aws

Panduan Pengguna

Layanan Terkelola Amazon untuk Prometheus

Layanan Terkelola Amazon untuk Prometheus: Panduan Pengguna

Table of Contents

Apa itu Layanan Terkelola Amazon untuk Prometheus?

Amazon Managed Service for Prometheus adalah layanan pemantauan tanpa server yang kompatibel dengan Prometheus untuk metrik kontainer yang memudahkan pemantauan lingkungan kontainer dengan aman dalam skala besar. Dengan Amazon Managed Service for Prometheus, Anda dapat menggunakan model data Prometheus sumber terbuka dan bahasa kueri yang sama yang Anda gunakan saat ini untuk memantau kinerja beban kerja kontainer Anda, dan juga menikmati peningkatan skalabilitas, ketersediaan, dan keamanan tanpa harus mengelola infrastruktur yang mendasarinya.

Amazon Managed Service untuk Prometheus secara otomatis menskalakan konsumsi, penyimpanan, dan kueri metrik operasional saat beban kerja meningkat dan turun. Ini terintegrasi dengan layanan AWS keamanan untuk memungkinkan akses cepat dan aman ke data.

Amazon Managed Service untuk Prometheus dirancang agar sangat tersedia menggunakan beberapa penerapan Availability Zone (Multi-AZ). Data yang dicerna ke dalam ruang kerja direplikasi di tiga Availability Zone di Region yang sama.

Amazon Managed Service for Prometheus bekerja dengan kluster kontainer yang berjalan di Amazon Elastic Kubernetes Service dan lingkungan Kubernetes yang dikelola sendiri.

Dengan Amazon Managed Service untuk Prometheus, Anda menggunakan model data Prometheus sumber terbuka yang sama dan bahasa kueri PromQL yang Anda gunakan dengan Prometheus. Tim teknik dapat menggunakan PromQL untuk memfilter, mengumpulkan, dan alarm pada metrik dan dengan cepat mendapatkan visibilitas kinerja tanpa perubahan kode apa pun. Amazon Managed Service untuk Prometheus menyediakan kemampuan kueri yang fleksibel tanpa biaya operasional dan kompleksitas.

Metrik yang dicerna ke dalam ruang kerja disimpan selama 150 hari, dan kemudian dihapus secara otomatis.

Wilayah yang didukung

Layanan Terkelola Amazon untuk Prometheus saat ini mendukung Wilayah berikut:

Nama Wilayah	Wilayah	Titik akhir	Protokol
US East (Ohio)	us–east-2	aps.us-east-2.amazonaws.com aps-workspaces.us-east-2.amazonaws.com	HTTPS HTTPS
US East (N. Virginia)	as-east-1	aps.us-east-1.amazonaws.com aps-workspaces.us-east-1.amazonaws.com	HTTPS HTTPS
US West (Oregon)	us-west-2	aps.us-west-2.amazonaws.com aps-workspaces.us-west-2.amazonaws.com	HTTPS HTTPS
Asia Pacific (Mumbai)	ap-south-1	aps.ap-south-1.amazonaws.com aps-workspaces.ap-south-1.amazonaws.com	HTTPS HTTPS
Asia Pacific (Seoul)	ap-northeast-2	aps.ap-northeast-2.amazonaws.com aps-workspaces.ap-northeast-2.amazonaws.com	HTTPS HTTPS
Asia Pacific (Singapore)	ap-southeast-1	aps.ap-southeast-1.amazonaws.com aps-workspaces.ap-southeast-1.amazonaws.com	HTTPS HTTPS
Asia Pacific (Sydney)	ap-southeast-2	aps.ap-southeast-2.amazonaws.com aps-workspaces.ap-southeast-2.amazonaws.com	HTTPS HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com aps-workspaces.ap-northeast-1.amazonaws.com	HTTPS HTTPS

Nama Wilayah	Wilayah	Titik akhir	Protokol
Europe (Frankfurt)	eu-central-1	aps.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.amazonaws.com	HTTPS
Europe (Ireland)	eu-west-1	aps.eu-west-1.amazonaws.com	HTTPS
		aps-workspaces.eu-west-1.amazonaws.com	HTTPS
Europe (London)	eu-west-2	aps.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.amazonaws.com	HTTPS
Europe (Paris)	eu-west-3	aps.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	aps.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	aps.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.amazonaws.com	HTTPS

Harga

Anda dikenakan biaya untuk konsumsi dan penyimpanan metrik. Biaya penyimpanan didasarkan pada ukuran terkompresi sampel metrik dan metadata. Untuk informasi selengkapnya, lihat Layanan Terkelola Amazon untuk Harga Prometheus.

Anda dapat menggunakan Cost Explorer dan AWS Cost and Usage Reports untuk memantau tagihan Anda. Untuk informasi selengkapnya, lihat Menjelajahi data Anda menggunakan Cost Explorer dan Apa itu Laporan AWS Biaya dan Penggunaan.

Dukungan Premium

Jika Anda berlangganan ke tingkat paket dukungan AWS premium mana pun, dukungan premium Anda berlaku untuk Layanan Terkelola Amazon untuk Prometheus.

Memulai

Bagian ini menjelaskan cara membuat Amazon Managed Service untuk ruang kerja Prometheus dengan cepat, mengatur konsumsi metrik Prometheus ke ruang kerja tersebut, dan menanyakan metrik tersebut.

Ini juga mencakup informasi tentang pengaturanAkun AWS, jika Anda baruAWS.

Topik

- Mengatur
- Buat ruang kerja
- Menelan metrik Prometheus ke ruang kerja
- Kueri metrik Prometheus Anda

Mengatur

Selesaikan tugas di bagian ini untuk mengatur AWS untuk pertama kalinya. Jika Anda sudah memiliki AWS akun, lompat ke depanBuat ruang kerja.

Ketika Anda mendaftarAWS, AWS akun Anda secara otomatis memiliki akses ke semua layanan diAWS, termasuk Amazon Managed Service untuk Prometheus. Namun, Anda hanya dikenakan biaya untuk layanan yang Anda gunakan.

Topik

- Daftar Akun AWS
- Membuat pengguna administratif

Daftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar Akun AWS, Pengguna root akun AWS dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Membuat pengguna administratif

Setelah Anda mendaftarAkun AWS, amankanPengguna root akun AWS, aktifkanAWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke AWS Management Console sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Pada halaman berikutnya, masukkan kata sandi Anda.

   Untuk bantuan masuk menggunakan pengguna root, lihat Masuk sebagai pengguna root dalam Panduan Pengguna AWS Sign-In.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda (konsol) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk petunjuk, lihat Mengaktifkan AWS IAM Identity Center di Panduan AWS IAM Identity Center Pengguna.
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna administratif.

Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal akses AWS dalam Panduan Pengguna AWS Sign-In.

Buat ruang kerja

Ruang kerja adalah ruang logis yang didedikasikan untuk penyimpanan dan kueri metrik Prometheus. Ruang kerja mendukung kontrol akses berbutir halus untuk mengotorisasi pengelolaannya seperti memperbarui, membuat daftar, mendeskripsikan, dan menghapus, serta penyerapan dan kueri metrik. Anda dapat memiliki satu atau lebih ruang kerja di setiap Wilayah di akun Anda.

Untuk menyiapkan ruang kerja, ikuti langkah-langkah ini.

> **Note**
>
> Untuk informasi lebih rinci tentang membuat ruang kerja, lihatBuat ruang kerja.

Untuk membuat Amazon Managed Service untuk ruang kerja Prometheus

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di https://console.aws.amazon.com/prometheus/.
2. Untuk alias Workspace, masukkan alias untuk ruang kerja baru.

   Alias ruang kerja adalah nama ramah yang membantu Anda mengidentifikasi ruang kerja Anda. Mereka tidak harus unik. Dua ruang kerja dapat memiliki alias yang sama, tetapi semua ruang kerja akan memiliki ID ruang kerja yang unik, yang dihasilkan oleh Amazon Managed Service untuk Prometheus.
3. (Opsional) Untuk menambahkan tag ke namespace, pilih Tambahkan tag baru.

Kemudian, untuk Kunci, masukkan nama untuk tag tersebut. Anda dapat menambahkan sebuah nilai opsional untuk tag di Nilai.

Untuk menambahkan tag lainnya, silakan pilih Tambahkan tag baru lagi.

4. Pilih Buat ruang kerja.

   Halaman detail ruang kerja muncul. Ini menampilkan informasi termasuk status, ARN, ID ruang kerja, dan URL titik akhir untuk ruang kerja ini untuk penulisan dan kueri jarak jauh.

   Awalnya, statusnya mungkin CREATING. Tunggu hingga statusnya AKTIF sebelum Anda melanjutkan untuk mengatur konsumsi metrik Anda.

   Buat catatan URL yang ditampilkan untuk Endpoint - URL tulis jarak jauh dan Endpoint - URL kueri. Anda akan membutuhkannya saat mengonfigurasi server Prometheus Anda untuk menulis metrik jarak jauh ke ruang kerja ini dan saat Anda menanyakan metrik tersebut.

Menelan metrik Prometheus ke ruang kerja

Salah satu cara untuk mencerna metrik adalah dengan menggunakan agen Prometheus mandiri (instance Prometheus yang berjalan dalam mode agen) untuk mengikis metrik dari cluster Anda dan meneruskannya ke Layanan Terkelola Amazon untuk Prometheus untuk penyimpanan dan pemantauan. Bagian ini menjelaskan cara mengatur konsumsi metrik ke dalam Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus dari Amazon EKS dengan menyiapkan instance baru agen Prometheus menggunakan Helm.

Untuk informasi tentang cara lain untuk memasukkan data ke Layanan Terkelola Amazon untuk Prometheus, termasuk cara mengamankan metrik dan membuat metrik ketersediaan tinggi, lihat. Konsumsi metrik Prometheus ke ruang kerja Anda

> **Note**
>
> Metrik yang dicerna ke dalam ruang kerja disimpan selama 150 hari, dan kemudian dihapus secara otomatis.

Petunjuk di bagian ini membuat Anda siap dan menjalankan Layanan Terkelola Amazon untuk Prometheus dengan cepat. Anda menyiapkan server Prometheus baru di kluster Amazon EKS, dan

server baru menggunakan konfigurasi default untuk bertindak sebagai agen untuk mengirim metrik ke Amazon Managed Service untuk Prometheus. Metode ini memiliki prasyarat berikut:

- Anda harus memiliki kluster Amazon EKS tempat server Prometheus baru akan mengumpulkan metrik.
- Anda harus menggunakan Helm CLI 3.0 atau yang lebih baru
- Anda harus menggunakan komputer Linux atau macOS untuk melakukan langkah-langkah di bagian berikut.

Langkah 1: Tambahkan repositori bagan Helm baru

Untuk menambahkan repositori bagan Helm baru, masukkan perintah berikut. Untuk informasi selengkapnya tentang perintah ini, lihat Helm Repo.

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo add kube-state-metrics https://kubernetes.github.io/kube-state-metrics
helm repo update
```

Langkah 2: Buat namespace Prometheus

Masukkan perintah berikut untuk membuat namespace Prometheus untuk server Prometheus dan komponen pemantauan lainnya. Ganti *prometheus-agent-namespace*dengan nama yang Anda inginkan untuk namespace ini.

