Protezione dei dati sensibili utilizzando AWS Secrets Manager e HashiCorp Terraform

AWS Linee guida prescrittive

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Linee guida prescrittive: Protezione dei dati sensibili utilizzando AWS Secrets Manager e HashiCorp Terraform

Copyright © 2026 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 dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Introduzione 1
Destinatari principali 1
Obiettivi 1
Utilizzo di Secrets Manager e Terraform 3
AWS Secrets Manager 3
Terraform 3
Best practice 4
Generazione di password casuali 4
segreti rotanti 6
Limitazione dell'accesso 8
Utilizzo di una convenzione di denominazione 9
Gestione dei dati sensibili 11
Protezione dei dati sensibili in Terraform file di stato 11
Gestione dei segreti per Amazon EKS 12
Conservazione dei dati sensibili in reti note 14
Risorse 16
AWS documentazione 16
AWS post sul blog 16
Altre risorse 16
Collaboratori 17
Scrittura 17
Revisione 17
Scrittura tecnica 17
Cronologia dei documenti 18
..... xix

Protezione dei dati sensibili utilizzando AWS Secrets Manager e HashiCorp Terraform

Amazon Web Services (collaboratori)

Maggio 2025 (cronologia dei documenti)

La gestione dei dati sensibili, tra cui credenziali, stringhe segrete e password, è un pilastro riconosciuto della gestione dell'infrastruttura e dello sviluppo e dell'implementazione delle applicazioni. Per proteggere la tua organizzazione, adotta le migliori pratiche per la gestione dei dati sensibili nel cloud. La protezione dei dati sensibili è un prerequisito per la sicurezza e la conformità. AWS Secrets Managerpuò contribuire a proteggere i dati sensibili presenti nell'ambiente in uso come segreti.

Questa guida esamina le migliori pratiche relative ai segreti, ad esempio come ottenere segreti da Secrets Manager e come utilizzarli AWS Lambda per ruotare automaticamente i segreti per i dati sensibili. Fornisce inoltre consigli su come gestire e governare i segreti utilizzando nomi gerarchici. Infine, aiuta a gestire l'uso e l'accesso ai segreti, ad esempio considerazioni relative alla centralizzazione, Terraform all'integrazione e alla rete.

HashiCorp Terraformè stata ampiamente adottata come soluzione Infrastructure as Code (IaC) nel settore. Tuttavia, Terraform mostra i dati sensibili come testo semplice nel relativo file di stato. Questa guida contiene le migliori pratiche Terraform da utilizzare per gestire dati sensibili e per creare e utilizzare i segreti di Secrets Manager.

Destinatari principali

Questa guida è destinata alle organizzazioni che desiderano utilizzarla Terraform come soluzione IaC. Le migliori pratiche riportate in questa guida sono progettate per aiutare gli architetti di database, i team di infrastruttura e gli sviluppatori di applicazioni. La familiarità con Terraform è un prerequisito per questa guida.

Obiettivi

Di seguito sono riportati i risultati aziendali che ci si può aspettare di ottenere dopo l'implementazione delle raccomandazioni contenute in questa guida:

- Innova più velocemente automatizzando il processo di gestione dei segreti.
- Migliora il livello di sicurezza della tua organizzazione in. Cloud AWS

Di seguito sono riportati i risultati tecnici che puoi aspettarti di ottenere dopo aver implementato le raccomandazioni contenute in questa guida:

- Usa Secrets Manager per prevenire l'esposizione di dati sensibili nel file di Terraform stato.
- Centralizza la gestione dei segreti e dei dati sensibili per migliorare la governance e raggiungere la conformità.
- Applica le migliori pratiche di sicurezza nei processi della tua organizzazione per l'implementazione dell'infrastruttura cloud.

Utilizzo di Secrets Manager e Terraform

AWS Secrets Manager

AWS Secrets Manager è un servizio per crittografare, archiviare e ruotare in modo sicuro le credenziali per database e altri servizi. Ti aiuta a sostituire le credenziali codificate nel codice, comprese le password, con una chiamata API per recuperare il segreto a livello di codice. In Secrets Manager, un segreto è costituito dalle informazioni sulle credenziali (che è il valore segreto) e dai relativi metadati. Il valore segreto può essere binario, una stringa singola o più stringhe. Per ulteriori informazioni, consulta Secret.

Secrets Manager utilizza chiavi dati simmetriche Advanced Encryption Standard (AES) a 256 bit per crittografare i valori segreti. Per ulteriori informazioni consulta la sezione Crittografia e decrittografia dei segreti in AWS Secrets Manager.

È possibile accedere e lavorare con Secrets Manager utilizzando uno dei seguenti approcci:

- Console Secrets Manager
- Strumenti a riga di comando
- AWS SDKs
- API di interrogazione HTTPS, chiamata anche API Secrets Manager
- AWS Secrets Manager endpoint

Terraform

Terraformè uno strumento IaC HashiCorp che consente di creare e gestire risorse cloud e locali. È possibile utilizzare Terraform per distribuire risorse e infrastrutture in. Cloud AWS

Terraformarchivia informazioni sull' AWS infrastruttura gestita e sulle relative configurazioni. Queste informazioni sono chiamate stato. Per impostazione predefinita, lo stato viene memorizzato in un file locale denominato`Terraform.tfstate`. Questo file è in formato JSON e Terraform potrebbe memorizzare dati sensibili in questo file di stato in formato testo semplice. Ciò rappresenta un rischio per i dati sensibili perché qualsiasi utente con accesso al file di stato può accedere ai dati sensibili.

