoraggio dei costi di AWS stimati.

Argomenti

- Parametri e dimensioni di Secrets Manager
- Creare gli allarmi per monitorare i parametri di Secrets Manager
- Amazon CloudWatch Synthetics canarini

Parametri e dimensioni di Secrets Manager

Lo spazio dei nomi `AWS/SecretsManager` include le metriche descritte di seguito.

Metrica	Descrizione
`ResourceCount`	Il numero di segreti nel tuo account, inclusi i segreti contrassegnati per l'eliminazione. La metrica viene pubblicata ogni ora. Unità: numero

Dimensioni per i parametri di Secrets Manager.

Dimensione	Descrizione
`Service`	Il nome del servizio AWS contenente la risorsa. Per Secrets Manager, il valore di questa dimensione è `Secrets Manager`.
`Type`	Il tipo di entità che viene segnalato. Per Secrets Manager, il valore di questa dimensione è `Resource`.
`Resource`	Il tipo di risorsa in esecuzione. Per Secrets Manager, il valore di questa dimensione è `SecretCount` .
`Class`	Nessuna.

Le richieste API di Secrets Manager che puoi monitorare utilizzando CloudWatch le metriche includono `GetSecretValue`,`DescribeSecret`,`ListSecrets`, e altre. Per trovare le metriche, nella CloudWatch console, scegli Tutte le metriche, quindi nella casella di ricerca, inserisci il termine di ricerca, ad esempio. **`secrets`**

Creare gli allarmi per monitorare i parametri di Secrets Manager

Puoi creare un CloudWatch allarme che invia un messaggio Amazon SNS quando il valore della metrica cambia e causa il cambio di stato dell'allarme. Un allarme controlla un parametro in un periodo di tempo specificato ed esegue le operazioni in base al valore del parametro relativo a una determinata soglia in una serie di periodi di tempo. Gli allarmi richiamano azioni solo per cambiamenti di stato sostenuti. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi.

Per ulteriori informazioni, consulta Usare gli CloudWatch allarmi Amazon e Creare un CloudWatch allarme basato sul rilevamento delle anomalie.

Amazon CloudWatch Synthetics canarini

Amazon CloudWatch Synthetics Canaries sono script configurabili che vengono eseguiti secondo una pianificazione per monitorare endpoint e API. I canary seguono gli stessi percorsi ed eseguono le stesse azioni di un cliente, il che ti consente di verificare continuamente la tua esperienza cliente anche quando non hai alcun traffico clienti sulle tue applicazioni.

Per un esempio di come integrare Secrets Manager, consulta Integrare il tuo Canary con altri servizi AWS.

Monitoraggio dei segreti AWS Secrets Manager pianificati per l'eliminazione utilizzando Amazon CloudWatch

Puoi utilizzare una combinazione di AWS CloudTrail, Amazon CloudWatch Logs e Amazon Simple Notification Service (Amazon SNS) per creare un allarme che invia una notifica nel caso in cui si verifichino dei tentativi di accedere a un segreto in attesa di essere eliminato. Se ricevi una notifica mediante un allarme, puoi annullare l'eliminazione del segreto per avere più tempo per stabilire se vuoi effettivamente eliminarlo. La tua indagine potrebbe determinare il ripristino del segreto, se questo si rivela ancora effettivamente necessario. In alternativa, potrebbe essere necessario aggiornare l'utente con i dettagli del nuovo segreto da utilizzare.

Le procedure seguenti spiegano come ricevere una notifica quando una richiesta per l'operazione `GetSecretValue` che genera un messaggio di errore specifico viene scritta nei tuoi file di log CloudTrail. Altre operazioni API possono essere eseguite sul segreto senza attivare l'allarme. Questo allarme CloudWatch rileva l'utilizzo che potrebbe indicare una persona o un'applicazione che utilizza credenziali obsolete.

Prima di avviare queste procedure, devi avere già attivato CloudTrail nell' Regione AWS account e nella regione in cui desideri monitorare AWS Secrets Managerle richieste API . Per istruzioni, consulta Creazione di un Trail per la prima volta nella AWS CloudTrailGuida per l'utente.

Fase 1: Configurazione della consegna di file di log CloudTrail ai log CloudWatch

Devi configurare la consegna dei tuoi CloudTrail file di log a CloudWatch Logs. Tale operazione è necessaria per permettere a CloudWatch Logs di monitorare tali file per richieste API Secrets Manager per il recupero di un segreto che è in attesa di eliminazione.

Come configurare la consegna di file di log CloudTrail a CloudWatch Logs

1. Aprire la console CloudTrail all'indirizzo https://console.aws.amazon.com/cloudtrail/.
2. Nella barra di navigazione superiore, scegli la regione AWS in cui monitorare i segreti.
3. Nel riquadro di navigazione a sinistra, scegli Trails (Trail), quindi seleziona il nome del trail da configurare per CloudWatch.

4. Nella pagina Trails Configuration (Configurazione trail) scorri verso il basso alla sezione CloudWatch Logs e scegli l'icona di modifica (✏️).
5. Per un New or existing log group (Gruppo nuovo o esistente di log), digita un nome per il gruppo di log, ad esempio **`CloudTrail/MyCloudWatchLogGroup`**.
6. Per IAM role, è possibile utilizzare il ruolo predefinito denominato CloudTrail_CloudWatchLogs_Role. Questo ruolo dispone di una policy del ruolo predefinita con le autorizzazioni richieste per distribuire gli eventi CloudTrail al gruppo di log.
7. Scegli Continue (Continua) per salvare la configurazione.
8. Nella pagina AWS CloudTrail consegnerà eventi CloudTrail associati con le attività API nel tuo account nel tuo gruppo di log CloudWatch Logs, seleziona Allow (Consenti).

Fase 2: Creazione di allarme CloudWatch

Per ricevere una notifica quando un'operazione API Secrets Manager `GetSecretValue` richiede di accedere a un segreto che è in attesa di essere eliminato, è necessario creare un allarme CloudWatch e configurare la notifica.

Creazione di un allarme CloudWatch

1. Accedi alla e apri la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nella barra di navigazione superiore, scegli la regione AWS nella quale desideri monitorare i segreti.
3. Nel pannello di navigazione a sinistra, scegli Logs (Log).
4. Nell'elenco Log Groups (Gruppi di log), seleziona la casella di controllo accanto al gruppo di log creato nella procedura precedente, ad esempio CloudTrail/MyCloudWatchLogGroup. Quindi scegli Create Metric Filter (Crea filtro parametro).
5. Per Filter Pattern (Modello filtri), digita o incolla quanto segue:

   ```
   { $.eventName = "GetSecretValue" && $.errorMessage = "*secret because it was marked
    for deletion*" }
   ```

   Scegli Assign Metric (Assegna parametro).

6. Nella pagina Create Metric Filter and Assign a Metric (Crea filtro parametri e assegna parametro), procedi nel seguente modo:

   a. Per Metric Namespace (Spazio dei nomi parametro), digita **`CloudTrailLogMetrics`**.
   b. Per Metric Name (Nome parametro) digita **`AttemptsToAccessDeletedSecrets`**.
   c. Scegli Show advanced metric settings (Mostra impostazioni avanzate parametro) quindi, se necessario per Metric Value (Valore parametro) digita **`1`**.
   d. Scegli Create Filter (Crea filtro).
7. Nella casella di filtro, scegli Create Alarm (Crea allarme).
8. Nella finestra Create Alarm (Crea allarme), procedi nel seguente modo:

   a. In Name (Nome) digitare **`AttemptsToAccessDeletedSecretsAlarm`**.
   b. In Whenever: (Ogni volta che:), per is: (è:), seleziona **`>=`**, quindi digita **`1`**.
   c. Nel campo Send notification to: (Invia notifica a), procedi in uno dei seguenti modi:

      - Per creare e utilizzare un nuovo argomento Amazon SNS, scegli New list (Nuovo elenco), quindi digita un nuovo nome argomento. Per Email list: (Elenco e-mail), digita almeno un indirizzo e-mail. È possibile digitare più di un indirizzo e-mail utilizzando la virgola come separatore.
      - Per usare un argomento Amazon SNS esistente, scegli il nome dell'argomento da utilizzare. Se non esiste un elenco, seleziona Select list (Elenco di selezione).
   d. Scegli Create Alarm (Crea allarme).