```
kubectl create namespace prometheus-agent-namespace
```

Langkah 3: Siapkan peran IAM untuk akun layanan

Untuk metode konsumsi ini, Anda perlu menggunakan peran IAM untuk akun layanan di klaster Amazon EKS tempat agen Prometheus berjalan.

Dengan peran IAM untuk akun layanan, Anda dapat mengaitkan peran IAM dengan akun layanan Kubernetes. Akun layanan ini kemudian dapat menyediakan izin AWS ke kontainer-kontainer di setiap pod yang menggunakan akun layanan tersebut. Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan.

Jika Anda belum mengatur peran ini, ikuti instruksi di Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS untuk mengatur peran. Instruksi di bagian itu memerlukan

penggunaaneksctl. Untuk informasi selengkapnya, lihat Memulai dengan Amazon Elastic Kubernetes Service —. eksctl

> **Note**
>
> Saat Anda tidak menggunakan EKS atau AWS dan hanya menggunakan kunci akses dan kunci rahasia untuk mengakses Layanan Terkelola Amazon untuk Prometheus, Anda tidak dapat menggunakan SigV4 berbasis. EKS-IAM-ROLE

Langkah 4: Siapkan server baru dan mulai menelan metrik

Untuk menginstal agen Prometheus baru dan mengirim metrik ke Layanan Terkelola Amazon untuk ruang kerja Prometheus, ikuti langkah-langkah berikut.

Untuk menginstal agen Prometheus baru dan mengirim metrik ke Layanan Terkelola Amazon untuk ruang kerja Prometheus

1. Gunakan editor teks untuk membuat file bernama my_prometheus_values_yaml dengan konten berikut.

   - Ganti *IAM_PROXY_PROMETHEUS_ROLE_ARN* dengan ARN yang Anda buat. amp-iamproxy-ingest-roleMenyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS
   - Ganti *WORKSPACE_ID dengan ID* Amazon Managed Service untuk ruang kerja Prometheus.
   - Ganti *WILAYAH* dengan Wilayah Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

```
## The following is a set of default values for prometheus server helm chart which
 enable remoteWrite to AMP
## For the rest of prometheus helm chart values see: https://github.com/prometheus-
community/helm-charts/blob/main/charts/prometheus/values.yaml
##
serviceAccounts:
  server:
    name: amp-iamproxy-ingest-service-account
    annotations:
      eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}
server:
  remoteWrite:
```

```
    - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
      sigv4:
        region: ${REGION}
      queue_config:
        max_samples_per_send: 1000
        max_shards: 200
        capacity: 2500
```

2. Masukkan perintah berikut untuk membuat server Prometheus.

   - Ganti *prometheus-chart-name* dengan nama rilis Prometheus Anda.
   - Ganti *prometheus-agent-namespace* dengan nama namespace Prometheus Anda.

```
helm install prometheus-chart-name prometheus-community/prometheus -n prometheus-
agent-namespace \
-f my_prometheus_values_yaml
```

Kueri metrik Prometheus Anda

Sekarang metrik sedang dicerna ke ruang kerja, Anda dapat menanyakannya. Cara umum untuk menanyakan metrik Anda adalah dengan menggunakan layanan seperti Grafana untuk menanyakan metrik. Di bagian ini, Anda akan mempelajari cara menggunakan Grafana Terkelola Amazon untuk menanyakan metrik dari Amazon Managed Service untuk Prometheus.

> **Note**
>
> Untuk mempelajari cara lain untuk menanyakan metrik Layanan Terkelola Amazon untuk Prometheus, atau gunakan Layanan Terkelola Amazon untuk API Prometheus, lihat. Kueri metrik Prometheus Anda

Anda melakukan kueri Anda menggunakan bahasa kueri Prometheus standar, PromQL. Untuk informasi selengkapnya tentang PromQL dan sintaksnya, lihat Meminta Prometheus dalam dokumentasi Prometheus.

Grafana Terkelola Amazon adalah layanan yang dikelola sepenuhnya untuk Grafana open-source yang menyederhanakan koneksi ke sumber terbuka, ISV pihak ketiga, AWS dan layanan untuk memvisualisasikan dan menganalisis sumber data Anda dalam skala besar.

Layanan Terkelola Amazon untuk Prometheus mendukung penggunaan Grafana Terkelola Amazon untuk menanyakan metrik di ruang kerja. Di konsol Grafana Terkelola Amazon, Anda dapat menambahkan Layanan Terkelola Amazon untuk ruang kerja Prometheus sebagai sumber data dengan menemukan Layanan Terkelola Amazon untuk akun Prometheus yang ada. Grafana yang Dikelola Amazon mengelola konfigurasi kredensyal otentikasi yang diperlukan untuk mengakses Layanan Terkelola Amazon untuk Prometheus. Untuk petunjuk mendetail tentang cara membuat sambungan ke Layanan Terkelola Amazon untuk Prometheus dari Grafana yang Dikelola Amazon, lihat petunjuk di Panduan Pengguna Grafana Terkelola Amazon.

Anda juga dapat melihat peringatan Layanan Terkelola Amazon untuk Prometheus di Grafana Terkelola Amazon. Untuk petunjuk mengatur integrasi dengan peringatan, lihatMengintegrasikan peringatan dengan Grafana Terkelola Amazon atau Grafana open source.

> **Note**
>
> Jika Anda telah mengonfigurasi ruang kerja Grafana Terkelola Amazon untuk menggunakan VPC Pribadi, Anda harus menghubungkan Layanan Terkelola Amazon untuk ruang kerja Prometheus ke VPC yang sama. Untuk informasi selengkapnya, lihat Menghubungkan ke Grafana yang Dikelola Amazon dalam VPC pribadi.

Mengelola ruang kerja

Ruang kerja adalah ruang logis yang didedikasikan untuk penyimpanan dan kueri metrik Prometheus. Ruang kerja mendukung kontrol akses berbutir halus untuk mengotorisasi pengelolaannya seperti pembaruan, daftar, deskripsi, dan penghapusan, serta konsumsi dan kueri metrik. Anda dapat memiliki satu atau lebih ruang kerja di setiap Wilayah di akun Anda.

Gunakan prosedur di bagian ini untuk membuat dan mengelola Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Topik

Buat ruang kerja

Ikuti langkah-langkah ini untuk membuat Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Untuk membuat ruang kerja menggunakan AWS CLI

1. Masukkan perintah berikut untuk membuat ruang kerja. Contoh ini membuat ruang kerja bernama`my-first-workspace`, tetapi Anda dapat menggunakan alias yang berbeda (atau tidak ada) jika Anda mau. Alias ruang kerja adalah nama ramah yang membantu Anda mengidentifikasi ruang kerja Anda. Mereka tidak harus unik. Dua ruang kerja dapat memiliki alias yang sama, tetapi semua ruang kerja memiliki ID ruang kerja unik, yang dihasilkan oleh Amazon Managed Service untuk Prometheus.

   (Opsional) Untuk menggunakan kunci KMS Anda sendiri untuk mengenkripsi data yang disimpan di ruang kerja Anda, Anda dapat menyertakan `kmsKeyArn` parameter dengan AWS KMS kunci yang akan digunakan. Meskipun Layanan Terkelola Amazon untuk Prometheus tidak membebankan biaya kepada Anda untuk menggunakan kunci yang dikelola pelanggan, mungkin ada biaya yang terkait dengan kunci dari. AWS Key Management Service Untuk informasi selengkapnya tentang enkripsi data Amazon Managed Service for Prometheus di ruang kerja,

atau cara membuat, mengelola, dan menggunakan kunci terkelola pelanggan Anda sendiri, lihat. Enkripsi diam

Parameter dalam tanda kurung (`[]`) bersifat opsional, jangan sertakan tanda kurung dalam perintah Anda.

```
aws amp create-workspace [--alias my-first-workspace] [--kmsKeyArn arn:aws:aps:us-west-2:111122223333:workspace/ws-sample-1234-abcd-56ef-7890abcd12ef]  [--tags Status=Secret,Team=My-Team]
```

Perintah ini mengembalikan data berikut:

- `workspaceId`adalah ID unik untuk ruang kerja ini. Catat ID ini.
- `arn`adalah ARN untuk ruang kerja ini.
- `status`adalah status ruang kerja saat ini. Segera setelah Anda membuat ruang kerja, ini mungkin akan terjadi`CREATING`.
- `kmsKeyArn`adalah kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi data ruang kerja, jika diberikan.

> **Note**
>
> Ruang kerja yang dibuat dengan kunci terkelola pelanggan tidak dapat menggunakan kolektor AWS terkelola untuk konsumsi.
> Pilih apakah akan menggunakan kunci yang dikelola pelanggan atau kunci AWS yang dimiliki dengan hati-hati. Ruang kerja yang dibuat dengan kunci yang dikelola pelanggan tidak dapat dikonversi untuk menggunakan kunci yang AWS dimiliki nanti (dan sebaliknya).

- `tags`daftar tag ruang kerja, jika ada.

2. Jika `create-workspace` perintah Anda mengembalikan status`CREATING`, Anda kemudian dapat memasukkan perintah berikut untuk menentukan kapan ruang kerja siap. Ganti *`my-workspace-id`*dengan nilai yang dikembalikan `create-workspace` perintah`workspaceId`.

```
aws amp describe-workspace --workspace-id my-workspace-id
```

Ketika `describe-workspace` perintah kembali `ACTIVE` untuk`status`, ruang kerja siap digunakan.

Untuk membuat ruang kerja menggunakan Amazon Managed Service untuk konsol Prometheus

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di https://console.aws.amazon.com/prometheus/.
2. Pilih Buat.
3. Untuk alias Workspace, masukkan alias untuk ruang kerja baru.

   Alias ruang kerja adalah nama ramah yang membantu Anda mengidentifikasi ruang kerja Anda. Mereka tidak harus unik. Dua ruang kerja dapat memiliki alias yang sama, tetapi semua ruang kerja memiliki ID ruang kerja unik, yang dihasilkan oleh Amazon Managed Service untuk Prometheus.
4. (Opsional) Untuk menggunakan kunci KMS Anda sendiri untuk mengenkripsi data yang disimpan di ruang kerja Anda, Anda dapat memilih Sesuaikan pengaturan enkripsi, dan memilih AWS KMS kunci yang akan digunakan (atau membuat yang baru). Anda dapat memilih kunci di akun Anda dari daftar drop-down, atau masukkan ARN untuk kunci apa pun yang dapat Anda akses. Meskipun Layanan Terkelola Amazon untuk Prometheus tidak membebankan biaya kepada Anda untuk menggunakan kunci yang dikelola pelanggan, mungkin ada biaya yang terkait dengan kunci dari. AWS Key Management Service

   Untuk informasi selengkapnya tentang enkripsi data Amazon Managed Service for Prometheus di ruang kerja, atau cara membuat, mengelola, dan menggunakan kunci terkelola pelanggan Anda sendiri, lihat. Enkripsi diam

   > **Note**
   >
   > Ruang kerja yang dibuat dengan kunci terkelola pelanggan tidak dapat menggunakan kolektor AWS terkelola untuk konsumsi.
   > Pilih apakah akan menggunakan kunci yang dikelola pelanggan atau kunci AWS yang dimiliki dengan hati-hati. Ruang kerja yang dibuat dengan kunci yang dikelola pelanggan tidak dapat dikonversi untuk menggunakan kunci yang AWS dimiliki nanti (dan sebaliknya).
5. (Opsional) Untuk menambahkan satu atau beberapa tag ke ruang kerja, pilih Tambahkan tag baru. Kemudian, di Key, masukkan nama untuk tag. Anda dapat menambahkan sebuah nilai opsional untuk tag di Nilai.

   Untuk menambahkan tag lainnya, silakan pilih Tambahkan tag baru lagi.

6. Pilih Buat ruang kerja.

   Halaman detail ruang kerja muncul. Ini menampilkan informasi termasuk status, ARN, ID ruang kerja, dan URL titik akhir untuk ruang kerja ini untuk penulisan dan kueri jarak jauh.

   Status mengembalikan CREATING sampai ruang kerja siap. Tunggu hingga statusnya AKTIF sebelum Anda melanjutkan untuk mengatur konsumsi metrik Anda.

   Catat URL yang ditampilkan untuk Endpoint - URL tulis jarak jauh dan Endpoint - URL kueri. Anda akan membutuhkannya saat mengonfigurasi server Prometheus Anda untuk menulis metrik jarak jauh ke ruang kerja ini dan saat Anda menanyakan metrik tersebut.

Untuk informasi tentang cara memasukkan metrik ke dalam ruang kerja, lihat. Menelan metrik Prometheus ke ruang kerja

Mengedit ruang kerja

Anda dapat mengedit ruang kerja untuk mengubah aliasnya. Untuk mengubah alias ruang kerja menggunakanAWS CLI, masukkan perintah berikut.