Questa guida fornisce le migliori pratiche e consigli per aiutarti a proteggere i dati sensibili durante Terraform la gestione AWS delle risorse.

Procedure consigliate per creare, ruotare e utilizzare segreti

Questa sezione illustra le migliori pratiche e gli standard che è possibile seguire durante la codifica di applicazioni e infrastrutture per la distribuzione. Terraform Contiene i seguenti dati:

- Generazione di password casuali utilizzando AWS Secrets Manager: genera un segreto casuale di complessità specificata, in base alla politica di conformità dell'organizzazione.
- Utilizzo AWS Lambda per ruotare i segreti: AWS Lambda può essere programmato per l'esecuzione automatica per la rotazione dei segreti.
- Limitazione dell'accesso ai segreti: limita chi può accedere ai segreti e come accedervi.
- Utilizzo di una convenzione di denominazione gerarchica per i segreti: definisci e utilizza una convenzione di denominazione che ti aiuti a gestire i segreti su larga scala.

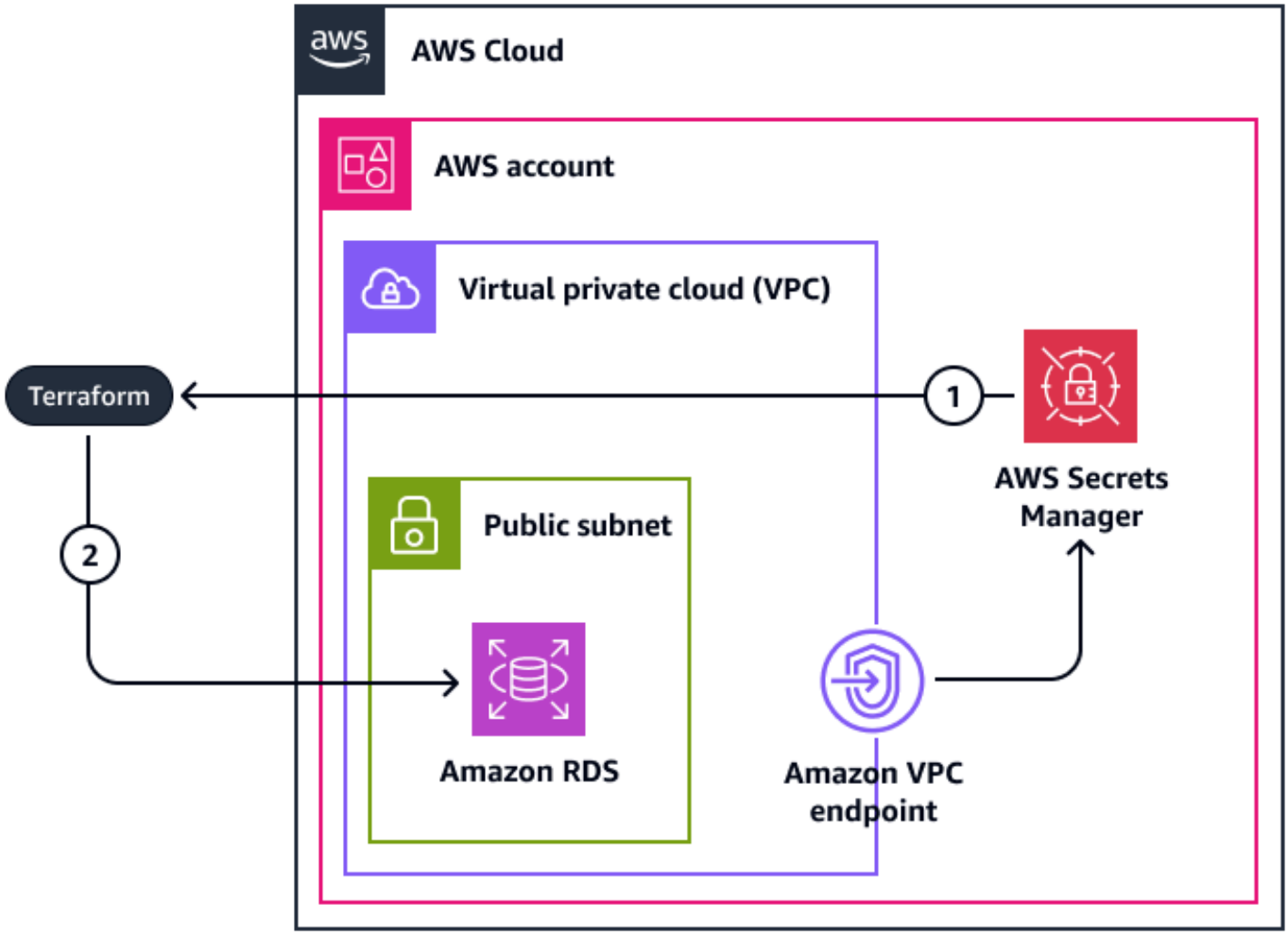
Generazione di password casuali utilizzando AWS Secrets Manager

