Mengamankan data sensitif dengan menggunakan AWS Secrets Manager dan HashiCorp Terraform

AWS Panduan Preskriptif

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

AWS Panduan Preskriptif: Mengamankan data sensitif dengan menggunakan AWS Secrets Manager dan HashiCorp Terraform

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Pengantar 1
Audiens yang dituju 1
Tujuan 1
Menggunakan Secrets Manager dan Terraform 3
AWS Secrets Manager 3
Terraform 3
Praktik terbaik 5
Menghasilkan kata sandi acak 5
Rahasia berputar 7
Membatasi akses 10
Menggunakan konvensi penamaan 11
Mengelola data sensitif 13
Melindungi data sensitif di Terraform berkas negara 13
Mengelola rahasia untuk Amazon EKS 14
Menyimpan data sensitif di jaringan yang dikenal 16
Sumber daya 18
AWS dokumentasi 18
AWS posting blog 18
Sumber daya lainnya 18
Kontributor 19
Mengotorisasi 19
Meninjau 19
Penulisan teknis 19
Riwayat dokumen 20
..... xxi

Mengamankan data sensitif dengan menggunakan AWS Secrets Manager dan HashiCorp Terraform

Amazon Web Services (kontributor)

Mei 2025 (riwayat dokumen)

Manajemen data sensitif, termasuk kredensi, string rahasia, dan kata sandi, adalah pilar manajemen infrastruktur dan pengembangan dan penyebaran aplikasi yang diakui. Untuk membantu melindungi organisasi Anda, terapkan praktik terbaik untuk mengelola data sensitif di cloud. Perlindungan data sensitif merupakan prasyarat untuk keamanan dan kepatuhan. AWS Secrets Managerdapat membantu mengamankan data sensitif di lingkungan Anda sebagai rahasia.

Panduan ini mengulas praktik terbaik untuk rahasia, seperti cara mendapatkan rahasia dari Secrets Manager dan cara menggunakannya AWS Lambda untuk memutar rahasia secara otomatis untuk data sensitif. Ini juga memberikan rekomendasi tentang cara mengelola dan mengatur rahasia dengan menggunakan nama hierarkis. Akhirnya, ini membantu Anda mengelola penggunaan dan akses ke rahasia, seperti sentralisasi, Terraform integrasi, dan pertimbangan jaringan.

HashiCorp Terraformtelah diadopsi secara luas sebagai solusi infrastruktur sebagai kode (IAc) di industri. Namun, Terraform menunjukkan data sensitif sebagai teks biasa dalam file statusnya. Panduan ini berisi praktik terbaik untuk digunakan Terraform untuk mengelola data sensitif dan untuk membuat dan menggunakan rahasia Secrets Manager.

Audiens yang dituju

Panduan ini ditujukan untuk organisasi yang ingin digunakan Terraform sebagai solusi IAc. Praktik terbaik dalam panduan ini dirancang untuk membantu arsitek database, tim infrastruktur, dan pengembang aplikasi. Keakraban dengan Terraform adalah prasyarat untuk panduan ini.

Tujuan

Berikut ini adalah hasil bisnis yang dapat Anda harapkan setelah menerapkan rekomendasi dalam panduan ini:

- Berinovasi lebih cepat dengan mengotomatiskan proses pengelolaan rahasia.

- Tingkatkan postur keamanan organisasi Anda di AWS Cloud.

Berikut ini adalah hasil teknis yang dapat Anda harapkan untuk dicapai setelah menerapkan rekomendasi dalam panduan ini:

- Gunakan Secrets Manager untuk membantu mencegah pemaparan data sensitif dalam file Terraform status.
- Memusatkan pengelolaan rahasia dan data sensitif untuk meningkatkan tata kelola dan mencapai kepatuhan.
- Menerapkan praktik terbaik keamanan dalam proses organisasi Anda untuk menerapkan infrastruktur cloud.

Menggunakan Secrets Manager dan Terraform

AWS Secrets Manager

AWS Secrets Manageradalah layanan untuk mengenkripsi, menyimpan, dan memutar kredensil dengan aman untuk database dan layanan lainnya. Ini membantu Anda mengganti kredensi hardcode dalam kode Anda, termasuk kata sandi, dengan panggilan API untuk mengambil rahasia secara terprogram. Di Secrets Manager, sebuah rahasia terdiri dari informasi kredensil (yang merupakan nilai rahasia) dan metadata-nya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi lebih lanjut, lihat Rahasia.

Secrets Manager menggunakan kunci data simetris Advanced Encryption Standard (AES) 256-bit untuk mengenkripsi nilai rahasia. Untuk informasi selengkapnya, lihat Enkripsi rahasia dan dekripsi di. AWS Secrets Manager

Anda dapat mengakses dan bekerja dengan Secrets Manager dengan menggunakan salah satu pendekatan berikut:

- Konsol Secrets Manager
- Alat baris perintah
- AWS SDKs
- HTTPS Query API, juga disebut Secrets Manager API
- AWS Secrets Manager titik akhir

Terraform

Terraformadalah alat IAc HashiCorp yang membantu Anda membuat dan mengelola sumber daya cloud dan lokal. Anda dapat menggunakan Terraform untuk menyebarkan sumber daya dan infrastruktur di. AWS Cloud

Terraformmenyimpan informasi tentang AWS infrastruktur terkelola Anda dan konfigurasinya. Informasi ini disebut negara. Secara default, status disimpan dalam file lokal bernama`Terraform.tfstate`. File ini dalam format JSON, dan Terraform mungkin menyimpan data sensitif dalam file status ini dalam teks biasa. Ini menimbulkan risiko terhadap data sensitif karena setiap pengguna dengan akses ke file negara dapat mengakses data sensitif.