```
aws amp update-workspace-alias --workspace-id my-workspace-id --alias "new-alias"
```

Untuk mengedit ruang kerja menggunakan Amazon Managed Service untuk konsol Prometheus

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di https://console.aws.amazon.com/prometheus/.
2. Di sudut kiri atas halaman, pilih ikon menu dan kemudian pilih Semua ruang kerja.
3. Pilih ID ruang kerja ruang kerja yang ingin Anda edit, lalu pilih Edit.
4. Masukkan alias baru untuk ruang kerja dan kemudian pilih Simpan.

Temukan ARN ruang kerja Anda

Anda dapat menemukan ARN Layanan Terkelola Amazon untuk ruang kerja Prometheus dengan menggunakan konsol atau. AWS CLI

Untuk menemukan ARN ruang kerja Anda menggunakan Amazon Managed Service untuk konsol Prometheus

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di https://console.aws.amazon.com/prometheus/.
2. Di sudut kiri atas halaman, pilih ikon menu dan kemudian pilih Semua ruang kerja.
3. Pilih ID Workspace dari ruang kerja.

   Ruang kerja ARN ditampilkan di bawah ARN.

Untuk menggunakan AWS CLI untuk menemukan ARN ruang kerja Anda, masukkan perintah berikut.

```
aws amp describe-workspace --workspace-id my-workspace-id
```

Temukan nilai `arn` dalam hasil.

Hapus ruang kerja

Saat Anda menghapus ruang kerja, data yang telah tertelan ke dalamnya tidak segera dihapus. Ini akan dihapus secara permanen dalam waktu satu bulan.

Untuk menghapus ruang kerja menggunakanAWS CLI, masukkan perintah berikut.

```
aws amp delete-workspace --workspace-id my-workspace-id
```

Untuk menghapus ruang kerja menggunakan Amazon Managed Service untuk konsol Prometheus

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di https://console.aws.amazon.com/prometheus/.
2. Di sudut kiri atas halaman, pilih ikon menu dan kemudian pilih Semua ruang kerja.
3. Pilih ID ruang kerja ruang kerja yang ingin Anda hapus, lalu pilih Hapus.
4. Masukkan **`delete`** di kotak konfirmasi, dan pilih Hapus.

Konsumsi metrik Prometheus ke ruang kerja Anda

Bagian ini menjelaskan cara mengatur konsumsi metrik ke dalam ruang kerja Anda.

Ada dua metode untuk memasukkan metrik ke dalam Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

- Menggunakan kolektor AWS terkelola - Amazon Managed Service for Prometheus menyediakan scraper tanpa agen yang dikelola sepenuhnya untuk mengikis metrik secara otomatis dari cluster Amazon Elastic Kubernetes Service (Amazon EKS) Anda. Mengikis secara otomatis menarik metrik dari titik akhir yang kompatibel dengan Prometheus.
- Menggunakan kolektor yang dikelola pelanggan — Anda memiliki banyak pilihan untuk mengelola kolektor Anda sendiri. Dua kolektor yang paling umum digunakan adalah menginstal instance Prometheus Anda sendiri, berjalan dalam mode agen, atau menggunakan Distro untuk. AWS OpenTelemetry Keduanya dijelaskan secara rinci di bagian berikut.

  Kolektor mengirim metrik ke Amazon Managed Service untuk Prometheus menggunakan fungsionalitas tulis jarak jauh Prometheus. Anda dapat langsung mengirim metrik ke Amazon Managed Service untuk Prometheus dengan menggunakan Prometheus remote write di aplikasi Anda sendiri. Untuk detail selengkapnya tentang langsung menggunakan remote write, dan konfigurasi penulisan jarak jauh, lihat remote_write di dokumentasi Prometheus.

Topik

AWS kolektor terkelola

Kasus penggunaan umum untuk Amazon Managed Service untuk Prometheus adalah memantau klaster Kubernetes yang dikelola oleh Amazon Elastic Kubernetes Service (Amazon EKS). Cluster Kubernetes, dan banyak aplikasi yang berjalan di Amazon EKS, secara otomatis mengekspor metriknya untuk diakses oleh scraper yang kompatibel dengan Prometheus.

> **Note**
>
> Banyak teknologi dan aplikasi yang berjalan di lingkungan Kubernetes menyediakan metrik yang kompatibel dengan Prometheus. Untuk daftar eksportir yang terdokumentasi dengan baik, lihat Eksportir dan integrasi dalam dokumentasi Prometheus.

Amazon Managed Service untuk Prometheus menyediakan scraper, atau kolektor yang dikelola sepenuhnya tanpa agen, yang secara otomatis menemukan dan menarik metrik yang kompatibel dengan Prometheus. Anda tidak perlu mengelola, menginstal, menambal, atau memelihara agen atau pencakar. Layanan Terkelola Amazon untuk kolektor Prometheus menyediakan koleksi metrik yang andal, stabil, sangat tersedia, dan diskalakan secara otomatis untuk kluster Amazon EKS Anda. Layanan Terkelola Amazon untuk kolektor yang dikelola Prometheus bekerja dengan kluster Amazon EKS, termasuk EC2 dan Fargate.

Layanan Terkelola Amazon untuk kolektor Prometheus membuat Antarmuka Jaringan Elastis (ENI) per subnet yang ditentukan saat membuat scraper. Kolektor mengikis metrik melalui ENI ini, dan menggunakannya `remote_write` untuk mendorong data ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus menggunakan titik akhir VPC. Data yang tergores tidak pernah bepergian di internet publik.

Topik berikut memberikan informasi selengkapnya tentang cara menggunakan Layanan Terkelola Amazon untuk kolektor Prometheus di klaster Amazon EKS Anda, dan tentang metrik yang dikumpulkan.

Topik

Menggunakan kolektor AWS terkelola

Untuk menggunakan Layanan Terkelola Amazon untuk kolektor Prometheus, Anda harus membuat scraper yang menemukan dan menarik metrik di cluster Amazon EKS Anda.

- Anda dapat membuat scraper sebagai bagian dari pembuatan cluster Amazon EKS Anda. Untuk informasi selengkapnya tentang membuat klaster Amazon EKS, termasuk membuat scraper, lihat Membuat klaster Amazon EKS di Panduan Pengguna Amazon EKS.

- Anda dapat membuat scraper Anda sendiri, secara terprogram dengan AWS API atau dengan menggunakan. AWS CLI

> **Note**
>
> Layanan Terkelola Amazon untuk ruang kerja Prometheus yang dibuat dengan kunci yang dikelola pelanggan tidak dapat menggunakan kolektor terkelola untuk konsumsi. AWS

Layanan Terkelola Amazon untuk kolektor Prometheus menggores metrik yang kompatibel dengan Prometheus. Untuk informasi selengkapnya tentang metrik yang kompatibel dengan Prometheus, lihat. Apa itu metrik yang kompatibel dengan Prometheus?

Topik berikut menjelaskan cara membuat, mengelola, dan mengonfigurasi pencakar.

Topik

Buat scraper

Layanan Dikelola Amazon untuk kolektor Prometheus terdiri dari scraper yang menemukan dan mengumpulkan metrik dari cluster Amazon EKS. Amazon Managed Service for Prometheus mengelola scraper untuk Anda, memberi Anda skalabilitas, keamanan, dan keandalan yang Anda butuhkan, tanpa harus mengelola instans, agen, atau pencakar apa pun sendiri.

Scraper dibuat secara otomatis untuk Anda saat Anda membuat cluster Amazon EKS melalui konsol Amazon EKS. Namun, dalam beberapa situasi Anda mungkin ingin membuat scraper sendiri. Misalnya, jika Anda ingin menambahkan kolektor AWS terkelola ke kluster Amazon EKS yang ada, atau jika Anda ingin mengubah konfigurasi kolektor yang ada.

Anda dapat membuat scraper menggunakan AWS API atau file. AWS CLI

Ada beberapa prasyarat untuk membuat scraper Anda sendiri:

- Anda harus memiliki kluster Amazon EKS yang dibuat.
- Cluster Amazon EKS Anda harus memiliki kontrol akses titik akhir cluster yang disetel untuk menyertakan akses pribadi. Ini dapat mencakup pribadi dan publik, tetapi harus mencakup pribadi.

Untuk membuat scraper menggunakan API AWS

Gunakan operasi `CreateScraper` API untuk membuat scraper dengan AWS API. Contoh berikut membuat scraper di `us-west-2` Wilayah. Anda perlu mengganti informasi cluster Akun AWS, ruang kerja, keamanan, dan Amazon EKS dengan ID Anda sendiri, dan menyediakan konfigurasi yang akan digunakan untuk scraper Anda.

> **Note**
>
> Anda harus menyertakan setidaknya dua subnet, setidaknya dalam dua zona ketersediaan.

`scrapeConfiguration`Ini adalah file YAMAL konfigurasi Prometheus yang dikodekan base64. Anda dapat mengunduh konfigurasi tujuan umum dengan operasi `GetDefaultScraperConfiguration` API. Bagian selanjutnya berisi rincian lebih lanjut tentang format file`scrapeConfiguration`.

```
POST /scrapers HTTP/1.1
Content-Length: 415
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: aws-cli/1.18.147 Python/2.7.18 Linux/5.4.58-37.125.amzn2int.x86_64
 botocore/1.18.6

{
    "alias": "myScraper",
    "destination":  {
        "ampConfiguration": {
            "workspaceArn": "arn:aws:aps:us-west-2:account-id:workspace/ws-workspace-
id"
        }
    },
    "source": {
        "eksConfiguration": {
```

```
            "clusterArn": "arn:aws:eks:us-west-2:account-id:cluster/cluster-name",
            "securityGroupIds": ["sg-security-group-id"],
            "subnetIds": ["subnet-subnet-id-1", "subnet-subnet-id-2"]
        }
    },
    "scrapeConfiguration": {
        "configurationBlob": <base64-encoded-blob>
    }
}
```

Untuk membuat scraper menggunakan AWS CLI

Gunakan `create-scraper` perintah untuk membuat scraper di `us-west-2` Region. Seperti pada contoh API, Anda harus mengganti informasi yang dibutuhkan dengan informasi dari akun Anda sendiri.