Il AWS Well-Architected Framework consiglia di archiviare e utilizzare i segreti in modo sicuro. È possibile utilizzare l' AWS Secrets Manager API per generare password casuali e personalizzare i requisiti di complessità delle password. L'`GetRandomPassword`dazione supporta stringhe di password di lunghezza compresa tra 1 e 4.096 caratteri. Per ulteriori informazioni, consulta GetRandomPassword nella documentazione di riferimento dell'API AWS Secrets Manager . Si consiglia di utilizzare questo approccio per generare segreti anziché consentire agli utenti di definirli manualmente.

Il seguente esempio di codice mostra come generare password casuali di 20 caratteri che includono numeri, escludono caratteri di punteggiatura ed escludono spazi. È possibile modificare questo esempio di codice per soddisfare i requisiti di sicurezza delle password per l'organizzazione.

```
data "aws_secretsmanager_random_password" "test" {
password_length = 20
exclude_numbers = false
exclude_punctuation = true
include_space = false
}
```

L'utilizzo della generazione casuale di segreti durante l'implementazione di IaC aiuta a proteggere i dati sensibili sin dall'inizio, operazione nota come zero ore. I dati sensibili non sono mai noti a nessuno, fin dalla fase di implementazione.



1. TramiteTerraform, AWS Secrets Manager utilizzare per generare una password segreta casuale.
2. Terraformutilizza questa password segreta casuale, memorizzata in AWS Secrets Manager, per accedere al database.

> **Important**
>
> Quando si utilizza Terraform come fonte di dati, i segreti non vengono memorizzati nel file di stato. Tuttavia, dopo aver utilizzato quel segreto in un database o in qualsiasi servizio, viene archiviato nel file di stato. Ti consigliamo di ruotare immediatamente i segreti o di

creare autorizzazioni molto restrittive per accedere al file di stato. Per ulteriori informazioni sul tagging, consulta Protezione dei dati sensibili in Terraform file di statoin questa guida.

Utilizzo AWS Lambda per ruotare i segreti

Il AWS Well-Architected Framework consiglia di archiviare e utilizzare i segreti in modo sicuro. Questa best practice consiglia di automatizzare la rotazione delle credenziali a intervalli regolari. 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. Molti framework e normative di conformità richiedono inoltre la rotazione dei segreti.

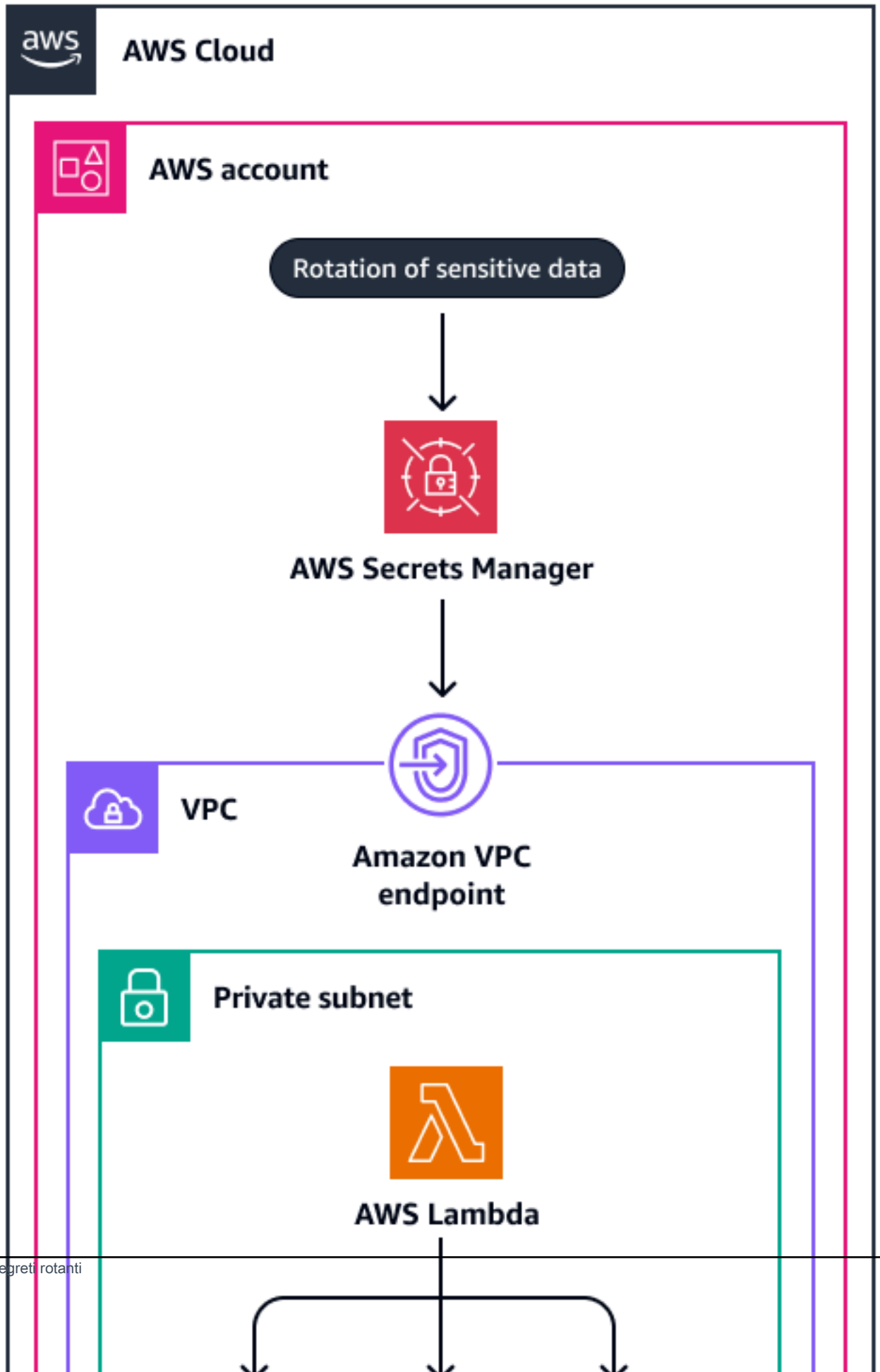
Per Terraform IAc, è possibile utilizzare AWS Secrets Manager e AWS Lambdastabilire la rotazione automatica. In Secrets Manager, è possibile impostare la rotazione automatica per i segreti. Quando Secrets Manager ruota un segreto, vengono aggiornate le credenziali sia del segreto che del database o del servizio.