Panduan ini memberikan praktik dan rekomendasi terbaik untuk membantu Anda melindungi data sensitif saat menggunakan Terraform untuk mengelola AWS sumber daya Anda.

Praktik terbaik untuk membuat, memutar, dan menggunakan rahasia

Bagian ini membahas praktik dan standar terbaik yang dapat Anda ikuti saat mengkodekan aplikasi dan infrastruktur untuk penerapan. Terraform Ini berisi yang berikut:

- Menghasilkan kata sandi acak dengan menggunakan AWS Secrets Manager — Hasilkan rahasia acak dengan kompleksitas tertentu, sesuai kebijakan kepatuhan organisasi Anda.
- Menggunakan AWS Lambda untuk memutar rahasia - AWS Lambda dapat dijadwalkan untuk berjalan secara otomatis untuk rotasi rahasia.
- Membatasi akses ke rahasia — Batasi siapa yang dapat mengakses rahasia dan bagaimana mereka mengaksesnya.
- Menggunakan konvensi penamaan hierarkis untuk rahasia — Tentukan dan gunakan konvensi penamaan yang membantu Anda mengelola rahasia dalam skala besar.

Menghasilkan kata sandi acak dengan menggunakan AWS Secrets Manager

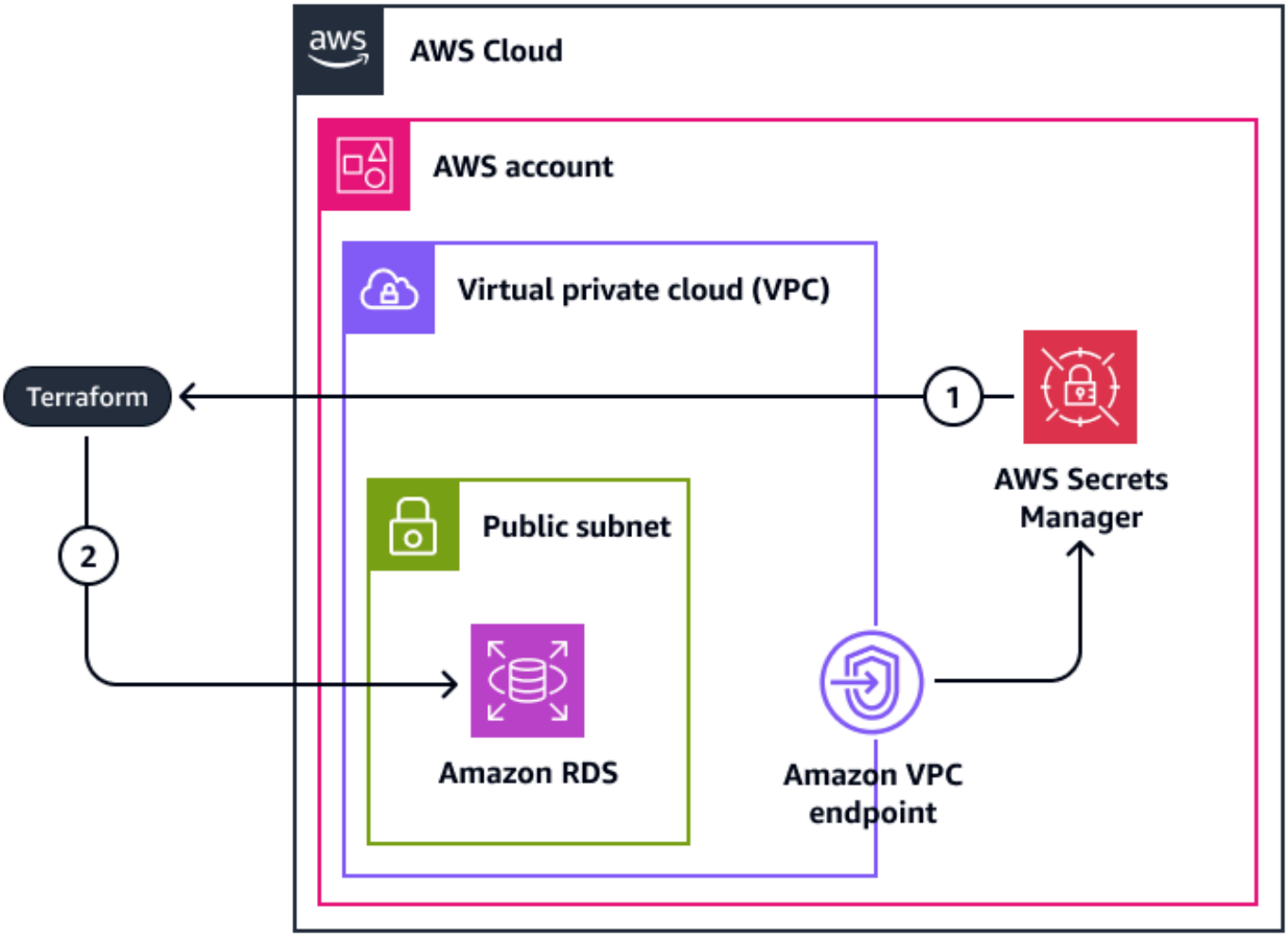
AWS Well-Architected Framework merekomendasikan agar Anda menyimpan dan menggunakan rahasia dengan aman. Anda dapat menggunakan AWS Secrets Manager API untuk menghasilkan kata sandi acak, dan Anda dapat menyesuaikan persyaratan kompleksitas kata sandi. `GetRandomPassword`Tindakan ini mendukung panjang string kata sandi antara 1 dan 4.096 karakter. Untuk informasi selengkapnya, lihat GetRandomPassword di dalam Referensi API AWS Secrets Manager . Kami menyarankan Anda menggunakan pendekatan ini untuk menghasilkan rahasia alih-alih mengizinkan pengguna untuk mendefinisikan rahasia secara manual.