```
aws amp create-scraper \
  --source eksConfiguration="{clusterArn='arn:aws:eks:us-west-2:account-
id:cluster/cluster-name', securityGroupIds=['sg-security-group-
id'],subnetIds=['subnet-subnet-id-1', 'subnet-subnet-id-2']}" \
  --scrape-configuration configurationBlob=<base64-encoded-blob> \
  --destination ampConfiguration="{workspaceArn='arn:aws:aps:us-west-2:account-
id:workspace/ws-workspace-id'}"
```

Berikut ini adalah daftar lengkap operasi scraper yang dapat Anda gunakan dengan AWS API:

- Buat scraper dengan operasi CreateScraperAPI.
- Buat daftar scraper yang ada dengan operasi ListScrapersAPI.
- Hapus scraper dengan operasi DeleteScraperAPI.
- Dapatkan detail selengkapnya tentang scraper dengan operasi DescribeScraperAPI.
- Dapatkan konfigurasi tujuan umum untuk pencakar dengan operasi GetDefaultScraperConfigurationAPI.

> **Note**
>
> Cluster Amazon EKS yang Anda gores harus dikonfigurasi untuk memungkinkan Amazon Managed Service untuk Prometheus mengakses metrik. Topik berikutnya menjelaskan cara mengonfigurasi klaster Anda.

Mengonfigurasi klaster Amazon EKS Anda

Cluster Amazon EKS Anda harus dikonfigurasi untuk memungkinkan scraper mengakses metrik. Langkah-langkah berikut akan memungkinkan akses. Prosedur ini menggunakan `kubectl` dan AWS CLI. Untuk informasi tentang menginstal`kubectl`, lihat Menginstal kubectl di Panduan Pengguna Amazon EKS.

Untuk mengonfigurasi klaster Amazon EKS Anda untuk pengikisan metrik terkelola

1. Buat file, bernama`clusterrole-binding.yml`, dengan teks berikut:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aps-collector-role
rules:
  - apiGroups: [""]
    resources: ["nodes", "nodes/proxy", "nodes/metrics", "services", "endpoints",
 "pods", "ingresses", "configmaps"]
    verbs: ["describe", "get", "list", "watch"]
  - apiGroups: ["extensions", "networking.k8s.io"]
    resources: ["ingresses/status", "ingresses"]
    verbs: ["describe", "get", "list", "watch"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aps-collector-user-role-binding
subjects:
- kind: User
  name: aps-collector-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aps-collector-role
  apiGroup: rbac.authorization.k8s.io
```

2. Jalankan perintah berikut di cluster Anda:

```
kubectl apply -f clusterrole-binding.yml
```

Ini akan membuat pengikatan dan aturan peran cluster. Contoh ini digunakan `aps-collector-role` sebagai nama peran, dan `aps-collector-user` sebagai nama pengguna.

3. *Perintah berikut memberi Anda informasi tentang scraper dengan ID scraper-id.* Ini adalah scraper yang Anda buat menggunakan perintah di bagian sebelumnya.

```
aws amp describe-scraper --scraper-id scraper-id
```

4. Dari hasil`describe-scraper`, temukan `roleArn` .This akan memiliki format berikut:

```
arn:aws:iam::account-id:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper_unique-id
```

Amazon EKS membutuhkan format yang berbeda untuk ARN ini. Anda harus menyesuaikan format ARN yang dikembalikan untuk digunakan pada langkah berikutnya. Edit agar sesuai dengan format ini:

```
arn:aws:iam::account-id:role/AWSServiceRoleForAmazonPrometheusScraper_unique-id
```

Misalnya, ARN ini:

```
arn:aws:iam::111122223333:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper_1234abcd-56ef-7
```

Harus ditulis ulang sebagai:

```
arn:aws:iam::111122223333:role/
AWSServiceRoleForAmazonPrometheusScraper_1234abcd-56ef-7
```

5. Jalankan perintah berikut di cluster Anda, menggunakan modifikasi `roleArn` dari langkah sebelumnya, serta nama cluster dan wilayah Anda. :

```
eksctl create iamidentitymapping --cluster cluster-name --region region-id --
arn roleArn --username aps-collector-user
```

Ini memungkinkan scraper untuk mengakses cluster menggunakan peran dan pengguna yang Anda buat dalam `clusterrole-binding.yml` file.

Temukan dan hapus pencakar

Anda dapat menggunakan AWS API atau AWS CLI untuk membuat daftar pencakar di akun Anda atau untuk menghapusnya.

Untuk mencantumkan semua pencakar di akun Anda, gunakan operasi ListScrapersAPI.

Atau, dengan AWS CLI, hubungi:

```
aws amp list-scrapers
```

`ListScrapers`mengembalikan semua pencakar di akun Anda, misalnya:

```
{
    "scrapers": [
        {
            "scraperId": "s-1234abcd-56ef-7890-abcd-1234ef567890",
            "arn": "arn:aws:aps:us-west-2:123456789012:scraper/s-1234abcd-56ef-7890-
abcd-1234ef567890",
            "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
AWSServiceRoleForAmazonPrometheusScraper_1234abcd-2931",
            "status": {
                "statusCode": "DELETING"
            },
            "createdAt": "2023-10-12T15:22:19.014000-07:00",
            "lastModifiedAt": "2023-10-12T15:55:43.487000-07:00",
            "tags": {},
            "source": {
                "eksConfiguration": {
                    "clusterArn": "arn:aws:eks:us-west-2:123456789012:cluster/my-
cluster",
                    "securityGroupIds": [
                        "sg-1234abcd5678ef90"
                    ],
                    "subnetIds": [
                        "subnet-abcd1234ef567890",
                        "subnet-1234abcd5678ab90"
                    ]
                }
            },
            "destination": {
                "ampConfiguration": {
```

```
                    "workspaceArn": "arn:aws:aps:us-west-2:123456789012:workspace/
ws-1234abcd-5678-ef90-ab12-cdef3456a78"
                }
            }
        }
    ]
}
```

Untuk menghapus scraper, cari scraper yang ingin Anda hapus, menggunakan `ListScrapers` operasi, dan kemudian gunakan DeleteScraperoperasi untuk menghapusnya. `scraperId`

Atau, dengan AWS CLI, hubungi:

```
aws amp delete-scraper --scraper-id scraperId
```

Konfigurasi scraper

Anda dapat mengontrol bagaimana scraper Anda menemukan dan mengumpulkan metrik dengan konfigurasi scraper yang kompatibel dengan Prometheus. Misalnya, Anda dapat mengubah interval metrik yang dikirim ke ruang kerja. Anda juga dapat menggunakan pelabelan ulang untuk menulis ulang label metrik secara dinamis. Konfigurasi scraper adalah file YAMG yang merupakan bagian dari definisi scraper.

Untuk informasi lebih lanjut tentang format konfigurasi scraper, termasuk rincian rinci dari nilai yang mungkin, lihat Konfigurasi dalam dokumentasi Prometheus. Opsi konfigurasi global, dan `<scrape_config>` opsi menjelaskan opsi yang paling umum dibutuhkan.

Saat scraper baru dibuat, Anda menentukan konfigurasi dengan menyediakan file YAMG yang dikodekan base64 dalam panggilan API. Anda dapat mengunduh file konfigurasi tujuan umum dengan `GetDefaultScraperConfiguration` operasi di Amazon Managed Service for Prometheus API.

Untuk memodifikasi konfigurasi scraper, hapus scraper dan buat ulang dengan konfigurasi baru.

Contoh file konfigurasi

Berikut ini adalah contoh file konfigurasi YAMAL dengan interval scrape 30 detik.

```
global:
   scrape_interval: 30s
   external_labels:
     clusterArn: apiserver-test-2
```

```
scrape_configs:
  - job_name: pod_exporter
    kubernetes_sd_configs:
      - role: pod
  - job_name: cadvisor
    scheme: https
    authorization:
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    kubernetes_sd_configs:
      - role: node
    relabel_configs:
      - action: labelmap
        regex: __meta_kubernetes_node_label_(.+)
      - replacement: kubernetes.default.svc:443
        target_label: __address__
      - source_labels: [__meta_kubernetes_node_name]
        regex: (.+)
        target_label: __metrics_path__
        replacement: /api/v1/nodes/$1/proxy/metrics/cadvisor
  # apiserver metrics
  - scheme: https
    authorization:
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    job_name: kubernetes-apiservers
    kubernetes_sd_configs:
    - role: endpoints
    relabel_configs:
    - action: keep
      regex: default;kubernetes;https
      source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_service_name
      - __meta_kubernetes_endpoint_port_name
  # kube proxy metrics
  - job_name: kube-proxy
    honor_labels: true
    kubernetes_sd_configs:
    - role: pod
    relabel_configs:
    - action: keep
      source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_pod_name
      separator: '/'
```

```
      regex: 'kube-system/kube-proxy.+'
    - source_labels:
      - __address__
      action: replace
      target_label: __address__
      regex: (.+?)(\\:\\d+)?
      replacement: $1:10249
```

Ada dua batasan khusus untuk kolektor yang AWS dikelola:

- Interval mengikis - Konfigurasi scraper tidak dapat menentukan interval gesekan kurang dari 30 detik.
- Target — Target dalam `static_config` harus ditentukan sebagai alamat IP.

Memecahkan masalah konfigurasi scraper

Layanan Terkelola Amazon untuk kolektor Prometheus secara otomatis menemukan dan mengikis metrik. Tetapi bagaimana Anda bisa memecahkan masalah ketika Anda tidak melihat metrik yang Anda harapkan untuk dilihat di Layanan Terkelola Amazon untuk ruang kerja Prometheus?

upMetrik adalah alat yang bermanfaat. Untuk setiap titik akhir yang ditemukan oleh Amazon Managed Service untuk kolektor Prometheus, secara otomatis menjual metrik ini. Ada tiga status metrik ini yang dapat membantu Anda memecahkan masalah apa yang terjadi di dalam kolektor.

- uptidak ada — Jika tidak ada `up` metrik untuk titik akhir, maka itu berarti kolektor tidak dapat menemukan titik akhir.

  Jika Anda yakin bahwa titik akhir ada, Anda mungkin perlu menyesuaikan konfigurasi scrape. Penemuan ini `relabel_config` mungkin perlu disesuaikan, atau mungkin ada masalah dengan yang `role` digunakan untuk penemuan.
- upada, tetapi selalu 0 — Jika `up` ada, tetapi 0, maka kolektor dapat menemukan titik akhir, tetapi tidak dapat menemukan metrik yang kompatibel dengan Prometheus.

  Dalam hal ini, Anda dapat mencoba menggunakan `curl` perintah terhadap titik akhir secara langsung. Anda dapat memvalidasi bahwa Anda memiliki detail yang benar, misalnya, protokol (`http`atau`https`, titik akhir, atau port yang Anda gunakan. Anda juga dapat memeriksa apakah titik akhir merespons dengan respons yang valid, dan mengikuti `200` format Prometheus. Akhirnya, tubuh respons tidak bisa lebih besar dari ukuran maksimum yang diizinkan. (Untuk batasan kolektor AWS terkelola, lihat bagian berikut.)

- uphadir dan lebih besar dari 0 — Jika up ada, dan lebih besar dari 0, maka metrik sedang dikirim ke Amazon Managed Service untuk Prometheus.

  Validasi bahwa Anda mencari metrik yang benar di Amazon Managed Service untuk Prometheus (atau dasbor alternatif Anda, seperti Grafana yang Dikelola Amazon). Anda dapat menggunakan curl lagi untuk memeriksa data yang diharapkan di titik `/metrics` akhir Anda. Periksa juga apakah Anda belum melampaui batas lain, seperti jumlah titik akhir per scraper.

Keterbatasan scraper

Ada beberapa batasan untuk pencakar yang dikelola sepenuhnya yang disediakan oleh Amazon Managed Service untuk Prometheus.

- Wilayah — Cluster EKS Anda, scraper terkelola, dan Layanan Terkelola Amazon untuk ruang kerja Prometheus semuanya harus berada di Wilayah yang sama. AWS
- Akun — Kluster EKS Anda, scraper terkelola, dan Layanan Terkelola Amazon untuk ruang kerja Prometheus semuanya harus sama. Akun AWS
- Kolektor - Anda dapat memiliki maksimal 10 Layanan Dikelola Amazon untuk pencakar Prometheus per wilayah per akun.

  > **Note**
  >
  > Anda dapat meminta kenaikan batas ini dengan meminta kenaikan kuota.