Per i database, si consiglia di gestire le credenziali primarie in Secrets Manager e di ruotare le chiavi segrete a intervalli regolari. Secrets Manager fornisce modelli di funzioni di rotazione per Lambda per diversi tipi di credenziali del database. Per ulteriori informazioni, consultate i modelli AWS Secrets Manager delle funzioni di rotazione nella documentazione di Secrets Manager e gli esempi di codice in GitHub. Di seguito è riportato un esempio di Terraform IaC che è possibile utilizzare per ruotare dati segreti o sensibili.

```
resource "aws_secretsmanager_secret_rotation" "createrotation" {
  count               = var.needrotation == true ? 1 : 0
  secret_id           = aws_secretsmanager_secret.initiatesecret.id
  rotation_lambda_arn = aws_lambda_function.rotationlambda.arn

  rotation_rules {
    automatically_after_days = 1
  }
}
```

Il seguente diagramma di architettura mostra come utilizzare Secrets Manager, un endpoint Amazon VPC e una funzione Lambda per ruotare i dati sensibili in un. Account AWS


aws
AWS Cloud
AWS account
Rotation of sensitive data
AWS Secrets Manager
VPC
Amazon VPC endpoint
Private subnet
AWS Lambda

Limitare l'accesso ai segreti

La sicurezza è la massima priorità in AWS. Nel AWS Well-Architected Framework, la best practice Archivia e usa i segreti in modo sicuro consiglia di limitare l'accesso ai segreti, registrare e controllare l'accesso ai segreti e di ottenere l'accesso tramite un'API anziché concedere l'accesso a una persona. Ciò è correlato alla generazione di password casuali mediante utilizzo AWS Secrets Manager, illustrata in precedenza in questa guida.

In passato, le credenziali utilizzate per l'autenticazione su database, API di terze parti, token e altri segreti potevano essere incorporate nel codice sorgente o nei file di ambiente. In molte organizzazioni, gli amministratori dei database possono ancora accedere a tutte le credenziali utente del database. Tuttavia, è più sicuro se non lo fanno. Ti consigliamo di:

- Limita l'accesso a Secrets Manager tramite una policy AWS Identity and Access Management (IAM) associata a un ruolo di AWS servizio o a un ruolo IAM per gli utenti.
- Estendi l'accesso programmatico ai segreti in Secrets Manager solo tramite chiamate API a Secrets Manager.
- Evita di usare segreti codificati.
- Non passate valori di credenziali nelle variabili.
- Assicurati che i segreti vengano utilizzati direttamente dall'applicazione o dall'infrastruttura, anziché da un utente umano.

Il seguente esempio di codice mostra un approccio che è possibile utilizzare per proteggere i dati sensibili durante la creazione di un database. Quando si crea il database, è possibile richiedere l'immissione dell'utente come password e quindi ruotare la password, oppure è possibile utilizzare una password casuale dall'inizio. La generazione di una password casuale fin dall'inizio può aiutarvi a gestire e gestire le credenziali in modo più sicuro.