Contoh kode berikut menunjukkan bagaimana Anda dapat menghasilkan kata sandi acak yang panjangnya 20 karakter dan yang mencakup angka, mengecualikan karakter tanda baca, dan mengecualikan spasi. Anda dapat mengubah contoh kode ini untuk memenuhi persyaratan keamanan kata sandi untuk organisasi Anda.

```
data "aws_secretsmanager_random_password" "test" {
password_length = 20
exclude_numbers = false
```

```
exclude_punctuation = true
include_space = false
}
```

Menggunakan generasi rahasia acak saat Anda menggunakan IAc membantu Anda melindungi data sensitif sejak awal, yang dikenal sebagai nol jam. Data sensitif tidak pernah diketahui siapa pun, langsung dari fase penerapan.



1. MelaluiTerraform, gunakan AWS Secrets Manager untuk menghasilkan rahasia kata sandi acak.
2. Terraformmenggunakan rahasia kata sandi acak ini, yang disimpan di AWS Secrets Manager, untuk mengakses database.

> **Important**
>
> Ketika Anda menggunakan Terraform sebagai sumber data, rahasia tidak disimpan dalam file negara. Tetapi setelah Anda menggunakan rahasia itu dalam database atau layanan apa pun, maka itu disimpan dalam file negara. Kami menyarankan Anda segera memutar rahasia atau membuat izin yang sangat ketat untuk mengakses file negara. Untuk informasi selengkapnya, lihat Melindungi data sensitif di Terraform berkas negara dalam panduan ini.

Penggunaan AWS Lambda untuk memutar rahasia

AWS Well-Architected Framework merekomendasikan agar Anda menyimpan dan menggunakan rahasia dengan aman. Praktik terbaik ini merekomendasikan agar Anda mengotomatiskan rotasi kredenal secara berkala. Rotasi adalah proses memperbarui rahasia secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensialnya. Banyak kerangka kerja dan peraturan kepatuhan juga mengharuskan Anda memutar rahasia.