- Respons metrik - Tubuh respons dari salah satu permintaan `/metrics` titik akhir tidak boleh lebih dari 50 megabyte (MB).
- Titik akhir per scraper - Scraper dapat mengikis maksimum 30.000 titik akhir. `/metrics`
- Interval mengikis - Konfigurasi scraper tidak dapat menentukan interval gesekan kurang dari 30 detik.

Apa itu metrik yang kompatibel dengan Prometheus?

Untuk mengikis metrik Prometheus dari aplikasi dan infrastruktur Anda untuk digunakan di Amazon Managed Service for Prometheus, metrik tersebut harus instrumen dan mengekspos metrik yang kompatibel dengan Prometheus dari titik akhir yang kompatibel dengan Prometheus. `/metrics` Anda dapat menerapkan metrik Anda sendiri, tetapi Anda tidak harus melakukannya. Kubernetes

(termasuk Amazon EKS) dan banyak pustaka dan layanan lainnya mengimplementasikan metrik ini secara langsung.

Saat metrik di Amazon EKS diekspor ke titik akhir yang kompatibel dengan Prometheus, metrik tersebut dapat dikikis secara otomatis oleh kolektor Layanan Terkelola Amazon untuk Prometheus.

Untuk informasi selengkapnya, lihat topik berikut:

- Untuk informasi selengkapnya tentang pustaka dan layanan yang ada yang mengekspor metrik sebagai metrik Prometheus, lihat Eksportir dan integrasi dalam dokumentasi Prometheus.
- Untuk informasi selengkapnya tentang mengekspor metrik yang kompatibel dengan Prometheus dari kode Anda sendiri, lihat Menulis eksportir di dokumentasi Prometheus.
- Untuk informasi selengkapnya tentang cara menyiapkan Layanan Terkelola Amazon untuk kolektor Prometheus untuk mengikis metrik dari kluster Amazon EKS Anda secara otomatis, lihat. Menggunakan kolektor AWS terkelola

Kolektor yang dikelola pelanggan

Bagian ini berisi informasi tentang menelan data dengan menyiapkan kolektor Anda sendiri yang mengirim metrik ke Amazon Managed Service untuk Prometheus menggunakan Prometheus remote write.

Saat Anda menggunakan kolektor Anda sendiri untuk mengirim metrik ke Amazon Managed Service untuk Prometheus, Anda bertanggung jawab untuk mengamankan metrik Anda dan memastikan bahwa proses konsumsi memenuhi kebutuhan ketersediaan Anda.

Sebagian besar kolektor yang dikelola pelanggan menggunakan salah satu alat berikut:

- AWSDistro for OpenTelemetry (ADOT) — ADOT adalah distribusi open source yang didukung penuh, aman, dan siap produksi OpenTelemetry yang menyediakan agen untuk mengumpulkan metrik. Anda dapat menggunakan ADOT untuk mengumpulkan metrik dan mengirimkannya ke Layanan Terkelola Amazon untuk ruang kerja Prometheus. Untuk informasi selengkapnya tentang Kolektor ADOT, lihat AWSDistro untuk. OpenTelemetry
- Agen Prometheus — Anda dapat mengatur instance Anda sendiri dari server Prometheus open source, berjalan sebagai agen, untuk mengumpulkan metrik dan meneruskannya ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

Topik berikut menjelaskan penggunaan kedua alat ini dan menyertakan informasi umum tentang pengaturan kolektor Anda sendiri.

Topik

Amankan konsumsi metrik Anda

Layanan Terkelola Amazon untuk Prometheus menyediakan cara untuk membantu Anda mengamankan konsumsi metrik Anda.

Menggunakan AWS PrivateLink dengan Amazon Managed Service untuk Prometheus

Lalu lintas jaringan untuk memasukkan metrik ke Amazon Managed Service untuk Prometheus dapat dilakukan melalui titik akhir internet publik, atau melalui titik akhir VPC. AWS PrivateLink Menggunakan AWS PrivateLink memastikan bahwa lalu lintas jaringan dari VPC Anda diamankan dalam AWS jaringan tanpa melalui internet publik. Untuk membuat titik akhir AWS PrivateLink VPC untuk Amazon Managed Service untuk Prometheus, lihat. Menggunakan Amazon Managed Service untuk Prometheus dengan titik akhir VPC antarmuka

Autentikasi dan otorisasi

AWSIdentity and Access Management (IAM) adalah layanan web yang membantu Anda mengontrol akses ke sumber daya dengan aman. AWS Anda menggunakan IAM untuk mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan sumber daya. Amazon Managed Service for Prometheus terintegrasi dengan IAM untuk membantu Anda menjaga keamanan data. Saat menyiapkan Amazon Managed Service untuk Prometheus, Anda perlu membuat beberapa peran IAM yang memungkinkannya menyerap metrik dari server Prometheus, dan yang memungkinkan server Grafana untuk menanyakan metrik yang disimpan di Amazon Managed Service untuk ruang kerja Prometheus. Untuk informasi selengkapnya tentang IAM, lihat Apa itu IAM?.

Fitur AWS keamanan lain yang dapat membantu Anda menyiapkan Amazon Managed Service untuk Prometheus adalah AWS proses penandatanganan Signature Version 4 (SigV4). AWS Signature

Versi 4 adalah proses untuk menambahkan informasi autentikasi ke permintaan AWS yang dikirim oleh HTTP. Demi keamanan, sebagian besar permintaan untuk AWS harus ditandatangani dengan kunci akses, yang terdiri dari ID kunci akses dan kunci akses rahasia. Kedua kunci ini umumnya disebut sebagai kredensial keamanan Anda. Untuk informasi selengkapnya tentang SiGv4, lihat proses penandatanganan Sigv4 Versi Tanda Tangan 4.

Menggunakan AWS Distro untuk OpenTelemetry sebagai kolektor

Topik berikut menjelaskan berbagai cara untuk mengatur AWS Distro OpenTelemetry sebagai kolektor untuk metrik Anda.

Topik

- Siapkan konsumsi metrik menggunakan AWS Distro untuk Open Telemetry di klaster Amazon Elastic Kubernetes Service
- Siapkan konsumsi metrik dari Amazon ECS menggunakan AWS Distro untuk Open Telemetry
- Mengatur konsumsi metrik dari instans Amazon EC2 menggunakan penulisan jarak jauh

Siapkan konsumsi metrik menggunakan AWS Distro untuk Open Telemetry di klaster Amazon Elastic Kubernetes Service

Bagian ini menjelaskan cara mengonfigurasi Kolektor AWS Distro for OpenTelemetry (ADOT) untuk mengikis dari aplikasi yang diinstrumentasi Prometheus, dan mengirim metrik ke Amazon Managed Service untuk Prometheus. Untuk informasi selengkapnya tentang Kolektor ADOT, lihat AWSDistro untuk. OpenTelemetry

Mengumpulkan metrik Prometheus dengan ADOT melibatkan OpenTelemetry tiga komponen: Penerima Prometheus, Eksportir Tulis Jarak Jauh Prometheus, dan Ekstensi Otentikasi Sigv4.

Anda dapat mengonfigurasi Penerima Prometheus menggunakan konfigurasi Prometheus yang ada untuk melakukan penemuan layanan dan pengikisan metrik. Penerima Prometheus menggores metrik dalam format eksposisi Prometheus. Setiap aplikasi atau titik akhir yang ingin Anda kikis harus dikonfigurasi dengan pustaka klien Prometheus. Penerima Prometheus mendukung set lengkap konfigurasi pengikisan dan pelabelan ulang Prometheus yang dijelaskan dalam Konfigurasi dalam dokumentasi Prometheus. Anda dapat menempelkan konfigurasi ini langsung ke konfigurasi ADOT Collector Anda.

Prometheus Remote Write Exporter menggunakan titik akhir untuk mengirim metrik `remote_write` yang tergores ke ruang kerja portal manajemen Anda. Permintaan HTTP untuk mengekspor data

akan ditandatangani dengan AWS SiGv4, AWS protokol untuk otentikasi aman, dengan Ekstensi Otentikasi Sigv4. Untuk informasi selengkapnya, lihat proses penandatanganan Signature Version 4.

Kolektor secara otomatis menemukan titik akhir metrik Prometheus di Amazon EKS dan menggunakan konfigurasi yang ditemukan di. <kubernetes_sd_config>

Demo berikut adalah contoh konfigurasi ini pada cluster yang menjalankan Amazon Elastic Kubernetes Service atau Kubernetes yang dikelola sendiri. Untuk melakukan langkah-langkah ini, Anda harus memiliki AWS kredensil dari salah satu opsi potensial dalam rantai AWS kredensi default. Untuk informasi selengkapnya, lihat Mengonfigurasi AWS SDK for Go. Demo ini menggunakan contoh aplikasi yang digunakan untuk pengujian integrasi proses. Aplikasi sampel mengekspos metrik di `/metrics` titik akhir, seperti pustaka klien Prometheus.

Prasyarat

Sebelum memulai langkah-langkah penyiapan konsumsi berikut, Anda harus menyiapkan peran IAM Anda untuk akun layanan dan kebijakan kepercayaan.

Untuk mengatur peran IAM untuk akun layanan dan kebijakan kepercayaan

1. Buat peran IAM untuk akun layanan dengan mengikuti langkah-langkah diMenyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS.

   Kolektor ADOT akan menggunakan peran ini saat menggores dan mengekspor metrik.
2. Selanjutnya, edit kebijakan kepercayaan. Buka konsol IAM di https://console.aws.amazon.com/iam/.
3. Di panel navigasi kiri, pilih Peran dan temukan amp-iamproxy-ingest-roleyang Anda buat di langkah 1.
4. Pilih tab Trust relationship dan pilih Edit trust relationship.
5. Dalam kebijakan hubungan kepercayaan JSON, ganti `aws-amp` dengan `adot-col` lalu pilih Perbarui Kebijakan Kepercayaan. Kebijakan kepercayaan yang Anda hasilkan akan terlihat seperti berikut:

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

```
        "Federated": "arn:aws:iam::account-id:oidc-provider/
oidc.eks.region.amazonaws.com/id/openid"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region.amazonaws.com/id/openid:sub":
 "system:serviceaccount:adot-col:amp-iamproxy-ingest-service-account"
        }
      }
    }
  ]
}
```

6. Pilih tab Izin dan pastikan bahwa kebijakan izin berikut dilampirkan ke peran.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:RemoteWrite",
                "aps:GetSeries",
                "aps:GetLabels",
                "aps:GetMetricMetadata"
            ],
            "Resource": "*"
        }
    ]
}
```

Mengaktifkan koleksi metrik Prometheus

> **Note**
>
> Saat Anda membuat namespace di Amazon EKS, `alertmanager` dan pengekspor node dinonaktifkan secara default.

Untuk mengaktifkan koleksi Prometheus di Amazon EKS atau klaster Kubernetes

1. Fork dan kloning aplikasi sampel dari repositori di. aws-otel-community

   Kemudian jalankan perintah berikut.

   ```
   cd ./sample-apps/prometheus-sample-app
   docker build . -t prometheus-sample-app:latest
   ```

2. Dorong gambar ini ke registri seperti Amazon ECR atau DockerHub.
3. Terapkan aplikasi sampel di cluster dengan menyalin konfigurasi Kubernetes ini dan menerapkannya. Ubah gambar ke gambar yang baru saja Anda dorong `{{PUBLIC_SAMPLE_APP_IMAGE}}` dengan mengganti `prometheus-sample-app.yaml` file.

   ```
   curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/
   main/examples/eks/aws-prometheus/prometheus-sample-app.yaml -o prometheus-sample-
   app.yaml
   kubectl apply -f prometheus-sample-app.yaml
   ```

4. Masukkan perintah berikut untuk memverifikasi bahwa aplikasi sampel telah dimulai. Dalam output perintah, Anda akan melihat `prometheus-sample-app` di `NAME` kolom.

   ```
   kubectl get all -n aoc-prometheus-pipeline-demo
   ```

5. Mulai contoh default dari ADOT Collector. Untuk melakukannya, pertama-tama masukkan perintah berikut untuk menarik konfigurasi Kubernetes untuk ADOT Collector.

   ```
   curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
   examples/eks/aws-prometheus/prometheus-daemonset.yaml -o prometheus-daemonset.yaml
   ```

   Kemudian edit file template, ganti titik akhir remote_write untuk Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus untuk dan Wilayah Anda. `YOUR_ENDPOINT` `YOUR_REGION` Gunakan endpoint remote_write yang ditampilkan di Amazon Managed Service untuk konsol Prometheus saat Anda melihat detail ruang kerja Anda.

   Anda juga harus mengubah `YOUR_ACCOUNT_ID` bagian akun layanan konfigurasi Kubernetes ke ID akun AndaAWS.

   Dalam contoh ini, konfigurasi ADOT Collector menggunakan anotasi (`scrape=true`) untuk memberi tahu titik akhir target mana yang akan dikikis. Hal ini memungkinkan Kolektor ADOT

untuk membedakan titik akhir aplikasi sampel dari titik akhir kube-system di cluster Anda. Anda dapat menghapus ini dari konfigurasi label ulang jika Anda ingin mengikis aplikasi sampel yang berbeda.

6. Masukkan perintah berikut untuk menyebarkan kolektor ADOT.