Il seguente frammento di codice crea una risorsa di database e `password = local.passwddata` recupera la password da una variabile locale.

```
resource "aws_db_instance" "default" {
  count                  = var.createdb == true ? 1 : 0
  allocated_storage      = 20
  identifier             = var.identifier
  db_name                = var.databasename
  engine                 = "mysql"
```

```
  engine_version         = "8.0.32"
  instance_class         = "db.t3.micro"
  username               = "admin"
  password               = local.passwddata
  skip_final_snapshot    = true
  publicly_accessible    = false
  vpc_security_group_ids = [aws_security_group.db.id]
  db_subnet_group_name   = aws_db_subnet_group.db.0.name
}
```

Il seguente frammento di codice crea una password casuale o accetta la password fornita dall'utente. Terraformmemorizza questa password come variabile locale in modo da poterla utilizzare più avanti nel codice.

```
locals {
  passwddata = var.passwddata != null ? var.passwddata :
 data.aws_secretsmanager_random_password.test.random_secrets
}
```

Utilizzo di una convenzione di denominazione gerarchica per i segreti

Quando inserite e memorizzate segreti in AWS Secrets Manager, utilizzate una convenzione di denominazione definita. Ad esempio, la tua convenzione di denominazione potrebbe essere. `<client name>/<dev or prod>/<project>/<version>` Scegliere una convenzione di denominazione che abbia senso per la propria organizzazione. Utilizzate una struttura gerarchica che passi dalla maggior parte delle informazioni generali alle informazioni più specifiche.

Ciò consente di gestire i segreti, soprattutto quando un account centralizzato viene utilizzato per molti segreti in tutto l'ambiente. AWS Una convenzione di denominazione gerarchica può aiutarvi nei seguenti modi:

- Ti aiuta a separare l'accesso a diversi segreti. Ad esempio, è possibile definire quale progetto o ambiente deve avere accesso a quali segreti.
- Ti aiuta a stabilire controlli di accesso dettagliati ai segreti in base ai loro nomi.

Il seguente esempio di codice mostra un esempio di convenzione di denominazione che può aiutarvi a gestire i segreti su larga scala.

```
resource "aws_secretsmanager_secret" "initiatesecret" {
  name                    = "org-name/dev-env/project-name"
  kms_key_id              = var.kmskeyarn
  recovery_window_in_days = var.recoverywindow
}
```

Gestione dei dati sensibili durante l'utilizzo Terraform

I dati sensibili devono essere ben architettati. Con la scalabilità orizzontale delle applicazioni e dell'infrastruttura, è sempre più importante tracciare e gestire con attenzione i dati sensibili. È possibile utilizzare i seguenti approcci per proteggere i dati sensibili Account AWS durante l'implementazione di Terraform IaC:

- Protezione dei dati sensibili in Terraform file di stato— Puoi contribuire a proteggere i dati sensibili dal momento in cui vengono inseriti per la prima volta. AWS Secrets Manager Ad esempio, è possibile ruotare immediatamente il segreto per preservarne la segretezza.
- Accesso e gestione dei segreti per Amazon EKS— Gestisci tutti i segreti per Amazon Elastic Kubernetes Service (Amazon EKS) in Secrets Manager.
- Utilizzo degli endpoint VPC per conservare i dati sensibili in reti note— Il traffico di dati sensibili non deve lasciare le reti private. Questo aiuta a prevenire attacchi ed esfiltrazioni di dati.