Fase 3: Esecuzione del test dell'allarme CloudWatch

Per fare un test sull'allarme, crea un segreto e programmane l'eliminazione. Quindi, prova a recuperare il valore del segreto. Poco dopo riceverai un'e-mail all'indirizzo configurato nell'allarme. Questo avvisa di utilizzare un segreto pianificato per l'eliminazione.

Convalida della conformità per AWS Secrets Manager

La responsabilità per la compliance quando utilizzi i servizi Secrets Manager è determinata dalla riservatezza dei dati, dagli obiettivi di compliance dell'azienda e dalle normative vigenti. AWS fornisce le risorse seguenti per semplificare la compliance:

- Guide di riferimento rapido su sicurezza e conformità - Queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sulla progettazione per la sicurezza HIPAA e la conformità: questo whitepaper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni conformi ai requisiti HIPAA.
- Risorse per la conformità AWS: una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- AWS Config valuta il livello di conformità delle configurazioni delle risorse con le pratiche interne, le linee guida e i regolamenti di settore. Per ulteriori informazioni, consulta the section called “Controllare i segreti per la conformità”.
- AWS Security Hub fornisce una visione completa dello stato di sicurezza all'interno di AWS che consente di verificare la conformità con gli standard e le best practice di sicurezza del settore. Per ulteriori informazioni sull'utilizzo di Security Hub volto a valutare le risorse Lambda, consulta i controlli di AWS Secrets Manager nella Guida per l'utente di AWS Security Hub.
- IAM Access Analyzer analizza le policy, incluse le istruzioni di condizione in una policy, che consentono a un'entità esterna di accedere a un segreto. Per ulteriori informazioni, consulta Anteprima dell'accesso con Access Analyzer.
- AWS Systems Manager fornisce runbook predefiniti per Secrets Manager. Per ulteriori informazioni, consulta Systems Manager Automation runbook reference for Secrets Manager (Riferimento del runbook di automazione di Systems Manager).

AWS Secrets Manager è stato sottoposto a controllo per i seguenti standard e può essere parte della soluzione quando occorre ottenere una certificazione di conformità.

AWS ha ampliato il proprio programma di conformità con Health Insurance Portabili ty and Accountability Act (HIPAA) per includere AWS Secrets Manager come servizio idoneo ai fini HIPAA. Se disponi di un contratto di società in affari o BAA (Business Associate Agreement) con AWS, puoi utilizzare Secrets Manager per creare applicazi

	oni conformi a HIPAA. AWS offre un whitepaper incentrato sulle disposizioni HIPAA ai clienti interessati a scoprire come utilizzare AWS per l'elaborazione e lo storage dei dati sanitari. Per ulteriori informazioni, consulta Compliance HIPAA.
	AWS Secrets Manager ha un'attestazione di conformità per lo standard Payment Card Industry (PCI) Data Security (DSS) versione 3.2 come fornitore di servizi di livello 1. I clienti che utilizzano i prodotti e i servizi AWS per archiviare, elaborare o trasmette re dati dei titolari di carte di credito possono utilizzare AWS Secrets Manager nella gestione della propria certificazione di conformità PCI DSS. Per ulteriori informazi oni sullo standard PCI DSS, incluse le istruzioni su come richiedere una copia del Pacchetto conformità PCI di AWS, consulta PCI DSS livello 1.
	AWS Secrets Manager ha ottenuto con successo una certificazione di conformità con gli standard ISO/IEC 27001, ISO/IEC 27017, ISO/IEC 27018 e ISO 9001. Per ulteriori informazioni, consulta ISO 27001, ISO 27017, ISO 27018, ISO 9001.
	I report Service Organization Control (SOC) sono report di valutazione di terze parti indipendenti che documentano in che modo Secrets Manager consegue i controlli e gli obiettivi chiave di compliance. Lo scopo di questi report è consentire a clienti e revisori di raccogliere informazioni sui controlli previsti da AWS per supportare operatività e compliance. Per ulteriori informazioni, consulta la pagina Conformità SOC.
	FedRAMP sta per Federal Risk and Authorization Management Program; si tratta di un programma federale per la gestione di rischio e autorizzazioni applicato a livello di pubblica amministrazione che fornisce un approccio standardizzato a valutazioni di sicurezza, assegnazione di autorizzazioni e monitoraggio continuo nell'ambito di servizi e prodotti cloud. Il programma FedRAMP fornisce inoltre autorizzazioni provvisorie per servizi e regioni per East/West e GovCloud per l'utilizzo di dati governativi o regolamen tati. Per ulteriori informazioni, consulta Conformità FedRAMP.
	La Security Requirements Guide (SRG) al cloud computing del Dipartimento della Difesa (DoD) fornisce un processo di valutazione e autorizzazione standardizzato per i provider di servizi cloud (CSP) per ottenere un'autorizzazione provvisoria DoD, in modo che possano servire i clienti DoD. Per ulteriori informazioni, consulta DoD SRG Resources

	L'Information Security Registered Assessors Program (IRAP) consente ai clienti governativi australiani di convalidare l'esistenza di controlli appropriati e di determina re il modello di responsabilità appropriato per soddisfare i requisiti del Manuale per la sicurezza delle informazioni del governo australiano (ISM) prodotto dal Centro australia no per la sicurezza informatica (ACSC). Per ulteriori informazioni, consulta IRAP Resources
	Amazon Web Services (AWS) ha ottenuto l'attestazione di Audit Report (OSPAR) del fornitore di servizi in outsourcing.AWSl'allineamento con l'Associazione delle banche a Singapore (ABS) Linee guida sugli obiettivi e le procedure di controllo per fornitori di servizi in outsourcing (ABS Linee guida) dimostra ai clienti AWS impegno a soddisfar e le elevate aspettative per i provider di servizi cloud impostate dal settore dei servizi finanziari a Singapore. Per ulteriori informazioni, consulta Risorse OSPAR.

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Download di report in AWS Artifact.

Controllare i segreti AWS Secrets Manager per la conformità utilizzando AWS Config

Puoi utilizzare AWS Config per valutare i segreti e il livello di conformità con pratiche interne, linee guida e regolamenti industriali. Definisci i requisiti interni di sicurezza e conformità per i segreti utilizzando le regole AWS Config. In seguito, AWS Config è in grado di identificare segreti che non sono conformi alle tue regole. È inoltre possibile tenere traccia delle modifiche ai metadati segreti, alla configurazione di rotazione, alla chiave KMS utilizzata per la crittografia segreta, alla funzione di rotazione Lambda e ai tag associati a un segreto.

Puoi ricevere notifiche da Amazon SNS sulle configurazioni segrete. Ad esempio, puoi ricevere notifiche di Amazon SNS su un elenco di segreti non configurati per la rotazione che consente di guidare le best practice per la sicurezza per la rotazione dei segreti.

Se hai segreti in più Account AWS e Regioni AWS nell'organizzazione, è possibile aggregare i dati di conformità e configurazione.

Per aggiungere una nuova regola per i segreti

- Segui le istruzioni in Utilizzo delle regole gestite di AWS Config, quindi scegli una delle regole seguenti:
  - `secretsmanager-rotation-enabled-check`— Verifica se la rotazione è configurata per i segreti memorizzati in Secrets Manager.
  - `secretsmanager-scheduled-rotation-success-check`: verifica se l'ultima rotazione riuscita rientra nella frequenza di rotazione configurata. La frequenza minima per la verifica è giornaliera.
  - `secretsmanager-secret-periodic-rotation`— Controlla se i segreti sono stati ruotati entro il numero specificato di giorni.
  - `secretsmanager-secret-unused`— Controlla se sono stati eseguiti accessi ai segreti entro il numero specificato di giorni.
  - `secretsmanager-using-cmk`— Controlla se i segreti sono crittografati usando Chiave gestita da AWS `aws/secretsmanager` o una chiave gestita dal cliente creata in AWS KMS.