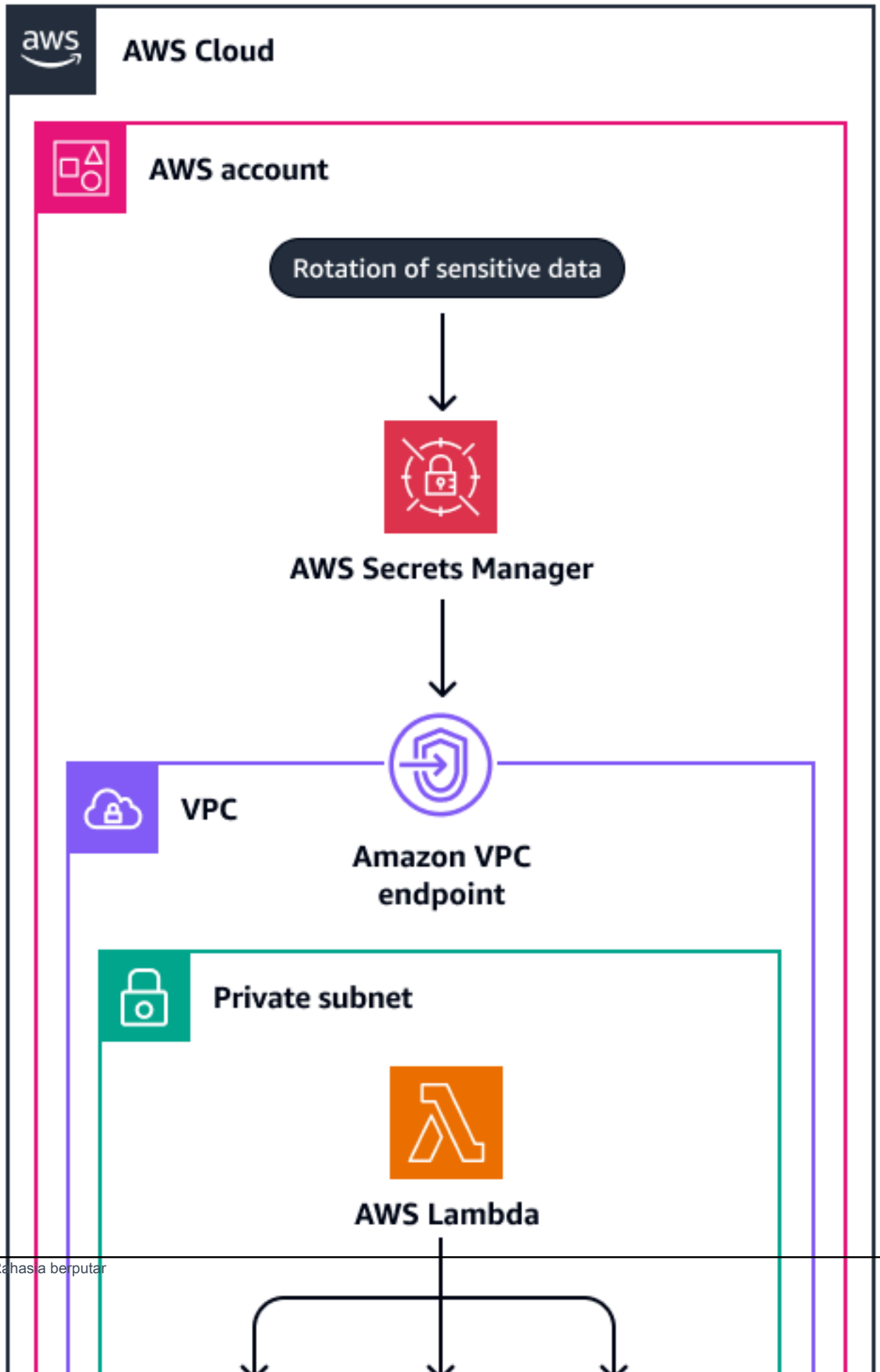
Untuk Terraform IAc, Anda dapat menggunakan AWS Secrets Manager dan AWS Lambdamembuat rotasi otomatis. Di Secrets Manager, Anda dapat mengatur rotasi otomatis untuk rahasia Anda. Ketika Secrets Manager memutar rahasia, ia memperbarui kredensialnya baik dalam rahasia maupun database atau layanan.

Untuk database, kami menyarankan Anda mengelola kredensi utama di Secrets Manager dan memutar rahasia secara berkala. Secrets Manager menyediakan template fungsi rotasi untuk Lambda untuk beberapa jenis kredensi database. Untuk informasi selengkapnya, lihat templat fungsi AWS Secrets Manager rotasi di dokumentasi Secrets Manager dan lihat contoh kode di GitHub. Berikut ini adalah contoh Terraform IAc yang dapat Anda gunakan untuk memutar rahasia atau data sensitif.

```
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
  }
}
```

Diagram arsitektur berikut menunjukkan bagaimana Anda dapat menggunakan Secrets Manager, titik akhir Amazon VPC, dan fungsi Lambda untuk memutar data sensitif dalam file. Akun AWS


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

Membatasi akses ke rahasia

Keamanan adalah prioritas utama di AWS. Dalam AWS Well-Architected Framework, Store and use secrets dengan aman praktik terbaik merekomendasikan agar Anda membatasi akses ke rahasia, log dan mengaudit akses ke rahasia, dan bahwa Anda mendapatkan akses melalui API alih-alih memberikan akses ke seseorang. Ini terkait dengan Menghasilkan kata sandi acak dengan menggunakan AWS Secrets Manager, yang sebelumnya dibahas dalam panduan ini.

Di masa lalu, kredensi yang digunakan untuk mengautentikasi ke database, API pihak ketiga, token, dan rahasia lainnya mungkin telah disematkan dalam kode sumber atau dalam file lingkungan. Di banyak organisasi, administrator database masih dapat mengakses semua kredensi pengguna database. Namun, itu lebih aman jika mereka tidak melakukannya. Kami menyarankan Anda:

- Batasi akses ke Secrets Manager melalui kebijakan AWS Identity and Access Management (IAM) yang dilampirkan ke peran AWS layanan atau peran IAM untuk pengguna.
- Memperluas akses terprogram ke rahasia di Secrets Manager hanya melalui panggilan API ke Secrets Manager.
- Hindari menggunakan rahasia hard-code.
- Jangan lewatkan nilai kredensi dalam variabel.
- Pastikan bahwa rahasia langsung digunakan oleh aplikasi atau infrastruktur, bukan oleh pengguna manusia.

Contoh kode berikut menunjukkan pendekatan yang dapat Anda gunakan untuk melindungi data sensitif saat membuat database. Saat Anda membuat database, Anda dapat meminta input pengguna sebagai kata sandi dan kemudian memutar kata sandi, atau Anda dapat menggunakan kata sandi acak dari awal. Membuat kata sandi acak dari awal dapat membantu Anda menangani dan mengelola kredensitas dengan lebih aman.