Protezione dei dati sensibili in Terraform file di stato

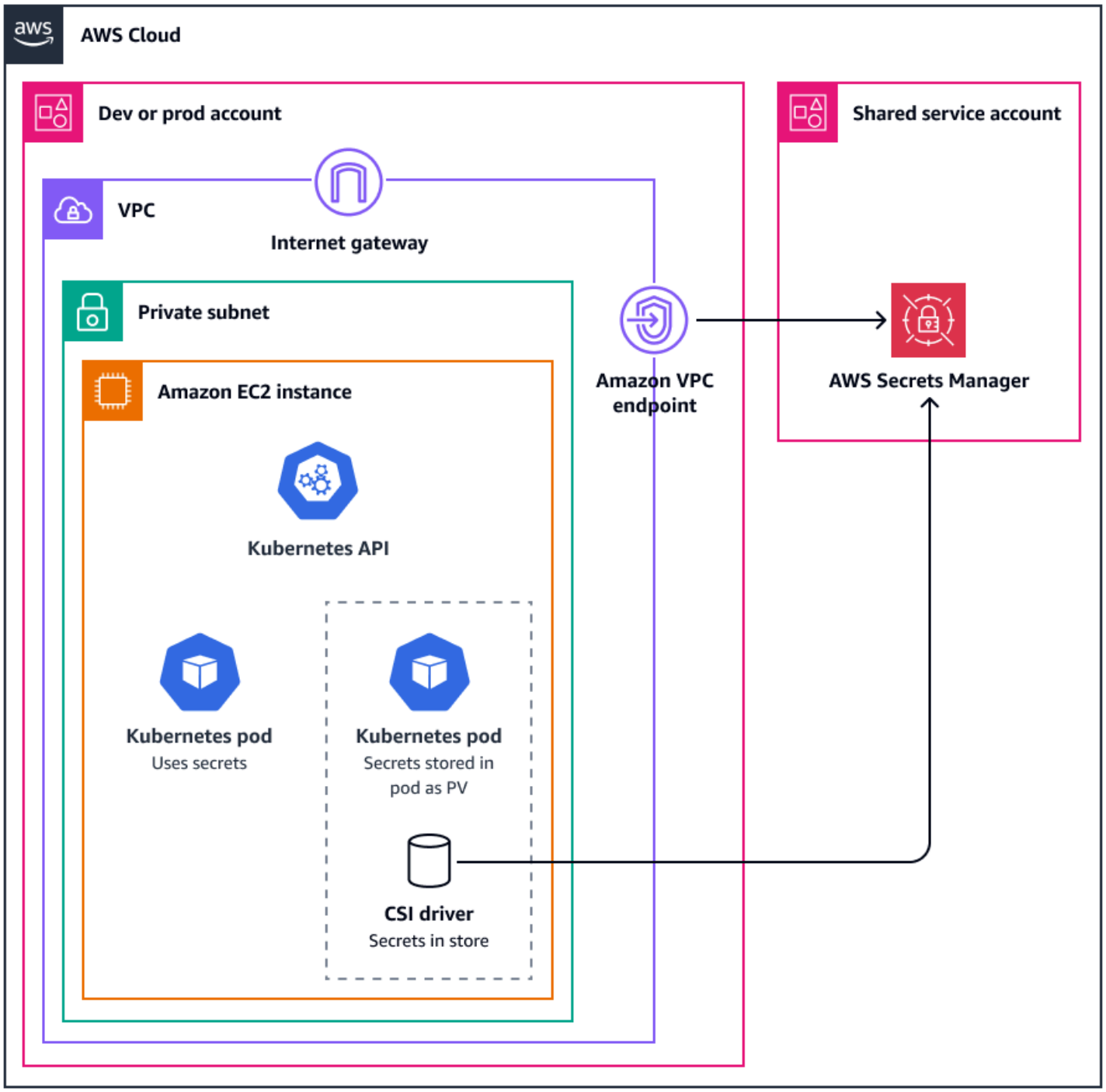
Questa sezione illustra l'offuscamento dei segreti e dei suggerimenti per gestire i dati sensibili nel file di stato, chiamato. Terraform `tfstate` In genere, si tratta di un file di testo semplice che contiene dati sulle Terraform distribuzioni e include tutti i dati sensibili e non sensibili sull'infrastruttura distribuita. I dati sensibili sono visibili in testo semplice nel file di stato. Terraform Per proteggere i dati sensibili, procedi come segue:

- Quando acquisisci un segreto, scegli di ruotarlo immediatamente. Per ulteriori informazioni, consulta Ruotare immediatamente un AWS Secrets Manager segreto nella documentazione di Secrets Manager.
- Archivia il file di Terraform stato nel sistema centralizzato in Account AWS cui gestisci Secrets Manager. Archivia il file in un bucket Amazon Simple Storage Service (Amazon S3) e configura policy che ne limitino l'accesso. Per ulteriori informazioni, consulta le politiche di Bucket e le politiche degli utenti nella documentazione di Amazon S3.
- Puoi bloccare lo Terraform stato per prevenire la corruzione. Per ulteriori informazioni sul blocco dello stato e sulla protezione del file di stato, consulta il backend Amazon S3 nella documentazione. Terraform

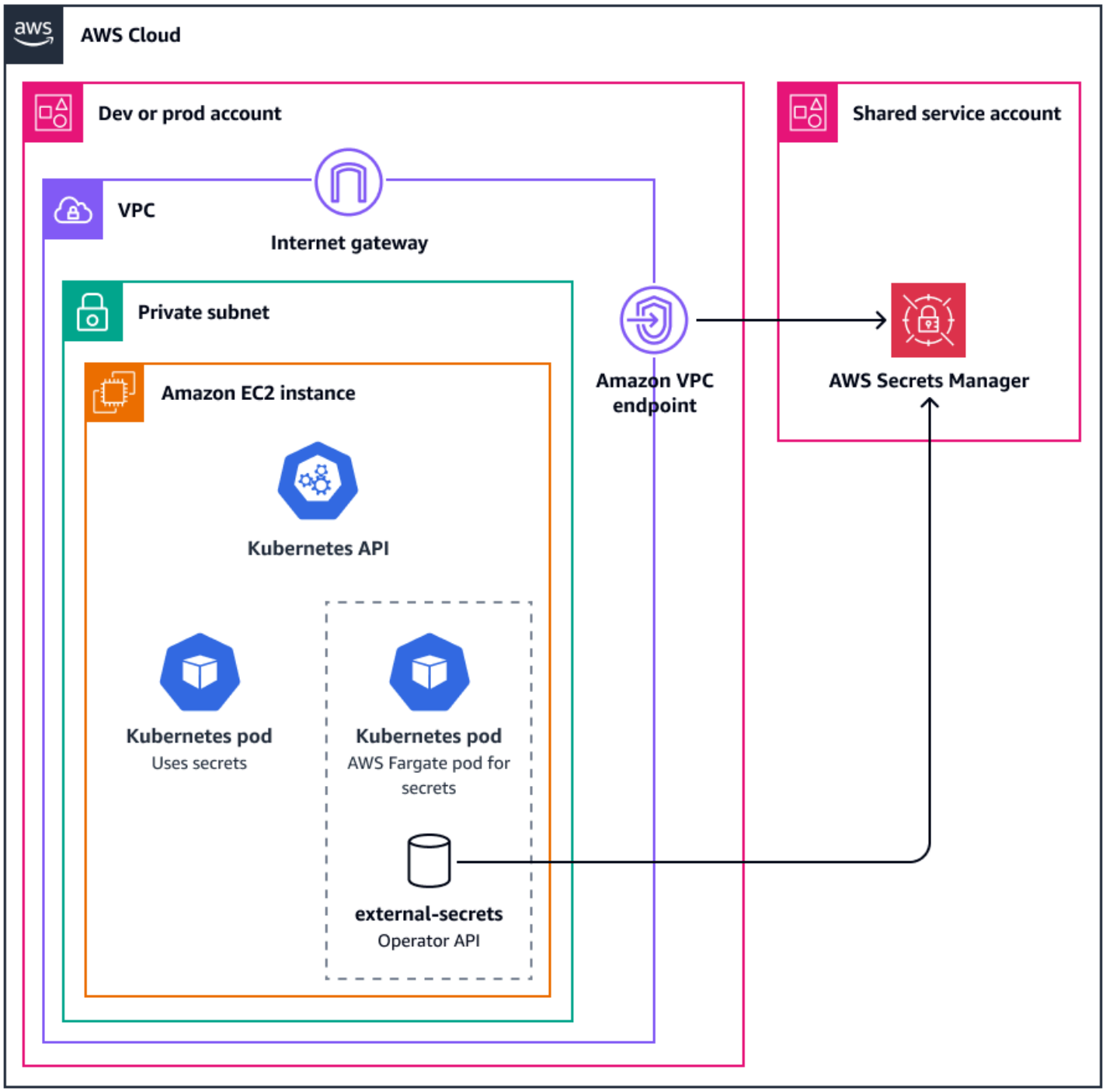
Accesso e gestione dei segreti per Amazon EKS