```
kubectl apply -f prometheus-daemonset.yaml
```

7. Masukkan perintah berikut untuk memverifikasi bahwa kolektor ADOT telah dimulai. Cari `adot-col` di `NAMESPACE` kolom.

```
kubectl get pods -n adot-col
```

8. Verifikasi bahwa pipeline berfungsi dengan menggunakan eksportir logging. Contoh template kami sudah terintegrasi dengan eksportir logging. Masukkan perintah berikut.

```
kubectl get pods -A
kubectl logs -n adot-col name_of_your_adot_collector_pod
```

Beberapa metrik yang tergores dari aplikasi sampel akan terlihat seperti contoh berikut.

```
Resource labels:
     -> service.name: STRING(kubernetes-service-endpoints)
     -> host.name: STRING(192.168.16.238)
     -> port: STRING(8080)
     -> scheme: STRING(http)
InstrumentationLibraryMetrics #0
Metric #0
Descriptor:
     -> Name: test_gauge0
     -> Description: This is my gauge
     -> Unit:
     -> DataType: DoubleGauge
DoubleDataPoints #0
StartTime: 0
Timestamp: 1606511460471000000
Value: 0.000000
```

9. Untuk menguji apakah Amazon Managed Service untuk Prometheus menerima metrik, gunakan. `awscurl` Alat ini memungkinkan Anda mengirim permintaan HTTP melalui baris perintah dengan otentikasi AWS Sigv4, jadi Anda harus memiliki AWS kredensyal yang disiapkan secara

lokal dengan izin yang benar untuk kueri dari Amazon Managed Service untuk Prometheus Untuk petunjuk tentang penginstalan, lihat awscurl. `awscurl`

Dalam perintah berikut, gantiAMP_REGION, dan AMP_ENDPOINT dengan informasi untuk Amazon Managed Service untuk ruang kerja Prometheus.

```
awscurl --service="aps" --region="AMP_REGION" "https://AMP_ENDPOINT/api/v1/query?
query=adot_test_gauge0"
{"status":"success","data":{"resultType":"vector","result":[{"metric":
{"__name__":"adot_test_gauge0"},"value":[1606512592.493,"16.87214000011479"]}]}}
```

Jika Anda menerima metrik sebagai respons, itu berarti penyiapan pipeline Anda telah berhasil dan metrik telah berhasil disebarkan dari aplikasi sampel ke Amazon Managed Service for Prometheus.

Membersihkan

Untuk membersihkan demo ini, masukkan perintah berikut.

```
kubectl delete namespace aoc-prometheus-pipeline-demo
kubectl delete namespace adot-col
```

Konfigurasi lanjutan

Penerima Prometheus mendukung set lengkap konfigurasi pengikisan dan pelabelan ulang Prometheus yang dijelaskan dalam Konfigurasi dalam dokumentasi Prometheus. Anda dapat menempelkan konfigurasi ini langsung ke konfigurasi ADOT Collector Anda.

Konfigurasi untuk Penerima Prometheus mencakup penemuan layanan Anda, konfigurasi pengikisan, dan konfigurasi pelabelan ulang. Konfigurasi penerima terlihat seperti berikut ini.

```
receivers:
  prometheus:
    config:
      [Your Prometheus configuration]
```

Berikut ini adalah contoh konfigurasi.

```
receivers:
```

```
  prometheus:
    config:
      global:
        scrape_interval: 1m
        scrape_timeout: 10s

      scrape_configs:
      - job_name: kubernetes-service-endpoints
        sample_limit: 10000
        kubernetes_sd_configs:
        - role: endpoints
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
          insecure_skip_verify: true
        bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
```

Jika Anda memiliki konfigurasi Prometheus yang ada, Anda harus mengganti $ karakter $$ dengan untuk menghindari nilai diganti dengan variabel lingkungan. *Ini sangat penting untuk nilai penggantian relabel_configurations. Misalnya, jika Anda memulai dengan relabel_configuration berikut:

```
relabel_configs:
- source_labels:
 [__meta_kubernetes_ingress_scheme,__address__,__meta_kubernetes_ingress_path]
  regex: (.+);(.+);(.+)
  replacement: ${1}://${2}${3}
  target_label: __param_target
```

Itu akan menjadi sebagai berikut:

```
relabel_configs:
- source_labels:
 [__meta_kubernetes_ingress_scheme,__address__,__meta_kubernetes_ingress_path]
  regex: (.+);(.+);(.+)
  replacement: $${1}://${2}${3}
  target_label: __param_target
```

Prometheus eksportir tulis jarak jauh dan ekstensi otentikasi Sigv4

Konfigurasi untuk Prometheus Remote Write Exporter dan Sigv4 Authentication Extension lebih sederhana daripada penerima Prometheus. Pada tahap ini, metrik telah dicerna, dan kami siap

mengekspor data ini ke Amazon Managed Service untuk Prometheus. Persyaratan minimum untuk konfigurasi yang berhasil untuk berkomunikasi dengan Amazon Managed Service untuk Prometheus ditampilkan dalam contoh berikut.

```
extensions:
  sigv4auth:
    service: "aps"
    region: "user-region"
exporters:
  prometheusremotewrite:
    endpoint: "https://aws-managed-prometheus-endpoint/api/v1/remote_write"
    auth:
      authenticator: "sigv4auth"
```

Konfigurasi ini mengirimkan permintaan HTTPS yang ditandatangani oleh AWS SigV4 menggunakan AWS kredensil dari rantai kredensil default. AWS Untuk informasi lebih lanjut, lihat Mengonfigurasi AWS SDK for Go. Anda harus menentukan layanan yang akan menjadi`aps`.

Terlepas dari metode penyebaran, kolektor ADOT harus memiliki akses ke salah satu opsi yang tercantum dalam rantai AWS kredensil default. Ekstensi Otentikasi Sigv4 bergantung pada AWS SDK for Go dan menggunakannya untuk mengambil kredensyal dan mengautentikasi. Anda harus memastikan bahwa kredensyal ini memiliki izin menulis jarak jauh untuk Amazon Managed Service for Prometheus.

Siapkan konsumsi metrik dari Amazon ECS menggunakan AWS Distro untuk Open Telemetry

Bagian ini menjelaskan cara mengumpulkan metrik dari Amazon Elastic Container Service (Amazon ECS) dan memasukkannya ke dalam Amazon Managed Service untuk Prometheus menggunakan Distro for Open Telemetry (ADOT). AWS Ini juga menjelaskan cara memvisualisasikan metrik Anda di Grafana Terkelola Amazon.

Prasyarat

> **Important**
>
> Sebelum memulai, Anda harus memiliki lingkungan Amazon ECS di AWS Fargate klaster dengan pengaturan default, Layanan Terkelola Amazon untuk ruang kerja Prometheus, dan ruang kerja Grafana yang Dikelola Amazon. Kami berasumsi bahwa Anda sudah familiar

dengan beban kerja container, Amazon Managed Service for Prometheus, dan Amazon Managed Grafana.

Untuk informasi selengkapnya, lihat tautan berikut:

- Untuk informasi tentang cara membuat lingkungan Amazon ECS di klaster Fargate dengan setelan default, lihat Membuat klaster di Panduan Pengembang Amazon ECS.
- Untuk informasi tentang cara membuat Layanan Terkelola Amazon untuk ruang kerja Prometheus, lihat Membuat ruang kerja di Panduan Pengguna Layanan Terkelola Amazon untuk Prometheus.
- Untuk informasi tentang cara membuat ruang kerja Grafana Terkelola Amazon, lihat Membuat ruang kerja di Panduan Pengguna Grafana Terkelola Amazon.

Tentukan gambar wadah kolektor ADOT khusus

Gunakan file konfigurasi berikut sebagai template untuk menentukan gambar kontainer kolektor ADOT Anda sendiri. Ganti *my-remote-URL dan my-region dengan nilai dan Anda*. `endpoint` `region` Simpan konfigurasi dalam file bernama adot-config.yaml.

> **Note**
>
> Konfigurasi ini menggunakan `sigv4auth` ekstensi untuk mengautentikasi panggilan ke Amazon Managed Service untuk Prometheus. Untuk informasi selengkapnya tentang mengonfigurasi`sigv4auth`, lihat Authenticator - Sigv4 on. GitHub

```
receivers:
  prometheus:
    config:
      global:
        scrape_interval: 15s
        scrape_timeout: 10s
      scrape_configs:
      - job_name: "prometheus"
        static_configs:
        - targets: [ 0.0.0.0:9090 ]
  awsecscontainermetrics:
    collection_interval: 10s
processors:
```

```
  filter:
    metrics:
      include:
        match_type: strict
        metric_names:
          - ecs.task.memory.utilized
          - ecs.task.memory.reserved
          - ecs.task.cpu.utilized
          - ecs.task.cpu.reserved
          - ecs.task.network.rate.rx
          - ecs.task.network.rate.tx
          - ecs.task.storage.read_bytes
          - ecs.task.storage.write_bytes
exporters:
  prometheusremotewrite:
    endpoint: my-remote-URL
    auth:
      authenticator: sigv4auth
  logging:
    loglevel: info
extensions:
  health_check:
  pprof:
    endpoint: :1888
  zpages:
    endpoint: :55679
  sigv4auth:
    region: my-region
    service: aps
service:
  extensions: [pprof, zpages, health_check, sigv4auth]
  pipelines:
    metrics:
      receivers: [prometheus]
      exporters: [logging, prometheusremotewrite]
    metrics/ecs:
      receivers: [awsecscontainermetrics]
      processors: [filter]
      exporters: [logging, prometheusremotewrite]
```

Dorong gambar kontainer kolektor ADOT Anda ke repositori Amazon ECR

Gunakan Dockerfile untuk membuat dan mendorong image container Anda ke repositori Amazon Elastic Container Registry (ECR).

1. Bangun Dockerfile untuk menyalin dan menambahkan gambar kontainer Anda ke gambar OTEL Docker.

```
FROM public.ecr.aws/aws-observability/aws-otel-collector:latest
COPY adot-config.yaml /etc/ecs/otel-config.yaml
CMD ["--config=/etc/ecs/otel-config.yaml"]
```

2. Buat repositori Amazon ECR.

```
# create repo:
COLLECTOR_REPOSITORY=$(aws ecr create-repository --repository aws-otel-collector \
                                --query repository.repositoryUri --output text)
```

3. Buat gambar kontainer Anda.