Cuplikan kode berikut membuat sumber daya database, dan `password = local.passwddata` mengambil kata sandi dari variabel lokal.

```
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
}
```

Cuplikan kode berikut membuat kata sandi acak atau menerima kata sandi yang disediakan pengguna. Terraformmenyimpan kata sandi ini sebagai variabel lokal sehingga Anda dapat menggunakan variabel kata sandi ini nanti dalam kode.

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

Menggunakan konvensi penamaan hierarkis untuk rahasia

Saat Anda menelan dan menyimpan rahasia AWS Secrets Manager, gunakan konvensi penamaan yang ditentukan. Misalnya, konvensi penamaan Anda mungkin`<client name>/<dev or prod>/<project>/<version>`. Memilih konvensi penamaan yang masuk akal untuk organisasi Anda. Gunakan struktur hierarkis yang berkembang dari informasi paling umum ke informasi yang paling spesifik.

Ini membantu Anda mengelola rahasia, terutama ketika akun terpusat digunakan untuk banyak rahasia di AWS lingkungan Anda. Konvensi penamaan hierarkis dapat membantu Anda dengan cara-cara berikut:

- Ini membantu Anda memisahkan akses ke rahasia yang berbeda. Misalnya, Anda dapat menentukan proyek atau lingkungan mana yang harus memiliki akses ke rahasia mana.
- Ini membantu Anda membuat kontrol akses berbutir halus ke rahasia berdasarkan nama mereka.

Contoh kode berikut menunjukkan contoh konvensi penamaan yang dapat membantu Anda mengelola rahasia dalam skala besar.

```
resource "aws_secretsmanager_secret" "initiatesecret" {
```

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

Mengelola data sensitif saat menggunakan Terraform

Data sensitif harus dirancang dengan baik. Seiring bertambahnya skala aplikasi dan infrastruktur, semakin penting untuk melacak dan menangani data sensitif dengan hati-hati. Anda dapat menggunakan pendekatan berikut untuk membantu melindungi data sensitif Akun AWS saat menerapkan Terraform IAc:

- Melindungi data sensitif di Terraform berkas negara— Anda dapat membantu melindungi data sensitif dari saat pertama kali dicerna AWS Secrets Manager. Misalnya, Anda dapat segera memutar rahasia untuk membantu menjaga kerahasiaannya.
- Mengakses dan mengelola rahasia untuk Amazon EKS— Kelola semua rahasia untuk Amazon Elastic Kubernetes Service (Amazon EKS) di Secrets Manager.
- Menggunakan titik akhir VPC untuk menyimpan data sensitif di jaringan yang dikenalLalu lintas untuk data sensitif tidak boleh meninggalkan jaringan pribadi. Ini membantu mencegah serangan dan eksfiltrasi data.

Melindungi data sensitif di Terraform berkas negara

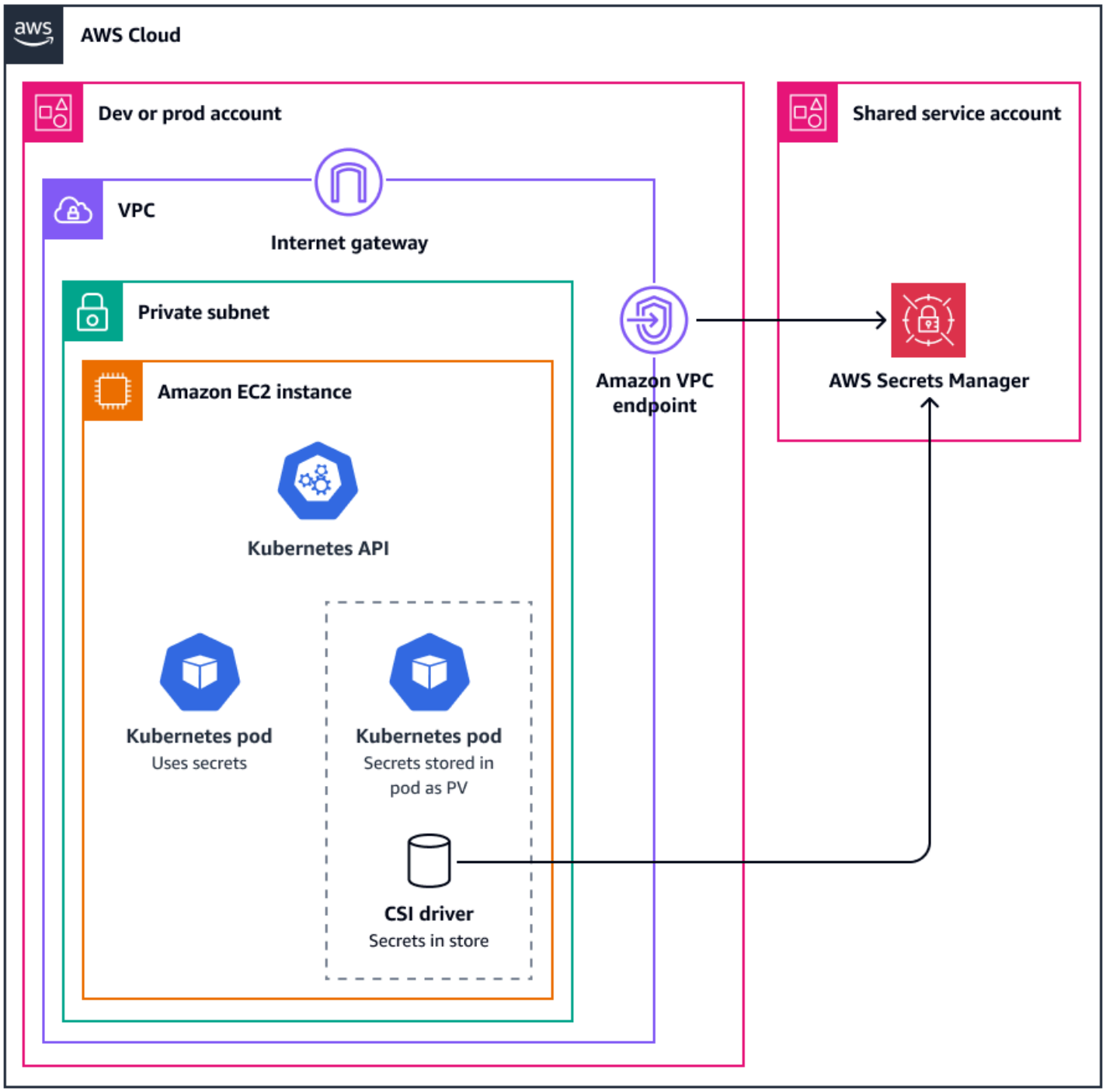
Bagian ini membahas pengaburan rahasia dan petunjuk untuk menangani data sensitif dalam file negara, yang disebut. Terraform `tfstate` Biasanya, ini adalah file teks biasa yang berisi data tentang Terraform penerapan, dan ini mencakup data sensitif dan tidak sensitif tentang infrastruktur yang digunakan. Data sensitif terlihat dalam teks biasa di file Terraform status. Untuk membantu melindungi data sensitif, lakukan hal berikut:

- Saat menelan rahasia, pilih untuk segera memutar rahasianya. Untuk informasi selengkapnya, lihat Memutar AWS Secrets Manager rahasia segera di dokumentasi Secrets Manager.
- Simpan file Terraform status di pusat Akun AWS tempat Anda mengoperasikan Secrets Manager. Simpan file di bucket Amazon Simple Storage Service (Amazon S3), dan konfigurasikan kebijakan yang membatasi akses ke file tersebut. Untuk informasi lebih lanjut, lihat Kebijakan bucket dan kebijakan pengguna dalam dokumentasi Amazon S3.
- Anda dapat mengunci Terraform negara untuk membantu mencegah korupsi. Untuk informasi selengkapnya tentang mengunci status dan melindungi file status, lihat backend Amazon S3 dalam dokumentasi. Terraform

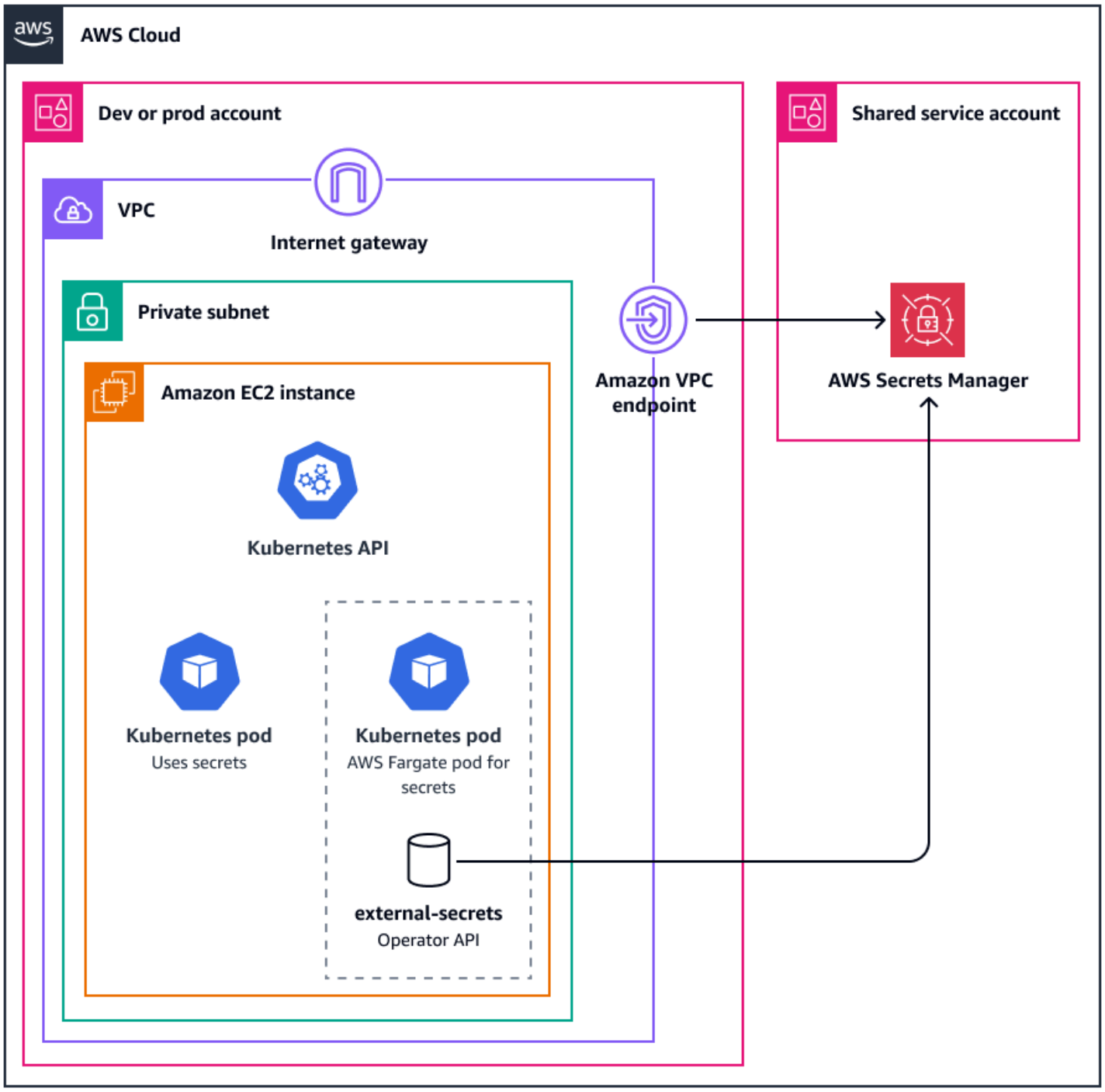
Mengakses dan mengelola rahasia untuk Amazon EKS