Amazon Elastic Kubernetes Service (Amazon EKS) ti aiuta a eseguire AWS Kubernetes senza dover installare o gestire il tuo piano di controllo o i tuoi nodi Kubernetes. Amazon EKS utilizza la codifica Base64 (Wikipedia) per proteggere i dati sensibili. La codifica è progettata per impedire la modifica dei dati durante il transito tra i sistemi e la crittografia è progettata per impedire l'accesso non autorizzato ai dati. La codifica Base64 non è sufficiente per proteggere i dati da accessi non autorizzati. AWS Secrets Manager Utilizzalo per proteggere i dati sensibili in Amazon EKS.

L'immagine seguente mostra Amazon EKS distribuito su un'istanza Amazon Elastic Compute Cloud (Amazon EC2), che funge da nodo di lavoro Kubernetes. È possibile utilizzare un driver CSI (Container Storage Interface) per recuperare segreti da Secrets Manager. Per ulteriori informazioni, consulta Come utilizzare AWS Secrets & Configuration Provider con il driver CSI Kubernetes Secrets Store nel Security Blog. AWS



L'immagine seguente mostra Amazon EKS distribuito su AWS Fargate. È possibile utilizzare l'API open source External Secrets Operator (GitHub) per recuperare segreti da Secrets Manager. Per ulteriori informazioni, consulta Sfruttare gli archivi AWS segreti di EKS Fargate con External Secrets Operator nel AWS blog Containers.