```
# build ADOT collector image:
docker build -t $COLLECTOR_REPOSITORY:ecs .
```

> **Note**
>
> Ini mengasumsikan Anda sedang membangun wadah Anda di lingkungan yang sama dengan yang akan dijalankan. Jika tidak, Anda mungkin perlu menggunakan `--platform` parameter saat membangun gambar.

4. Masuk ke repositori Amazon ECR. Ganti *wilayah saya dengan nilai* Anda`region`.

```
# sign in to repo:
aws ecr get-login-password --region my-region | \
        docker login --username AWS --password-stdin $COLLECTOR_REPOSITORY
```

5. Dorong gambar wadah Anda.

```
# push ADOT collector image:
docker push $COLLECTOR_REPOSITORY:ecs
```

Buat definisi tugas Amazon ECS untuk mengikis Layanan Terkelola Amazon untuk Prometheus

Buat definisi tugas Amazon ECS untuk mengikis Layanan Terkelola Amazon untuk Prometheus. Definisi tugas Anda harus menyertakan wadah bernama `adot-collector` dan wadah bernama`prometheus`. `prometheus`menghasilkan metrik, dan `adot-collector` goresan`prometheus`.

> **Note**
>
> Amazon Managed Service untuk Prometheus berjalan sebagai layanan, mengumpulkan metrik dari container. Kontainer dalam hal ini menjalankan Prometheus secara lokal, dalam mode Agen, yang mengirim metrik lokal ke Amazon Managed Service untuk Prometheus.

Contoh: Definisi tugas

Berikut ini adalah contoh bagaimana definisi tugas Anda mungkin terlihat. Anda dapat menggunakan contoh ini sebagai template untuk membuat definisi tugas Anda sendiri. Ganti `image` nilai `adot-collector` dengan URL repositori dan tag gambar ()`$COLLECTOR_REPOSITORY:ecs`. Ganti `region` nilai `adot-collector` dan `prometheus` dengan `region` nilai-nilai Anda.

```
{
  "family": "adot-prom",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "adot-collector",
      "image": "account_id.dkr.ecr.region.amazonaws.com/image-tag",
      "essential": true,
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/ecs-adot-collector",
          "awslogs-region": "my-region",
          "awslogs-stream-prefix": "ecs",
          "awslogs-create-group": "True"
        }
      }
    },
    {
      "name": "prometheus",
```

```
      "image": "prom/prometheus:main",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/ecs-prom",
          "awslogs-region": "my-region",
          "awslogs-stream-prefix": "ecs",
          "awslogs-create-group": "True"
        }
      }
    }
  ],
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "cpu": "1024"
}
```

Lampirkan kebijakan AWS terkelola **`AmazonPrometheusRemoteWriteAccess`** ke peran IAM untuk tugas Anda

Untuk mengirim metrik yang tergores ke Amazon Managed Service for Prometheus, tugas Amazon ECS Anda harus memiliki izin yang benar untuk memanggil operasi API untuk Anda. AWS Anda harus membuat peran IAM untuk tugas-tugas Anda dan melampirkan `AmazonPrometheusRemoteWriteAccess` kebijakan untuk itu. Untuk informasi selengkapnya tentang membuat peran ini dan melampirkan kebijakan, lihat Membuat peran dan kebijakan IAM untuk tugas Anda.

Setelah Anda melampirkan `AmazonPrometheusRemoteWriteAccess` ke peran IAM Anda, dan menggunakan peran itu untuk tugas Anda, Amazon ECS dapat mengirim metrik yang digores ke Amazon Managed Service for Prometheus.

Visualisasikan metrik Anda di Grafana Terkelola Amazon

> **Important**
>
> Sebelum memulai, Anda harus menjalankan tugas Fargate pada definisi tugas Amazon ECS Anda. Jika tidak, Layanan Terkelola Amazon untuk Prometheus tidak dapat menggunakan metrik Anda.

1. Dari panel navigasi di ruang kerja Grafana Terkelola Amazon, pilih Sumber data di bawah ikon. AWS
2. Pada tab Sumber data, untuk Layanan, pilih Amazon Managed Service for Prometheus dan pilih Wilayah Default Anda.
3. Pilih Tambahkan sumber data.
4. Gunakan `prometheus` awalan `ecs` dan untuk menanyakan dan melihat metrik Anda.

Mengatur konsumsi metrik dari instans Amazon EC2 menggunakan penulisan jarak jauh

Bagian ini menjelaskan cara menjalankan server Prometheus dengan penulisan jarak jauh di instance Amazon Elastic Compute Cloud (Amazon EC2). Ini menjelaskan cara mengumpulkan metrik dari aplikasi demo yang ditulis dalam Go dan mengirimkannya ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Prasyarat

> **Important**
>
> Sebelum Anda mulai, Anda harus menginstal Prometheus v2.26 atau yang lebih baru. Kami berasumsi bahwa Anda sudah familiar dengan Prometheus, Amazon EC2, dan Amazon Managed Service untuk Prometheus. Untuk informasi tentang cara menginstal Prometheus, lihat Memulai di situs web Prometheus.

Jika Anda tidak terbiasa dengan Amazon EC2 atau Amazon Managed Service untuk Prometheus, sebaiknya mulai dengan membaca bagian berikut:

- Apa itu Amazon Elastic Compute Cloud?
- Apa itu Layanan Dikelola Amazon untuk Prometheus?

Buat peran IAM untuk Amazon EC2

Untuk mengalirkan metrik, Anda harus terlebih dahulu membuat peran IAM dengan kebijakan AWS terkelola. AmazonPrometheusRemoteWriteAccess Kemudian, Anda dapat meluncurkan instance dengan metrik peran dan streaming ke ruang kerja Amazon Managed Service for Prometheus.

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Dari panel navigasi, pilih Peran, lalu pilih Buat peran.
3. Untuk jenis entitas tepercaya, pilih AWSlayanan. Untuk kasus penggunaan, pilih EC2. Pilih Next: Permissions (Selanjutnya: Izin).
4. Di bilah pencarian, masukkan AmazonPrometheusRemoteWriteAccess. Untuk nama Kebijakan, pilih AmazonPrometheusRemoteWriteAccess, lalu pilih Lampirkan kebijakan. Pilih Next:Tags.
5. (Opsional) Buat tag IAM untuk peran IAM Anda. Pilih Next: Review (Selanjutnya: Tinjauan).
6. Masukkan nama untuk peran Anda. Pilih Buat kebijakan.

Luncurkan instans Amazon EC2

Untuk meluncurkan instans Amazon EC2, ikuti petunjuk di Luncurkan instans di Panduan Pengguna Amazon Elastic Compute Cloud untuk Instans Linux.

Jalankan aplikasi demo

1. Gunakan template berikut untuk membuat file Go bernama`main.go`.

```
package main

import (
    "github.com/prometheus/client_golang/prometheus/promhttp"
    "net/http"
)

func main() {
    http.Handle("/metrics", promhttp.Handler())

    http.ListenAndServe(":8000", nil)
}
```

2. Jalankan perintah berikut untuk menginstal dependensi yang benar.

```
sudo yum update -y
sudo yum install -y golang
go get github.com/prometheus/client_golang/prometheus/promhttp
```

3. Jalankan aplikasi demo.

```
go run main.go
```

Aplikasi demo harus berjalan di port 8000 dan menampilkan semua metrik Prometheus yang terbuka. Berikut ini adalah contoh metrik ini.

```
curl -s http://localhost:8000/metrics
...
process_max_fds 4096# HELP process_open_fds Number of open file descriptors.# TYPE
 process_open_fds gauge
process_open_fds 10# HELP process_resident_memory_bytes Resident memory size in
 bytes.# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 1.0657792e+07# HELP process_start_time_seconds Start
 time of the process since unix epoch in seconds.# TYPE process_start_time_seconds
 gauge
process_start_time_seconds 1.61131955899e+09# HELP process_virtual_memory_bytes
 Virtual memory size in bytes.# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 7.77281536e+08# HELP process_virtual_memory_max_bytes
 Maximum amount of virtual memory available in bytes.# TYPE
 process_virtual_memory_max_bytes gauge
process_virtual_memory_max_bytes -1# HELP
 promhttp_metric_handler_requests_in_flight Current number of scrapes being
 served.# TYPE promhttp_metric_handler_requests_in_flight gauge
promhttp_metric_handler_requests_in_flight 1# HELP
 promhttp_metric_handler_requests_total Total number of scrapes by HTTP status
 code.# TYPE promhttp_metric_handler_requests_total counter
promhttp_metric_handler_requests_total{code="200"} 1
promhttp_metric_handler_requests_total{code="500"} 0
promhttp_metric_handler_requests_total{code="503"} 0
```

Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus

Untuk membuat Amazon Managed Service untuk ruang kerja Prometheus, ikuti petunjuk di Buat ruang kerja.

Jalankan server Prometheus

1. Gunakan contoh berikut file YAMAL sebagai template untuk membuat file baru bernama`prometheus.yaml`. Untuk`url`, ganti wilayah *`saya dengan nilai Wilayah`* Anda dan *`my-workspace-id`*dengan ID ruang kerja yang dihasilkan Amazon Managed Service untuk

Prometheus untuk Anda. Untuk`region`, ganti *wilayah saya dengan nilai Wilayah* Anda.

Contoh: file YAMM

```
global:
  scrape_interval: 15s
  external_labels:
    monitor: 'prometheus'

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:8000']

remote_write:
  -
    url: https://aps-workspaces.my-region.amazonaws.com/workspaces/my-workspace-id/
api/v1/remote_write
    queue_config:
        max_samples_per_send: 1000
        max_shards: 200
        capacity: 2500
    sigv4:
         region: my-region
```

2. Jalankan server Prometheus untuk mengirim metrik aplikasi demo ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

```
prometheus --config.file=prometheus.yaml
```

Server Prometheus sekarang harus mengirim metrik aplikasi demo ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

Menggunakan contoh Prometheus sebagai kolektor

Topik berikut menjelaskan berbagai cara untuk menyiapkan instance Prometheus yang berjalan dalam mode agen sebagai kolektor untuk metrik Anda.

> **Warning**
>
> Hindari mengekspos titik akhir Prometheus Scrape ke internet publik dengan mengaktifkan fitur keamanan.

Jika Anda menyiapkan beberapa instans Prometheus yang memantau kumpulan metrik yang sama dan mengirimkannya ke satu Layanan Terkelola Amazon untuk ruang kerja Prometheus untuk ketersediaan tinggi, Anda perlu menyiapkan deduplikasi. Jika Anda tidak mengikuti langkah-langkah untuk mengatur deduplikasi, Anda akan dikenakan biaya untuk semua sampel data yang dikirim ke Amazon Managed Service untuk Prometheus, termasuk sampel duplikat. Untuk petunjuk tentang pengaturan deduplikasi, lihat. Mendeduplikasi metrik ketersediaan tinggi yang dikirim ke Amazon Managed Service untuk Prometheus

Topik

- Mengatur konsumsi dari server Prometheus baru menggunakan Helm
- Siapkan konsumsi dari server Prometheus yang ada di Kubernetes di EC2
- Siapkan konsumsi dari server Prometheus yang ada di Kubernetes di Fargate

Mengatur konsumsi dari server Prometheus baru menggunakan Helm

Petunjuk di bagian ini membuat Anda siap dan menjalankan Layanan Terkelola Amazon untuk Prometheus dengan cepat. Anda menyiapkan server Prometheus baru di klaster Amazon EKS, dan server baru menggunakan konfigurasi default untuk mengirim metrik ke Amazon Managed Service untuk Prometheus. Metode ini memiliki prasyarat berikut:

- Anda harus memiliki cluster Amazon EKS tempat server Prometheus baru akan mengumpulkan metrik
- Anda harus menggunakan Helm CLI 3.0 atau yang lebih baru
- Anda harus menggunakan komputer Linux atau macOS untuk melakukan langkah-langkah di bagian berikut

Langkah 1: Tambahkan repositori bagan Helm baru

Untuk menambahkan repositori bagan Helm baru, masukkan perintah berikut. Untuk informasi selengkapnya tentang perintah ini, lihat Helm Repo.

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo add kube-state-metrics https://kubernetes.github.io/kube-state-metrics
helm repo update
```

Langkah 2: Buat namespace Prometheus

Masukkan perintah berikut untuk membuat namespace Prometheus untuk server Prometheus dan komponen pemantauan lainnya. Ganti *prometheus-namespace* dengan nama yang Anda inginkan untuk namespace ini.

```
kubectl create namespace prometheus-namespace
```

Langkah 3: Siapkan peran IAM untuk akun layanan

Untuk metode orientasi yang kami dokumentasikan, Anda perlu menggunakan peran IAM untuk akun layanan di cluster Amazon EKS tempat server Prometheus berjalan.