Una volta salvata la regola, AWS Config valuta i tuoi segreti ogni volta che i metadati di un segreto cambiano. È possibile configurare AWS Config per la notifica delle modifiche. Per ulteriori informazioni, consulta Notifiche che AWS Config invia a un argomento Amazon SNS.

Aggrega segreti dal tuo Account AWS e Regioni AWS

Puoi configurare l'aggregatore di dati AWS Config in più account e più regioni per esaminare le configurazioni dei segreti in tutti gli account e le regioni dell'organizzazione, quindi esaminare le configurazioni segrete e confrontarle con le best practice per la gestione dei segreti.

Prima di creare un aggregatore, devi abilitare le regole gestite specifiche AWS Config e AWS Config per i segreti in tutti gli account e in tutte le regioni. Per ulteriori informazioni, consulta Utilizzare CloudFormation StackSets per eseguire il provisioning di risorse su più Account AWS e Regioni.

Per ulteriori informazioni sull'aggregatore AWS Config, consulta Aggregazione di dati da più account e regioni e Impostazione di un aggregatore utilizzando la Console nella Guida per gli sviluppatori di AWS Config.

Sicurezza in AWS Secrets Manager

Per AWS, la sicurezza ha la massima priorità. In quanto cliente AWS, trai vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

Tu e AWS condividete la responsabilità per la sicurezza. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud – AWS è responsabile della protezione dell'infrastruttura che esegue i servizi AWS in AWS Cloud. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformità AWS. Per ulteriori informazioni sui programmi di conformità che si applicano a AWS Secrets Manager, consulta Servizi coperti dal programma di conformitàAWS.
- Sicurezza nel cloud: il servizio AWS determina la tua responsabilità. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda, le leggi e le normative applicabili.

Per altre risorse, vedere Pilastro della Sicurezza - Well-Architected Framework AWS.

Argomenti

- Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager
- Protezione dei dati in AWS Secrets Manager
- Crittografia e decrittografia segrete in AWS Secrets Manager
- Sicurezza dell'infrastruttura in AWS Secrets Manager
- Resilienza in AWS Secrets Manager
- TLS post-quantistico

Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager

Quando utilizzi AWS Command Line Interface (AWS CLI) per richiamare operazioni AWSimmetti tali comandi in una shell comando. Ad esempio, puoi usare il prompt dei comandi di Windows o Windows PowerShell, oppure la shell Bash o Z, tra gli altri. Molte di queste shell di comando

includono funzionalità progettate per aumentare la produttività. Ma queste funzionalità possono essere utilizzate per compromettere i tuoi segreti. Ad esempio, nella maggior parte delle shell puoi utilizzare il tasto freccia verso l'alto per visualizzare l'ultimo comando inserito. La funzionalità di cronologia dei comandi può essere sfruttata da chiunque acceda alla tua sessione non protetta. Inoltre, altre utility che funzionano in background potrebbero accedere ai parametri di comando, con l'obiettivo di aiutarti a eseguire le tue attività in modo più efficace. Per ridurre tali rischi, accertati di procedere nel modo seguente:

- Blocca sempre il tuo computer quando ti allontani dalla console.
- Disinstallare o disattivare le utility della console non necessarie o non più utilizzate.
- Assicurarsi che la shell o il programma di accesso remoto, se viene utilizzato l'uno o l'altro, non registrino i comandi digitati.
- Utilizzare le tecniche per inviare parametri non acquisiti dalla cronologia dei comandi della shell. L'esempio seguente mostra il modo in cui puoi digitare il testo di un segreto in un file di testo e inviare il file al comando AWS Secrets Manager ed eliminare immediatamente dopo il file. Ciò significa che la tipica cronologia dei comandi della shell non cattura il testo del segreto.

  L'esempio seguente mostra i comandi tipici di Linux, ma la tua shell potrebbe prevedere comandi lievemente diversi:

```
$ 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.
```

Dopo l'esecuzione di questi comandi, dovresti essere in grado di usare le frecce verso l'alto e il basso per scorrere la cronologia dei comandi e vedere che il testo del segreto non è presente in alcuna riga.

> **Important**
>
> Per impostazione predefinita, non puoi adottare una tecnica equivalente in Windows a meno che prima tu non riduca a 1 le dimensioni del buffer della cronologia dei comandi.

Per configurare il prompt dei comandi di Windows affinché presenti solo 1 buffer della cronologia dei comandi di 1 comando

1. Aprire un prompt dei comandi amministratore (Run as administrator (Esegui come amministratore)).
2. Scegliere l'icona in alto a sinistra, quindi selezionare Properties (Proprietà).
3. Nella scheda Options (Opzioni) imposta Buffer Size (Dimensioni buffer) e Number of Buffers (Numero di buffer) entrambi su **1**, quindi scegli OK.
4. Ogni volta che devi digitare un comando che non vuoi includere nella cronologia, fallo seguire immediatamente da un altro comando, ad esempio:

   ```
   echo.
   ```

   In questo modo sei sicuro che il comando sensibile non venga incluso.

Per la shell del prompt di comandi di Windows, puoi scaricare lo strumento SysInternals SDelete e utilizzare comandi simili ai seguenti:

```
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.
```

Protezione dei dati in AWS Secrets Manager

Il modello di responsabilità condivisa di AWS si applica alla protezione dei dati in AWS Secrets Manager. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che esegue tutto l'Cloud AWS. L'utente è responsabile di mantenere il controllo sui contenuti ospitati su questa infrastruttura. Questi contenuti comprendono la configurazione della protezione e le attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza negli AWS.

Per garantire la protezione dei dati, ti suggeriamo di proteggere le credenziali Account AWS e di configurare singoli account utente con AWS Identity and Access Management (IAM). In questo modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere il suo lavoro. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con le risorse AWS. Secrets Manager supporta TLS 1.2 e 1.3 in tutte le Regioni. Secrets Manager supporta anche un'opzione ibrida di scambio di chiavi post-quantistiche per il protocollo di crittografia di rete TLS (PQTLS).
- Firma le tue richieste programmatiche utilizzando sia un ID chiave di accesso che una chiave di accesso segreta associate a un'entità principale IAM. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.
- Configura la registrazione delle API e delle attività degli utenti con AWS CloudTrail. Per informazioni, consultare the section called “Accedi conAWS CloudTrail ”.
- Se si richiedono moduli crittografici convalidati FIPS 140-2 quando si accede ad AWS tramite una CLI o un'API, utilizzare un endpoint FIPS. Per informazioni, consultare the section called “Endpoint di Secrets Manager”.
- Se usi la AWS CLI per accedere a Secrets Manager, fai riferimento a the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

Crittografia dei dati inattivi

Secrets Manager utilizza la crittografia tramite AWS Key Management Service (AWS KMS) per proteggere la riservatezza dei dati inattivi. AWS KMS fornisce un servizio di crittografia e storage

delle chiavi utilizzato da moltiAWS servizi. Ogni segreto in Gestione dei segreti è crittografato con una chiave di dati univoca. Ogni chiave di dati è protetta da una chiave KMS. Puoi scegliere di utilizzare la crittografia predefinita con la Chiave gestita da AWS di Gestione dei segreti per l'account. In alternativa, puoi creare la tua chiave gestita dal cliente in AWS KMS. L'utilizzo di una chiave gestita dal cliente offre controlli di autorizzazione più granulari sulle principali attività del KMS. Per ulteriori informazioni, consulta the section called “Crittografia e decrittografia del segreto”.