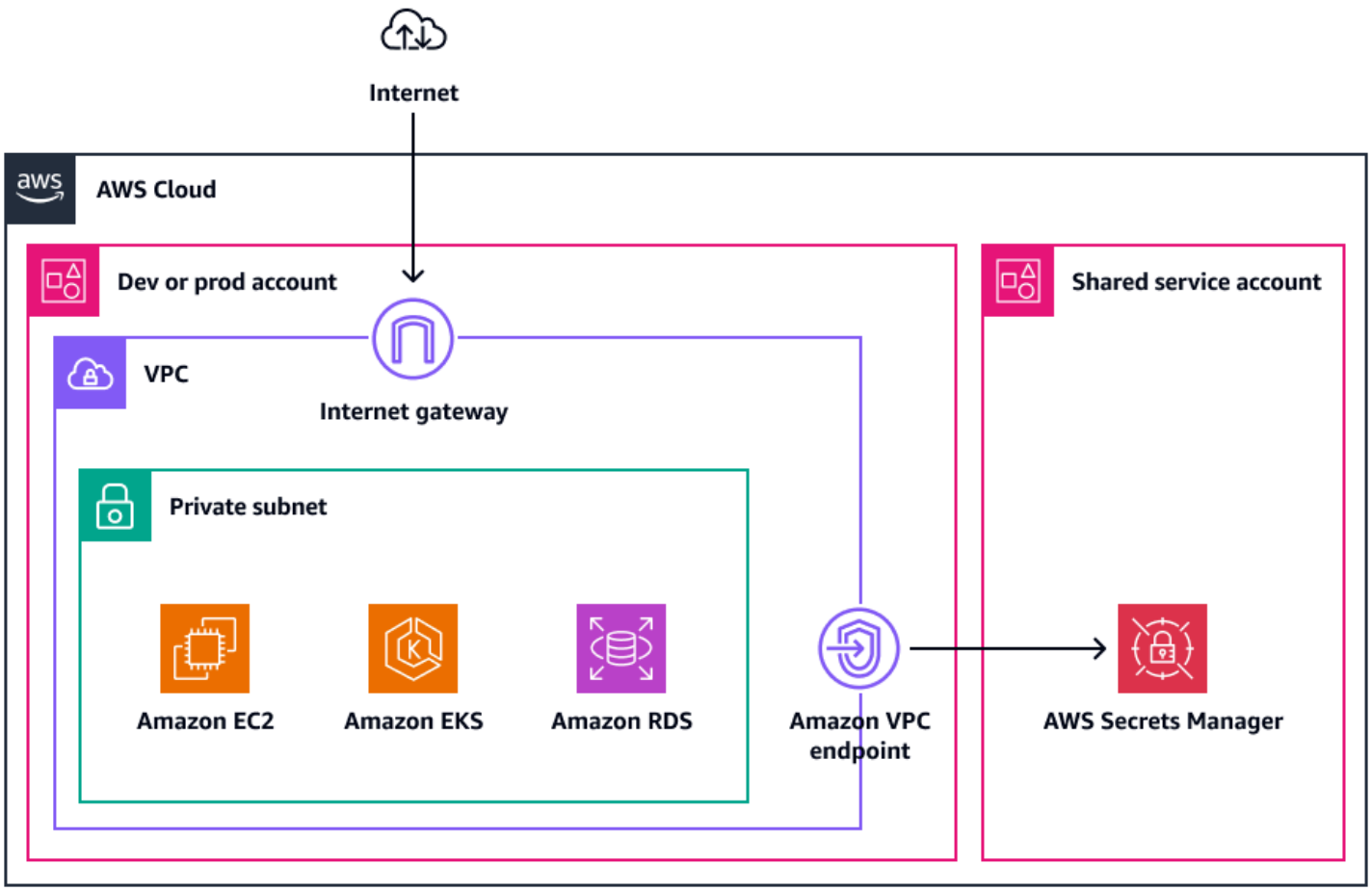
Utilizzo degli endpoint VPC per conservare i dati sensibili in reti note

I segreti non dovrebbero mai essere accessibili su Internet. AWS offre opzioni per mantenere la privacy durante l'instradamento del traffico attraverso percorsi di rete noti e privati.

Quando configuri il traffico tra client AWS Secrets Manager e applicazioni locali, puoi utilizzare uno dei seguenti approcci:

- Una AWS Site-to-Site VPN connessione VPN
- Una connessione AWS Direct Connect.

Se desideri proteggere il traffico tra Secrets Manager e client API con lo stesso Regione AWS, usalo AWS PrivateLinkper creare endpoint VPC di interfaccia. Usando questa opzione, mantieni tutto il traffico riservato all'interno della tua rete privata. Per ulteriori informazioni, consulta Utilizzo di un endpoint VPC AWS Secrets Manager.



Risorse

AWS documentazione

- Ruota i segreti AWS Secrets Manager
- Ruota immediatamente un segreto AWS Secrets Manager
- AWS Secrets Manager modelli di funzioni di rotazione
- Protezione dei dati in AWS Secrets Manager
- Utilizzo di un AWS Secrets Manager endpoint VPC
- Politiche dei bucket S3 e politiche degli utenti
- Archivia e utilizza i segreti in modo sicuro (AWS Well-Architected Framework)
- Verifica e ruota periodicamente le credenziali (Well-Architected AWS Framework)

AWS post sul blog

- Come accedere ai segreti in modo trasversale Account AWS allegando politiche basate sulle risorse
- Come utilizzare AWS Secrets & Configuration Provider con il driver CSI Kubernetes Secrets Store
- Sfrutta AWS i negozi segreti di EKS Fargate con External Secrets Operator

Altre risorse

- Backend Amazon S3 (documentazione) Terraform

Collaboratori

Le seguenti persone hanno contribuito a questa guida.

Scrittura

- Chintamani Aphale, consulente principale, AWS
- Aarti Rajput, consulente principale, AWS
- T.V.R.L. Phani Kumar Dadi, consulente, AWS
- Pradip Kumar Pandey, consulente principale, AWS
- Mayuri Shinde DevOps , consulente, AWS
- Pratap Kumar Nanda, consulente principale, AWS

Revisione

- Janardhan Malyala, consulente senior per la sicurezza, AWS
- Anand Krishna Varanasi, consulente principale, AWS
- Sandeep Gawande, consulente capo senior, AWS
- Mike Zazon, principale architetto dell'infrastruttura cloud, AWS

Scrittura tecnica

- Lilly AbouHarb, scrittrice tecnica senior, AWS

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Blocco dei file di stato	Abbiamo aggiornato la sezione Protezione dei dati sensibili nel file di Terraform stato. L'uso di una tabella Amazon DynamoDB per mantenere bloccato il file di stato non è più supportato.	20 maggio 2025
Pubblicazione iniziale	—	5 gennaio 2024

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.