Amazon Elastic Kubernetes Service (Amazon EKS) membantu Anda menjalankan AWS Kubernetes tanpa perlu menginstal atau memelihara control plane atau node Kubernetes Anda sendiri. Amazon EKS menggunakan pengkodean Base64 (Wikipedia) untuk membantu melindungi data sensitif. Encoding dirancang untuk mencegah modifikasi data selama transit antar sistem, dan enkripsi dirancang untuk mencegah akses tidak sah ke data. Pengkodean Base64 tidak cukup untuk membantu melindungi data dari akses yang tidak sah. Gunakan AWS Secrets Manager untuk membantu melindungi data sensitif di Amazon EKS.

Gambar berikut menunjukkan Amazon EKS yang diterapkan pada instance Amazon Elastic Compute Cloud (Amazon EC2), yang bertindak sebagai node pekerja Kubernetes. Anda dapat menggunakan driver Container Storage Interface (CSI) untuk mengambil rahasia dari Secrets Manager. Untuk informasi selengkapnya, lihat Cara menggunakan AWS Secrets & Configuration Provider dengan driver CSI Kubernetes Secrets Store di Security Blog. AWS



Gambar berikut menunjukkan Amazon EKS dikerahkan. AWS Fargate Anda dapat menggunakan open source External Secrets Operator API (GitHub) untuk mengambil rahasia dari Secrets Manager. Untuk informasi selengkapnya, lihat Memanfaatkan toko AWS rahasia dari EKS Fargate dengan Operator Rahasia Eksternal di Blog AWS Kontainer.