Crittografia dei dati in transito

Secrets Manager fornisce endpoint sicuri e privati per la crittografia dei dati in transito. Gli endpoint sicuri e privati consentono a AWS di proteggere l'integrità delle richieste API a Secrets Manager. AWS richiede che le chiamate API siano firmate dal chiamante utilizzando i certificati X.509 e/o la relativa chiave di accesso segreta. Questo requisito è indicato nel Processo di firma Signature Version 4 (Sigv4).

Se utilizzi il AWS Command Line Interface (AWS CLI) o una delle AWS SDK per effettuare chiamate a AWS, configuri la chiave di accesso da utilizzare. Quindi questi strumenti utilizzano automaticamente la chiave di accesso per firmare le richieste per te. Per informazioni, consultare the section called “Riduzione dei rischi dell'utilizzo di AWS CLI per archiviare i segreti AWS Secrets Manager”.

Riservatezza del traffico Internet

AWS offre opzioni per la gestione della privacy durante l'instradamento del traffico attraverso percorsi di rete noti e privati.

Traffico tra servizio e applicazioni e client locali

Sono disponibili due opzioni di connettività tra la rete privata e AWS Secrets Manager:

- Una connessione Site-to-Site VPN AWS Per ulteriori informazioni, consulta Che cos'è AWS Site-to-Site VPN?
- Una connessione AWS Direct Connect. Per ulteriori informazioni, consulta Che cos'è AWS Direct Connect?

Traffico tra risorse AWS nella stessa Regione

Se intendi proteggere il traffico tra Secrets Manager e i client API in AWS, configura un AWS PrivateLink per accedere privatamente agli endpoint Secret Manager dell'API.

Gestione delle chiavi di crittografia

Quando Secrets Manager deve crittografare una nuova versione dei dati segreti protetti, Secrets Manager invia una richiesta a AWS KMS per generare una nuova chiave di dati dalla chiave KMS. Secrets Manager utilizza questa chiave di dati per la crittografia envelope. Secrets Manager archivia la chiave di dati crittografati con il segreto crittografato. Quando il segreto deve essere decrittografato, Secrets Manager chiede AWS KMS per decrittografare la chiave di dati. Quindi Secrets Manager usa la chiave di dati decrittografata per decrittografare il segreto crittografato. La chiave di dati non viene mai memorizzata in forma non crittografata da Secrets Manager e la chiave viene rimossa dalla memoria il prima possibile. Per ulteriori informazioni, consulta the section called “Crittografia e decrittografia del segreto”.

Crittografia e decrittografia segrete in AWS Secrets Manager

Secrets Manager utilizza la crittografia a busta con AWS KMS chiavi e chiavi dati per proteggere ogni valore segreto. Ogni volta che il valore segreto di un segreto cambia, Secrets Manager richiede una nuova chiave dati AWS KMS per proteggerlo. La chiave di dati è crittografata sotto una chiave KMS e viene archiviata nei metadati della versione segreta. Per decrittografare il segreto, Secrets Manager decrittografa innanzitutto la chiave dati crittografata utilizzando la chiave KMS. AWS KMS

Secrets Manager non utilizza la chiave KMS per crittografare il valore del segreto direttamente. Utilizza invece la chiave KMS per generare e crittografare una simmetrica AES (Advanced Encryption Standard) a 256 bit chiave di dati e utilizza la chiave di dati per crittografare il valore del segreto. Secrets Manager utilizza la chiave dati in testo semplice per crittografare il valore segreto all'esterno di AWS KMS, quindi lo rimuove dalla memoria. Archivia la copia crittografata della chiave di dati nei metadati del segreto.

Quando crei un segreto, puoi scegliere qualsiasi chiave di crittografia simmetrica gestita dal cliente nella regione Account AWS and oppure puoi utilizzare Chiave gestita da AWS for Secrets Manager ()`aws/secretsmanager`. Se scegli il Chiave gestita da AWS `aws/secretsmanager` e non esiste ancora, Secrets Manager lo crea e lo associa al segreto. È possibile utilizzare la stessa chiave KMS o diverse chiavi KMS per ogni segreto nell'account. Potresti voler utilizzare chiavi KMS diverse per impostare autorizzazioni personalizzate sulle chiavi per un gruppo di segreti o per controllare determinate operazioni per tali chiavi. Secrets Manager supporta solo chiavi KMS di crittografia simmetrica. Se utilizzi una chiave KMS in un archivio di chiavi esterno, le operazioni crittografiche sulla chiave KMS potrebbero richiedere più tempo ed essere meno affidabili e durevoli perché la richiesta deve essere trasferita all'esterno di AWS.

Per informazioni sulla modifica della chiave di crittografia di un segreto, consulta the section called “Modifica la chiave di crittografia per un segreto”.

Quando si modifica la chiave di crittografia, Secrets Manager cripta nuovamente `AWSCURRENT` e `AWSPENDING` le `AWSPREVIOUS` versioni con la nuova chiave. Per evitare di nasconderti il segreto, Secrets Manager mantiene tutte le versioni esistenti crittografate con la chiave precedente. Ciò significa che è possibile decrittografare `AWSCURRENT` `AWSPENDING` le `AWSPREVIOUS` versioni con la chiave precedente o quella nuova.

Per fare in modo che `AWSCURRENT` possa essere decrittografato solo con la nuova chiave di crittografia, crea una nuova versione del segreto con la nuova chiave. Quindi, per poter decifrare la versione `AWSCURRENT` segreta, devi avere l'autorizzazione per la nuova chiave.

Per trovare la chiave KMS associata a un segreto, visualizza il segreto nella console o chiama o. ListSecretsDescribeSecret Quando il segreto è associato a Chiave gestita da AWS for Secrets Manager (`aws/secretsmanager`), queste operazioni non restituiscono un identificatore di chiave KMS.

Argomenti

- Che viene crittografato?
- Processi di crittografia e decrittografia
- Autorizzazioni per la chiave KMS
- Come Secrets Manager utilizza la chiave KMS
- Policy chiave della Chiave gestita da AWS (aws/secretsmanager)
- Contesto di crittografia di Secrets Manager
- Monitora l'interazione di Secrets Manager con AWS KMS

Che viene crittografato?

Secrets Manager crittografa il valore segreto, ma non crittografa quanto segue:

- Nome e descrizione del segreto
- Impostazioni di rotazione
- ARN della chiave KMS associata al segreto
- Eventuali tag allegati AWS

Processi di crittografia e decrittografia

Per crittografare il valore del segreto in un segreto, Secrets Manager utilizza il seguente processo.

1. Secrets Manager richiama l' AWS KMS GenerateDataKeyoperazione con l'ID della chiave KMS per il segreto e una richiesta per una chiave simmetrica AES a 256 bit. AWS KMS restituisce una chiave dati in testo semplice e una copia di tale chiave dati crittografata con la chiave KMS.
2. Secrets Manager utilizza la chiave dati in chiaro e l'algoritmo Advanced Encryption Standard (AES) per crittografare il valore segreto all'esterno di. AWS KMS Dopo averla utilizzata, rimuove la chiave in testo normale dalla memoria il prima possibile.
3. Secrets Manager archivia la chiave di dati crittografata nei metadati del segreto in modo che sia disponibile per decrittografare il valore del segreto. Tuttavia, nessuna delle API Secrets Manager restituisce il segreto crittografato o la chiave di dati crittografata.

Per decrittografare un valore del segreto crittografato:

1. Secrets Manager richiama l'operazione AWS KMS Decrypt e trasmette la chiave dati crittografata.
2. AWS KMS utilizza la chiave KMS come segreto per decrittografare la chiave dati. Restituisce la chiave di dati in testo normale.
3. Secrets Manager utilizza la chiave di dati in testo normale per decrittografare il valore del segreto. Quindi rimuove la chiave di dati dalla memoria il prima possibile.

Autorizzazioni per la chiave KMS

Quando Secrets Manager utilizza una chiave KMS in operazioni di crittografia, agisce per conto dell'utente che sta effettuando l'accesso al valore del segreto o che lo sta aggiornando. Puoi concedere le autorizzazioni in una policy IAM o in una policy delle chiavi. Le seguenti operazioni di Secrets Manager richiedono AWS KMS autorizzazioni.

- CreateSecret
- GetSecretValue
- PutSecretValue
- UpdateSecret
- ReplicateSecretToRegions

Per consentire l'utilizzo della chiave KMS solo per le richieste che hanno origine in Secrets Manager, nella politica delle autorizzazioni, puoi utilizzare la chiave kms: ViaService condition con il valore. `secretsmanager.<Region>.amazonaws.com`

Puoi inoltre utilizzare le chiavi o i valori nel contesto di crittografia come condizione per utilizzare la chiave KMS per le operazioni di crittografia. Ad esempio, è possibile utilizzare un operatore di condizione stringa in un documento di policy IAM o della chiave, oppure utilizzare un vincolo di concessione in un vincolo. La propagazione della concessione di chiavi KMS può richiedere fino a cinque minuti. Per ulteriori informazioni, vedere. CreateGrant

Come Secrets Manager utilizza la chiave KMS

Secrets Manager richiama le seguenti AWS KMS operazioni con la chiave KMS.

GenerateDataKey

Secrets Manager chiama l' AWS KMS GenerateDataKeyoperazione in risposta alle seguenti operazioni di Secrets Manager.

- CreateSecret— Se il nuovo segreto include un valore segreto, Secrets Manager richiede una nuova chiave dati per crittografarlo.
- PutSecretValue— Secrets Manager richiede una nuova chiave dati per crittografare il valore segreto specificato.
- ReplicateSecretToRegions— Per crittografare il segreto replicato, Secrets Manager richiede una chiave dati per la chiave KMS nella regione di replica.
- UpdateSecret— Se si modifica il valore segreto o la chiave KMS, Secrets Manager richiede una nuova chiave dati per crittografare il nuovo valore segreto.

L'RotateSecretoperazione non chiama`GenerateDataKey`, perché non modifica il valore segreto. Tuttavia, se `RotateSecret` invoca una funzione di rotazione Lambda che modifica il valore del segreto, la chiamata all'operazione `PutSecretValue` attiva una richiesta `GenerateDataKey`.

Decrypt

Secrets Manager chiama la Decrypt operazione in risposta alle seguenti operazioni di Secrets Manager.

- GetSecretValuee BatchGetSecretValue— Secrets Manager decripta il valore segreto prima di restituirlo al chiamante. Per decrittografare un valore segreto crittografato, Secrets Manager chiama l'operazione AWS KMS Decrypt per decrittografare la chiave dei dati crittografati nel segreto. Quindi utilizza la chiave di dati in testo normale per decrittografare il valore del segreto

crittografato. Per i comandi batch, Secrets Manager può riutilizzare la chiave decrittografata, in modo che non tutte le chiamate generino una richiesta `Decrypt`.

- PutSecretValuee UpdateSecret— La maggior parte delle `UpdateSecret` richieste non `PutSecretValue` attiva un'operazione. `Decrypt` Tuttavia, quando una richiesta `PutSecretValue` o `UpdateSecret` cerca di modificare il valore del segreto in una versione esistente di un segreto, Secrets Manager decrittografa il valore del segreto esistente e lo confronta con il valore del segreto nella richiesta per confermare che siano identici. Questa operazione garantisce che le operazioni Secrets Manager siano idempotenti. Per decrittografare un valore segreto crittografato, Secrets Manager chiama l'operazione AWS KMS Decrypt per decrittografare la chiave dei dati crittografati nel segreto. Quindi utilizza la chiave di dati in testo normale per decrittografare il valore del segreto crittografato.
- ReplicateSecretToRegions— Secrets Manager decrittografa innanzitutto il valore segreto nella regione primaria prima di ricrittografare il valore segreto con la chiave KMS nella regione di replica.

Crittografa

Secrets Manager chiama l'operazione Encrypt in risposta alle seguenti operazioni di Secrets Manager:

- UpdateSecret— Se si modifica la chiave KMS, Secrets Manager cripta nuovamente la chiave dati che protegge la `AWSCURRENT` chiave e le versioni `AWSPENDING` segrete con la nuova chiave. `AWSPREVIOUS`
- ReplicateSecretToRegions— Secrets Manager crittografa nuovamente la chiave dati durante la replica utilizzando la chiave KMS nella regione di replica.

DescribeKey

Secrets Manager richiama l'DescribeKeyoperazione per determinare se elencare la chiave KMS quando si crea o si modifica un segreto nella console Secrets Manager.

Convalida dell'accesso alla chiave KMS

Quando stabilisci o modifichi la chiave KMS associata al segreto, Secrets Manager chiama le operazioni `GenerateDataKey` e `Decrypt` con la chiave KMS specificata. Queste chiamate confermano che il chiamante ha l'autorizzazione di utilizzare la chiave KMS per queste operazioni. Secrets Manager scarta i risultati di queste operazioni e non li utilizza in alcuna operazione di crittografia.

È possibile identificare queste chiamate di convalida perché il valore della `SecretVersionId` chiave contesto di crittografia in queste richieste è `RequestToValidateKeyAccess`.

> **Note**
>
> In passato, le chiamate di convalida di Secrets Manager non includevano un contesto di crittografia. È possibile trovare chiamate senza contesto di crittografia nei AWS CloudTrail registri precedenti.

Policy chiave della Chiave gestita da AWS (`aws/secretsmanager`)

La politica chiave per Chiave gestita da AWS for Secrets Manager (`aws/secretsmanager`) consente agli utenti di utilizzare la chiave KMS per operazioni specifiche solo quando Secrets Manager effettua la richiesta per conto dell'utente. La policy delle chiavi non consente ad alcun utente di utilizzare la chiave KMS direttamente.

Questa policy delle chiavi, come le policy di tutte le Chiavi gestite da AWS, viene stabilita dal servizio. Non è possibile modificarla, ma è possibile visualizzarla in qualsiasi momento. Per informazioni dettagliate, consulta Visualizzazione di una policy di chiave.

Le istruzioni di policy nella policy delle chiavi hanno l'effetto seguente:

- Consenti agli utenti nell'account di utilizzare la chiave KMS per le operazioni di crittografia solo quando la richiesta proviene da Secrets Manager per conto loro. La chiave di condizione `kms:ViaService` applica questa limitazione.
- Consente all' AWS account di creare policy IAM che consentono agli utenti di visualizzare le proprietà delle chiavi KMS e revocare le concessioni.
- Sebbene Secrets Manager non utilizzi le sovvenzioni per accedere alla chiave KMS, la policy consente inoltre a Secrets Manager di creare sovvenzioni per la chiave KMS per conto dell'utente e consente all'account di revoca di qualsiasi concessione che consente a Secrets Manager di utilizzare la chiave KMS. Questi sono elementi standard del documento di policy per una Chiave gestita da AWS.

Di seguito è riportata una politica chiave per un Chiave gestita da AWS esempio di 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": "*"
    }
  ]
}
```

Contesto di crittografia di Secrets Manager

Un contesto di crittografia è un set di coppie chiave-valore che contiene dati arbitrari non segreti. Quando si include un contesto di crittografia in una richiesta di crittografia dei dati, associa AWS KMS crittograficamente il contesto di crittografia ai dati crittografati. lo stesso contesto di crittografia sia necessario per decrittografare i dati.

Nelle sue richieste GenerateDataKeye Decrypt a AWS KMS, Secrets Manager utilizza un contesto di crittografia con due coppie nome-valore che identificano il segreto e la relativa versione, come illustrato nell'esempio seguente. I nomi non variano, ma i valori del contesto di crittografia combinati saranno diversi per ogni valore del segreto.

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-
a1b2c3",
    "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
}
```

Puoi utilizzare il contesto di crittografia per identificare queste operazioni crittografiche nei record e nei log di controllo, come AWS CloudTrailAmazon CloudWatch Logs, e come condizione per l'autorizzazione nelle politiche e nelle concessioni.