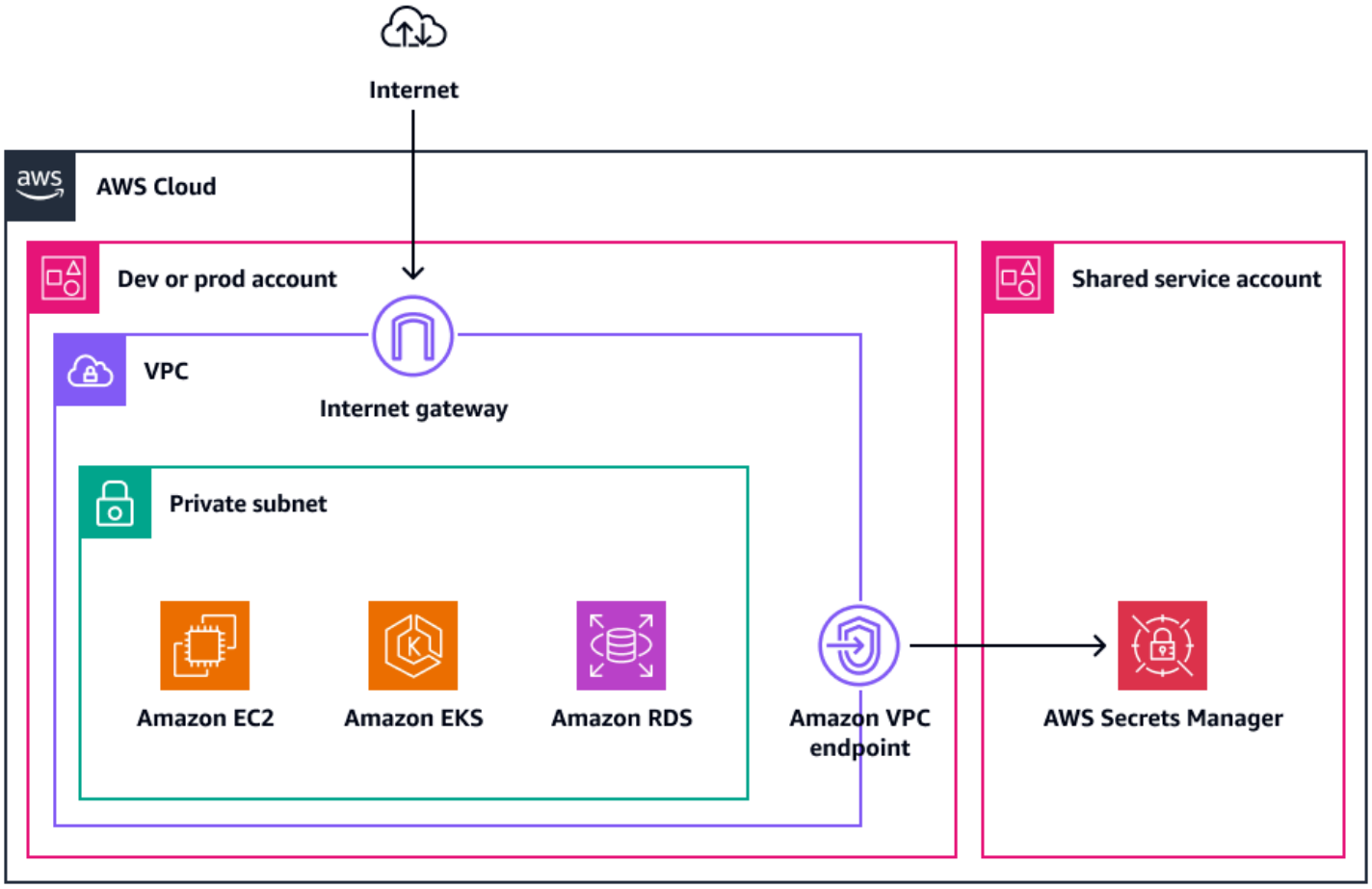
Menggunakan titik akhir VPC untuk menyimpan data sensitif di jaringan yang dikenal

Rahasia seharusnya tidak pernah dapat diakses melalui internet. AWS menawarkan opsi untuk menjaga privasi saat merutekan lalu lintas melalui rute jaringan yang dikenal dan pribadi.

Saat mengonfigurasi lalu lintas antara klien AWS Secrets Manager dan aplikasi lokal, Anda dapat menggunakan salah satu pendekatan berikut:

- Koneksi AWS Site-to-Site VPN VPN
- Koneksi AWS Direct Connect

Jika Anda ingin mengamankan lalu lintas antara Secrets Manager dan klien API dengan hal yang sama Wilayah AWS, gunakan AWS PrivateLinkuntuk membuat titik akhir VPC antarmuka. Dengan menggunakan opsi ini, Anda menyimpan semua lalu lintas untuk rahasia dalam jaringan pribadi Anda. Untuk informasi selengkapnya, silakan lihat Menggunakan sebuah titik akhir VPC AWS Secrets Manager.



Sumber daya

AWS dokumentasi

- Putar AWS Secrets Manager rahasia
- Putar AWS Secrets Manager rahasia segera
- AWS Secrets Manager templat fungsi rotasi
- Perlindungan data di AWS Secrets Manager
- Menggunakan titik akhir AWS Secrets Manager VPC
- Kebijakan bucket S3 dan kebijakan pengguna
- Simpan dan gunakan rahasia dengan aman (AWS Well-Architected Framework)
- Audit dan putar kredensional secara berkala (AWS Well-Architected Framework)

AWS posting blog

- Cara mengakses rahasia Akun AWS dengan melampirkan kebijakan berbasis sumber daya
- Cara menggunakan AWS Secrets & Configuration Provider dengan driver CSI Kubernetes Secrets Store
- Manfaatkan toko AWS rahasia dari EKS Fargate dengan Operator Rahasia Eksternal

Sumber daya lainnya

- Backend Amazon S3 (dokumentasi) Terraform

Kontributor

Individu berikut berkontribusi pada panduan ini.

Mengotorisasi

- Chintamani Aphale, Konsultan Utama, AWS
- Aarti Rajput, Konsultan Utama, AWS
- TVRL Phani Kumar Dadi, Konsultan, AWS
- Pradip Kumar Pandey, Konsultan Utama, AWS
- Mayuri Shinde, Konsultan, DevOps AWS
- Pratap Kumar Nanda, Konsultan Utama, AWS

Meninjau

- Janardhan Malyala, Konsultan Keamanan Senior, AWS
- Anand Krishna Varanasi, Konsultan Utama, AWS
- Sandeep Gawande, Konsultan Utama Senior, AWS
- Mike Zazon, Arsitek Infrastruktur Cloud Utama, AWS

Penulisan teknis

- Lilly AbouHarb, Penulis Teknis Senior, AWS

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Kunci file negara	Kami memperbarui Melindungi data sensitif di Terraform bagian file negara. Penggunaan tabel Amazon DynamoDB untuk menjaga file status terkunci tidak lagi didukung.	Mei 20, 2025
Publikasi awal	—	Januari 5, 2024

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.