Il contesto di crittografia di Secrets Manager è costituito da due coppie nome-valore.

- SecretArn – La coppia nome-valore identificherà il segreto. La chiave è `SecretARN`. Il valore è l'Amazon Resource Name (ARN) del segreto.

  ```
  "SecretARN": "ARN of an Secrets Manager secret"
  ```

  Ad esempio, se l'ARN del segreto è `arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3`, il contesto di crittografia include la seguente coppia.

  ```
  "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3"
  ```

- SecretVersionId— La seconda coppia nome-valore identifica la versione del segreto. La chiave è `SecretVersionId`. Il valore è l'ID della versione.

  ```
  "SecretVersionId": "<version-id>"
  ```

  Ad esempio, se l'ID della versione è `EXAMPLE1-90ab-cdef-fedc-ba987SECRET1`, il contesto di crittografia include la seguente coppia.

  ```
  "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
  ```

Quando stabilisci o modifichi la chiave KMS per un segreto, Secrets Manager invia GenerateDataKeye AWS KMS decrittografa le richieste per verificare che il chiamante sia autorizzato a utilizzare la chiave KMS per queste operazioni. Scarta le risposte, non le utilizza sul valore del segreto.

In queste richieste di convalida, il valore di `SecretARN` è l'ARN effettivo del segreto, ma il valore `SecretVersionId` è `RequestToValidateKeyAccess`, come visualizzato nel seguente contesto di crittografia di esempio. Questo valore speciale consente di identificare le richieste di convalida nei log e negli audit trail.

```
"encryptionContext": {
    "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-secret-a1b2c3",
    "SecretVersionId": "RequestToValidateKeyAccess"
}
```

> **Note**
>
> In passato, le richieste di convalida di Secrets Manager non includevano un contesto di crittografia. È possibile trovare chiamate senza contesto di crittografia nei registri più vecchi. AWS CloudTrail

Monitora l'interazione di Secrets Manager con AWS KMS

Puoi utilizzare AWS CloudTrail Amazon CloudWatch Logs per tenere traccia delle richieste a cui Secrets Manager invia per tuo AWS KMS conto. Per ulteriori informazioni sul monitoraggio dell'uso dei segreti , consulta Monitorare i segreti.

GenerateDataKey

Quando crei o modifichi il valore segreto in un segreto, Secrets Manager invia una GenerateDataKeyrichiesta a AWS KMS cui specifica la chiave KMS per il segreto.

L'evento che registra l'operazione `GenerateDataKey` è simile a quello del seguente evento di esempio. La richiesta viene richiamata da `secretsmanager.amazonaws.com`. I parametri includono l'Amazon Resource Name (ARN) della chiave KMS per il segreto, un identificatore della chiave che richiede una chiave a 256 bit e il contesto di crittografia che identifica il segreto e la versione.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AROAIGDTESTANDEXAMPLE:user01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/user01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-05-31T23:23:41Z"
            }
        },
        "invokedBy": "secretsmanager.amazonaws.com"
    },
    "eventTime": "2018-05-31T23:23:41Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "secretsmanager.amazonaws.com",
    "userAgent": "secretsmanager.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "SecretARN": "arn:aws:secretsmanager:us-east-2:111122223333:secret:test-
secret-a1b2c3",
            "SecretVersionId": "EXAMPLE1-90ab-cdef-fedc-ba987SECRET1"
        }
    },
    "responseElements": null,
    "requestID": "a7d4dd6f-6529-11e8-9881-67744a270888",
    "eventID": "af7476b6-62d7-42c2-bc02-5ce86c21ed36",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


Decrypt

Quando si ottiene o si modifica il valore segreto di un segreto, Secrets Manager invia una richiesta di decrittografia per AWS KMS decrittografare la chiave dati crittografata. Per i comandi batch, Secrets Manager può riutilizzare la chiave decrittografata, in modo che non tutte le chiamate generino una richiesta `Decrypt`.

L'evento che registra l'operazione `Decrypt` è simile a quello del seguente evento di esempio. L'utente è il principale del tuo AWS account che accede alla tabella. I parametri includono la chiave crittografata della tabella (come blob di testo cifrato) e il contesto di crittografia che identifica la tabella e l'account. AWS AWS KMS ricava l'ID della chiave KMS dal testo cifrato.

```
{
    "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"
```

```
}
```

Crittografa

Quando si modifica la chiave KMS associata a un segreto, Secrets Manager invia una richiesta di crittografia per AWS KMS crittografare nuovamente le versioni `AWSPENDING` segrete con la `AWSCURRENT` nuova chiave. `AWSPREVIOUS` Quando replichi un segreto in un'altra regione, Secrets Manager invia anche una richiesta Encrypt a AWS KMS.

L'evento che registra l'operazione `Encrypt` è simile a quello del seguente evento di esempio. L'utente è il principale del tuo AWS account che accede alla tabella.

```
{
    "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"
}
```


Sicurezza dell'infrastruttura in AWS Secrets Manager

Come servizio gestito, AWS Secrets Manager è protetto dalla sicurezza di rete globale AWS. Per informazioni sui servizi di sicurezza AWS e su come AWS protegge l'infrastruttura, consulta la pagina Sicurezza del cloud AWS. Per progettare l'ambiente AWS utilizzando le best practice per la sicurezza dell'infrastruttura, consulta la pagina Protezione dell'infrastruttura nel Pilastro della sicurezza di AWS Well-Architected Framework.

Acesso a Secrets Manager mediante la rete avviene attraverso le AWSAPI pubblicate usando TLS. Queste API di Secrets Manager possono essere invocate da qualsiasi posizione di rete. Tuttavia, Secrets Manager non supporta le policy di accesso basate sulle risorse, che possono includere limitazioni in base all'indirizzo IP di origine. È inoltre possibile utilizzare le policy delle risorse di Secrets Manager per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) o VPC specifici. Di fatto, questo isola l'accesso di rete a un segreto specificato solo dal VPC specifico all'interno della rete AWS. Per ulteriori informazioni, consulta Endpoint VPC.

Resilienza in AWS Secrets Manager

AWS crea l'infrastruttura globale intorno alle AWS Regioni e alle zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate che si connettono a reti altamente

ridondanti, a bassa latenza e throughput elevato. Con le Zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità consentono una maggiore disponibilità, tolleranza ai guasti e scalabilità rispetto alle infrastrutture a data center singolo o multiplo.

Per ulteriori informazioni sulla resilienza e il ripristino di emergenza, consultare Pilastro di affidabilità - AWS Framework Well-Architected.

Per ulteriori informazioni sulle Regioni AWS e sulle zone di disponibilit, consulta Infrastruttura globale di AWS.

TLS post-quantistico

Secrets Manager supporta un'opzione di scambio di chiavi post-quantistiche ibride per il protocollo di crittografia di rete Transport Layer Security (TLS). Puoi utilizzare questa opzione TLS quando ti connetti agli endpoint API di Secrets Manager. Offriamo questa funzionalità prima che gli algoritmi post-quantistici siano standardizzati in modo da poter iniziare a testare l'effetto di questi protocolli di scambio di chiavi sulle chiamate di Secrets Manager. Queste caratteristiche opzionali di scambio di chiavi post-quantistiche ibride sono sicure almeno quanto la crittografia TLS che utilizziamo oggi e potrebbero fornire ulteriori vantaggi per la sicurezza. Tuttavia, influenzano la latenza e il throughput rispetto ai protocolli di scambio di chiavi classici in uso oggi.

Per proteggere i dati crittografati oggi contro potenziali attacchi futuri, AWS sta partecipando con la comunità crittografica allo sviluppo di algoritmi resistenti alla quantistica o post-quantistici. Abbiamo implementato suite di crittografia di scambio di chiavi post-quantistiche ibride negli endpoint di Secrets Manager. Queste suite di crittografia ibride, che combinano elementi classici e post-quantistici, assicurano che la connessione TLS sia potente almeno quanto lo sarebbe con le suite di crittografia classiche. Tuttavia, poiché le funzionalità delle prestazioni e i requisiti di larghezza di banda delle suite di crittografia ibride sono diversi da quelli dei classici meccanismi di scambio di chiavi, consigliamo di testarli nelle chiamate API.

Secrets Manager supporta PQTLS in tutte le Regioni, ad eccezione di quelle cinesi.

Per configurare il protocollo TLS post-quantistico ibrido

1. Aggiungere il AWS Common Runtime del cliente alle dipendenze di Maven. Si consiglia di utilizzare l'ultima versione disponibile. Ad esempio, questa istruzione aggiunge la versione 2.20.0.

```
<dependency>
  <groupId>software.amazon.awssdk</groupId>
  <artifactId>aws-crt-client</artifactId>
  <version>2.20.0</version>
</dependency>
```

2. AggiungiAWS SDK for Java 2.x al progetto e inizializzalo. Abilita le suite di crittografia post-quantistica ibrida su tuo client HTTP.

```
SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
            .postQuantumTlsEnabled(true)
            .build();
```

3. Crea il client asincrono di Secrets Manager.

```
SecretsManagerAsyncClient SecretsManagerAsync = SecretsManagerAsyncClient.builder()
            .httpClient(awsCrtHttpClient)
            .build();
```

   Ora quando richiami le operazioni API di Secrets Manager, le chiamate vengono trasmesse all'endpoint di Secrets Manager utilizzando TLS post-quantistico ibrido.

Per ulteriori informazioni sull'utilizzo del protocollo TLS post-quantistico ibrido, consulta:

- Guida per gli sviluppatori di AWS SDK for Java 2.x e il post del blog AWS SDK for Java 2.x rilasciato.
- Presentazione di s2n-tls, una nuova implementazione TLS open source e Utilizzo di s2n-tls.
- Crittografia post-quantistica sul National Institute for Standards and Technology (NIST).
- Post-Quantum Key Encapsulation Methods (PQ KEM) ibrido per Transport Layer Security (TLS) 1.2.

Il protocollo TLS post-quantistico per Secrets Manager è disponibile in tutte le Regioni AWS ad eccezione della Cina.

Risoluzione dei problemi relativi a AWS Secrets Manager

Utilizza le informazioni contenute in questa pagina per diagnosticare e risolvere i problemi che possono verificarsi durante l'utilizzo di ruoli con Secrets Manager.

Per i problemi relativi alla rotazione, consulta the section called “Risoluzione dei problemi della rotazione”.

Argomenti

- Messaggi di “Accesso negato” quando si inviano richieste a Secrets Manager
- “Accesso negato” per le credenziali di sicurezza temporanee
- Le modifiche apportate non sono sempre immediatamente visibili.
- “Impossibile generare una chiave dati con una chiave KMS asimmetrica” durante la creazione di un segreto
- Un AWS CLI o AWS operazione SDK non riesce a trovare il mio segreto da un ARN parziale
- Questo segreto è gestito da un servizio AWS ed è necessario utilizzare tale servizio per aggiornarlo.

Messaggi di “Accesso negato” quando si inviano richieste a Secrets Manager

Verifica di disporre delle autorizzazioni necessarie per chiamare l'operazione e le risorse richieste. È necessario che un amministratore conceda le autorizzazioni collegando una policy IAM all'utente IAM o a un gruppo a cui appartieni. Se le dichiarazioni di policy che concedono tali autorizzazioni includono eventuali condizioni, ad esempio limitazioni relative a indirizzo IP o ora del giorno, sarà inoltre necessario che tali requisiti siano soddisfatti, quando invii la richiesta. Per informazioni sulla visualizzazione o sulla modifica delle policy per un ruolo, un gruppo o un utente IAM, consulta Lavorare con le policy nella Guida per l'utente di IAM. Per informazioni sulle autorizzazioni richieste per Secrets Manager, consulta Autenticazione e controllo degli accessi.

Se firmi manualmente le richieste API, senza utilizzare gli AWSSDK, verifica di aver firmato la richiesta correttamente.

“Accesso negato” per le credenziali di sicurezza temporanee

Verifica che l'utente o il ruolo IAM utilizzato per effettuare la richiesta disponga delle autorizzazioni corrette. Autorizzazioni per credenziali di sicurezza temporanee derivano da un utente o ruolo IAM. Questo significa che le autorizzazioni sono limitate a quelle concesse al ruolo o all'utente IAM. Per ulteriori informazioni su come sono determinate le autorizzazioni per le credenziali di sicurezza provvisorie, consulta controllo delle autorizzazioni per le credenziali di sicurezza provvisorie nella Guida IAM per lo sviluppatore.

Verifica che le richieste vengano firmate correttamente e che il formato della richiesta sia valido. Per ulteriori informazioni, consultare la documentazione del kit di strumenti dell'SDK scelto. In alternativa, consultare Utilizzo di credenziali di sicurezza provvisorie per richiedere l'accesso alle risorse di AWS nella Guida IAM per lo sviluppatore.

Verifica che le credenziali di sicurezza provvisorie non siano scadute. Per ulteriori informazioni, consulta Richiesta di credenziali di sicurezza provvisorie nella Guida per l'utente di IAM.

Per informazioni sulle autorizzazioni richieste per Secrets Manager, consulta Autenticazione e controllo degli accessi.

Le modifiche apportate non sono sempre immediatamente visibili.

Secrets Manager utilizza un modello di calcolo distribuito chiamato consistenza finale. È necessario del tempo affinché le modifiche apportate in Secrets Manager (o in altri servizi AWS) risultino visibili da tutti i possibili endpoint. Alcuni dei ritardi sono dovuti al tempo necessario per inviare i dati da un server a un altro, da una zona di replica a un'altra e da una regione a un'altra nel mondo. Secrets Manager utilizza inoltre la memorizzazione nella cache per migliorare le prestazioni, è possibile che ciò aumenti il tempo. in quanto la modifica potrebbe risultare visibile solo dopo il timeout dei dati memorizzati nella cache.

Progetta le tue applicazioni globali in modo da considerare questi potenziali ritardi e assicurati che funzionino come previsto, anche quando una modifica apportata in una posizione non è immediatamente visibile in un'altra.

Per ulteriori informazioni su come altri servizi AWS sono influenzati dalla consistenza finale, consulta:

- Gestione della consistenza dei dati nella Guida per sviluppatori di Amazon Redshift Database
- Modello di consistenza dati di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service

- Garantire la consistenza quando si utilizzano Amazon S3 e Amazon EMR per flussi di lavoro ETL nel blog dei Big Data AWS.
- Consistenza finale Amazon EC2 nella Riferimento all'API di Amazon EC2

“Impossibile generare una chiave dati con una chiave KMS asimmetrica” durante la creazione di un segreto

Secrets Manager utilizza una chiave KMS di crittografia simmetrica associata a un segreto per generare una chiave di dati per ogni valore del segreto. Non puoi utilizzare una chiave KMS asimmetrica. Verifica di utilizzare una chiave KMS di crittografia simmetrica anziché una chiave KMS asimmetrica. Per le istruzioni, consulta Individuazione delle chiavi KMS asimmetriche.

Un AWS CLI o AWS operazione SDK non riesce a trovare il mio segreto da un ARN parziale

In molti casi, Secrets Manager può trovare il segreto da una parte di un ARN anziché dall'ARN completo. Tuttavia, se il nome del tuo segreto termina con un trattino seguito da sei caratteri, Secrets Manager potrebbe non essere in grado di individuare il segreto da una sola parte di un ARN. Invece, ti consigliamo di utilizzare l'ARN completo o il nome del segreto.

Ulteriori dettagli

Secrets Manager include sei caratteri casuali alla fine del nome del segreto per garantire che l'ARN del segreto sia univoco. Se il segreto originale viene eliminato e quindi viene creato un nuovo segreto con lo stesso nome, i due segreti hanno ARN diversi grazie a questi caratteri. Gli utenti con accesso al vecchio segreto non ottengono automaticamente l'accesso al nuovo segreto perché gli ARN sono diversi.

Secrets Manager costruisce un ARN per un segreto con Regione, account, nome segreto e poi un trattino e altri sei caratteri, come segue:

```
arn:aws:secretsmanager:us-east-2:111122223333:secret:SecretName-abcdef
```

Se il tuo nome segreto termina con un trattino e sei caratteri, l'utilizzo di una sola parte dell'ARN può apparire in Secrets Manager come se si stesse specificando un ARN completo. Ad esempio, potresti avere un segreto denominato `MySecret-abcdef` con l'ARN

`arn:aws:secretsmanager:us-east-2:111122223333:secret:MySecret-abcdef-nutBrk`

Se si chiama la seguente operazione, che utilizza solo una parte dell'ARN segreto, Secrets Manager potrebbe non trovare il segreto.

```
$ aws secretsmanager describe-secret --secret-id arn:aws:secretsmanager:us-
east-2:111122223333:secret:MySecret-abcdef
```

Questo segreto è gestito da un servizio AWS ed è necessario utilizzare tale servizio per aggiornarlo.

Se questo messaggio viene visualizzato mentre provi a modificare un segreto, il segreto potrà essere aggiornato solo utilizzando il servizio di gestione riportato nel messaggio. Per ulteriori informazioni, consulta Segreti gestiti da altri servizi.

Per determinare chi gestisce un segreto, puoi rivedere il nome del segreto. I segreti gestiti da altri servizi sono preceduti dall'ID di quel servizio. Oppure, nella AWS CLI esegui descrive-secret e rivedi il campo `OwningService`.

Quote AWS Secrets Manager

Le API di lettura di Secrets Manager hanno quote TPS elevate mentre le API del piano di controllo che vengono chiamate meno frequentemente hanno quote TPS inferiori. Ti consigliamo di evitare di chiamare `PutSecretValue` o `UpdateSecret` ad una frequenza sostenuta di più di una volta ogni 10 minuti. Quando chiami `PutSecretValue` o `UpdateSecret` per aggiornare il valore del segreto, Secrets Manager crea una nuova versione del segreto. Secrets Manager rimuove le versioni obsolete quando sono più di 100, ma non rimuove le versioni create da meno di 24 ore. Se aggiorni il valore segreto più di una volta ogni 10 minuti, crei più versioni di quelle che Secrets Manager rimuove e raggiungerai la quota massima prevista per le versioni di un segreto.

Puoi utilizzare più regioni nel tuo account e ogni quota sarà specifica per ogni regione.

Quando un'applicazione in un Account AWS utilizza un segreto di proprietà di un altro account, si parla di richiesta tra account. Per le richieste tra account, Secrets Manager limita l'account che effettua le richieste di identità, non l'account proprietario del segreto. Ad esempio, se un'identità nell'account A utilizza un segreto nell'account B, l'uso del segreto viene applicato solo alle quote dell'account A.

Quote di Secrets Manager

Nome	Default	Adatta e	Descrizione
Frequenza combinata delle richieste API DeleteResourcePolicy, GetResourcePolicy, PutResourcePolicy e ValidateResourcePolicy	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API DeleteResourcePolicy, GetResourcePolicy, PutResourcePolicy e ValidateResourcePolicy combinate.
Frequenza combinata delle richieste API DescribeSecret e GetSecretValue	Ogni regione supportata: 10.000 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API DescribeSecret e

Nome	Default	Adatta e	Descrizione
			GetSecretValue combinate.
Frequenza combinata di richieste API PutSecretValue, RemoveRegionsFromReplication, ReplicateSecretToRegion, StopReplicationToReplica, UpdateSecret e UpdateSecretVersionStage	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API PutSecretValue, RemoveRegionsFromReplication, ReplicateSecretToRegion, StopReplicationToReplica, UpdateSecret e UpdateSecretVersionStage combinate.
Frequenza combinata di richieste API RestoreSecret	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API RestoreSecret.
Frequenza combinata delle richieste API RotateSecret e CancelRotateSecret	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API RotateSecret e CancelRotateSecret combinate.
Frequenza combinata di richieste API TagResource e UntagResource	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API TagResource e UntagResource combinate.

Nome	Default	Adatta e	Descrizione
Frequenza delle richieste API BatchGetSecretValue	Ogni regione supportata: 100 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API BatchGetSecretValue.
Frequenza delle richieste API CreateSecret	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API CreateSession.
Frequenza delle richieste API DeleteSecret	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API DeleteSecret.
Frequenza delle richieste API GetRandomPassword	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API GetRandomPassword.
Frequenza delle richieste API ListSecretVersionIds	Ogni regione supportata: 50 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API ListSecretVersionIds.
Frequenza delle richieste API ListSecrets	Ogni regione supportata: 100 al secondo	No	Il numero massimo di transazioni al secondo per le richieste API ListSecrets.
Lunghezza policy basata su risorse	Ogni regione supportata: 20.480	No	Il numero massimo di caratteri in una policy di autorizzazioni basata su risorse collegata a un segreto.

Nome	Default	Adatta e	Descrizione
Dimensione del valore del segreto	Ogni regione supportata: 65.536 byte	No	La dimensione massima di un valore del segreto crittografato. Se il valore del segreto è una stringa, questo è il numero di caratteri consentiti nel valore del segreto.
Segreti	Ogni regione supportata: 500.000	No	Il numero massimo di segreti in ogni regione AWS di questo account AWS.
Etichette allegate su tutte le versioni di un segreto	Ogni regione supportata: 20	No	Il numero massimo di etichette di staging allegate su tutte le versioni di un segreto.
Versioni per segreto	Ogni regione supportata: 100	No	Il numero massimo di versioni di un segreto.

Aggiungi tentativi alla tua applicazione

Il tuo AWS client potrebbe vedere le chiamate a Secrets Manager fallire a causa di problemi imprevisti sul lato client. Il tuo potrebbe vedere che le chiamate a Secrets Manager non riescono a causa della limitazione della velocità. Quando superi una quota di richiesta API, Secrets Manager limita la richiesta. Rifiuta una richiesta altrimenti valida e restituisce unthrottlingErrore. Per entrambi i tipi di guasti, ti consigliamo di riprovare la chiamata dopo un breve periodo di attesa. Questo è chiamato a[strategia di backoff e riprova](#).

Se si verificano errori come quelli riportati di seguito, potrebbe essere necessario aggiungere al codice dell'applicazione:

Errori ed eccezioni transitori

- `RequestTimeout`
- `RequestTimeoutException`
- `PriorRequestNotComplete`
- `ConnectionError`
- `HTTPClientError`

Limitazione lato servizio e limitazione di errori ed eccezioni

- `Throttling`
- `ThrottlingException`
- `ThrottledException`
- `RequestThrottledException`
- `TooManyRequestsException`
- `ProvisionedThroughputExceededException`
- `TransactionInProgressException`
- `RequestLimitExceeded`
- `BandwidthLimitExceeded`
- `LimitExceededException`
- `RequestThrottled`
- `SlowDown`

Per ulteriori informazioni, oltre al codice di esempio, su tentativi, backoff esponenziale e jitter, vedere le seguenti risorse:

- Backoff esponenziale e jitter
- Timeout, tentativi e backoff con jitter
- Ripetizione dei tentativi in caso di errore e backoff esponenziale in AWS.

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di AWS Secrets Manager. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Passaggio da Secrets Manager a policy AWS gestita	La policy `SecretsManagerReadWrite` gestita ora include `redshift-serverless` l'autorizzazione. Per ulteriori informazioni, consulta la politica AWS gestita per AWS Secrets Manager	12 marzo 2024

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti apportate in ogni versione della Guida per l' AWS Secrets Manager utente prima di febbraio 2024.

Modifica	Descrizione	Data
Disponibilità generale	Questa è la versione pubblica iniziale di Secrets Manager.	4 aprile 2018

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.