Dengan peran IAM untuk akun layanan, Anda dapat mengaitkan peran IAM dengan akun layanan Kubernetes. Akun layanan ini kemudian dapat menyediakan izin AWS ke kontainer-kontainer di setiap pod yang menggunakan akun layanan tersebut. Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan.

Jika Anda belum mengatur peran ini, ikuti instruksi di Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS untuk mengatur peran. Instruksi di bagian itu memerlukan penggunaan`eksctl`. Untuk informasi selengkapnya, lihat Memulai dengan Amazon Elastic Kubernetes Service —. `eksctl`

> **Note**
>
> Saat Anda tidak menggunakan EKS atau AWS dan hanya menggunakan kunci akses dan kunci rahasia untuk mengakses Layanan Terkelola Amazon untuk Prometheus, Anda tidak dapat menggunakan SigV4 berbasis. `EKS-IAM-ROLE`

Langkah 4: Siapkan server baru dan mulai menelan metrik

Untuk menginstal server Prometheus baru yang mengirimkan metrik ke Layanan Terkelola Amazon untuk ruang kerja Prometheus, ikuti langkah-langkah berikut.

Untuk menginstal server Prometheus baru untuk mengirim metrik ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus

1. Gunakan editor teks untuk membuat file bernama `my_prometheus_values_yaml` dengan konten berikut.

   - Ganti *IAM_PROXY_PROMETHEUS_ROLE_ARN* dengan ARN yang Anda buat. amp-iamproxy-ingest-roleMenyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS
   - Ganti *WORKSPACE_ID dengan ID* Amazon Managed Service untuk ruang kerja Prometheus.
   - Ganti *WILAYAH* dengan Wilayah Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

```
## The following is a set of default values for prometheus server helm chart which
 enable remoteWrite to AMP
## For the rest of prometheus helm chart values see: https://github.com/prometheus-
community/helm-charts/blob/main/charts/prometheus/values.yaml
##
serviceAccounts:
  server:
    name: amp-iamproxy-ingest-service-account
    annotations:
      eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}
server:
  remoteWrite:
    - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
      sigv4:
        region: ${REGION}
      queue_config:
        max_samples_per_send: 1000
        max_shards: 200
        capacity: 2500
```

2. Masukkan perintah berikut untuk membuat server Prometheus.

   - Ganti *prometheus-chart-name*dengan nama rilis Prometheus Anda.
   - Ganti *prometheus-namespace dengan nama namespace* Prometheus Anda.

```
helm install prometheus-chart-name prometheus-community/prometheus -n prometheus-
namespace \
-f my_prometheus_values_yaml
```

> **Note**
>
> Anda dapat menyesuaikan `helm install` perintah dengan banyak cara. Untuk informasi selengkapnya, lihat Helm install di dokumentasi Helm.

Siapkan konsumsi dari server Prometheus yang ada di Kubernetes di EC2

Layanan Terkelola Amazon untuk Prometheus mendukung pengambilan metrik dari server Prometheus di cluster yang menjalankan Amazon EKS dan di cluster Kubernetes yang dikelola sendiri yang berjalan di Amazon EC2. Petunjuk terperinci di bagian ini adalah untuk server Prometheus di cluster Amazon EKS. Langkah-langkah untuk klaster Kubernetes yang dikelola sendiri di Amazon EC2 adalah sama, kecuali Anda perlu menyiapkan sendiri peran penyedia OIDC dan IAM untuk akun layanan di klaster Kubernetes.

Instruksi di bagian ini menggunakan Helm sebagai manajer paket Kubernetes.

Topik

- Langkah 1: Siapkan peran IAM untuk akun layanan
- Langkah 2: Tingkatkan server Prometheus Anda yang ada menggunakan Helm

Langkah 1: Siapkan peran IAM untuk akun layanan

Untuk metode orientasi yang kami dokumentasikan, Anda perlu menggunakan peran IAM untuk akun layanan di cluster Amazon EKS tempat server Prometheus berjalan. Peran ini juga disebut peran layanan.

Dengan peran layanan, Anda dapat mengaitkan peran IAM dengan akun layanan Kubernetes. Akun layanan ini kemudian dapat menyediakan izin AWS ke kontainer-kontainer di setiap pod yang menggunakan akun layanan tersebut. Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan.

Jika Anda belum mengatur peran ini, ikuti instruksi di Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS untuk mengatur peran.

Langkah 2: Tingkatkan server Prometheus Anda yang ada menggunakan Helm

Petunjuk di bagian ini mencakup pengaturan penulisan jarak jauh dan sigv4 untuk mengautentikasi dan mengotorisasi server Prometheus untuk menulis jarak jauh ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

Menggunakan Prometheus versi 2.26.0 atau yang lebih baru

Ikuti langkah-langkah ini jika Anda menggunakan bagan Helm dengan gambar Prometheus Server versi 2.26.0 atau yang lebih baru.

Untuk mengatur penulisan jarak jauh dari server Prometheus menggunakan bagan Helm

1. Buat bagian penulisan jarak jauh baru di file konfigurasi Helm Anda:

   - Ganti `${IAM_PROXY_PROMETHEUS_ROLE_ARN}` dengan ARN dari amp-iamproxy-ingest-roleyang Anda buat. Langkah 1: Siapkan peran IAM untuk akun layanan Peran ARN harus memiliki format. `arn:aws:iam::`*`your account ID`*`:role/amp-iamproxy-ingest-role`
   - Ganti `${WORKSPACE_ID}` dengan Layanan Terkelola Amazon Anda untuk ID ruang kerja Prometheus.
   - Ganti `${REGION}` dengan Wilayah Layanan Terkelola Amazon untuk ruang kerja Prometheus (seperti). `us-west-2`

```
## The following is a set of default values for prometheus server helm chart which
 enable remoteWrite to AMP
    ## For the rest of prometheus helm chart values see: https://github.com/
prometheus-community/helm-charts/blob/main/charts/prometheus/values.yaml
    ##
    serviceAccounts:
      server:
        name: amp-iamproxy-ingest-service-account
        annotations:
          eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}
    server:
      remoteWrite:
```

```
        - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
          sigv4:
            region: ${REGION}
          queue_config:
            max_samples_per_send: 1000
            max_shards: 200
            capacity: 2500
```

2. Perbarui konfigurasi Server Prometheus Anda yang ada menggunakan Helm:

   - Ganti `prometheus-chart-name` dengan nama rilis Prometheus Anda.
   - Ganti `prometheus-namespace` dengan namespace Kubernetes tempat Server Prometheus Anda diinstal.
   - Ganti `my_prometheus_values_yaml` dengan path ke file konfigurasi Helm Anda.
   - Ganti `current_helm_chart_version` dengan versi grafik Helm Server Prometheus Anda saat ini. Anda dapat menemukan versi bagan saat ini dengan menggunakan perintah helm list.

```
helm upgrade prometheus-chart-name prometheus-community/prometheus \
       -n prometheus-namespace \
       -f my_prometheus_values_yaml \
       --version current_helm_chart_version
```

Menggunakan Prometheus versi sebelumnya

Ikuti langkah-langkah ini jika Anda menggunakan versi Prometheus lebih awal dari 2.26.0. Langkah-langkah ini menggunakan pendekatan sespan, karena versi Prometheus sebelumnya tidak mendukung AWS proses penandatanganan Signature Version 4 (SigV4). AWS

Instruksi ini mengasumsikan bahwa Anda menggunakan Helm untuk menyebarkan Prometheus.

Untuk mengatur penulisan jarak jauh dari server Prometheus

1. Di server Prometheus Anda, buat konfigurasi penulisan jarak jauh baru. Pertama, buat file pembaruan baru. Kami akan memanggil file tersebutamp_`ingest_override_values.yaml`.

   Tambahkan nilai berikut ke file YAMM.

```
serviceAccounts:
```

```
        server:
            name: "amp-iamproxy-ingest-service-account"
            annotations:
                eks.amazonaws.com/role-arn:
 "${SERVICE_ACCOUNT_IAM_INGEST_ROLE_ARN}"
    server:
        sidecarContainers:
            - name: aws-sigv4-proxy-sidecar
              image: public.ecr.aws/aws-observability/aws-sigv4-proxy:1.0
              args:
              - --name
              - aps
              - --region
              - ${REGION}
              - --host
              - aps-workspaces.${REGION}.amazonaws.com
              - --port
              - :8005
              ports:
              - name: aws-sigv4-proxy
                containerPort: 8005
        statefulSet:
            enabled: "true"
        remoteWrite:
            - url: http://localhost:8005/workspaces/${WORKSPACE_ID}/api/v1/
remote_write
```

Ganti `${REGION}` dengan Wilayah Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Ganti `${SERVICE_ACCOUNT_IAM_INGEST_ROLE_ARN}` dengan ARN dari amp-iamproxy-ingest-roleyang Anda buat. Langkah 1: Siapkan peran IAM untuk akun layanan Peran ARN harus memiliki format. `arn:aws:iam::`*`your account ID`*`:role/amp-iamproxy-ingest-role`

Ganti `${WORKSPACE_ID}` dengan ID ruang kerja Anda.

2. Tingkatkan bagan Prometheus Helm Anda. Pertama, temukan nama bagan Helm Anda dengan memasukkan perintah berikut. Pada output dari perintah ini, cari bagan dengan nama yang disertakan`prometheus`.

```
helm ls --all-namespaces
```

Masukkan perintah berikut ini.

```
helm upgrade --install prometheus-helm-chart-name prometheus-community/prometheus -
n prometheus-namespace -f ./amp_ingest_override_values.yaml
```

Ganti *prometheus-helm-chart-name*dengan nama bagan helm Prometheus yang dikembalikan pada perintah sebelumnya. Ganti *prometheus-namespace dengan nama namespace* Anda.

Mengunduh grafik Helm

Jika Anda belum mengunduh bagan Helm secara lokal, Anda dapat menggunakan perintah berikut untuk mengunduhnya.

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
    helm pull prometheus-community/prometheus --untar
```

Siapkan konsumsi dari server Prometheus yang ada di Kubernetes di Fargate

Layanan Terkelola Amazon untuk Prometheus mendukung pengambilan metrik dari server Prometheus di cluster Kubernetes yang dikelola sendiri yang berjalan di Fargate. Untuk menyerap metrik dari server Prometheus di kluster Amazon EKS yang berjalan di Fargate, ganti konfigurasi default dalam file konfigurasi bernama amp_ingest_override_values.yaml sebagai berikut:

```
prometheus-node-exporter:
        enabled: false

    alertmanager:
        enabled: false

    serviceAccounts:
      server:
        name: amp-iamproxy-ingest-service-account
        annotations:
          eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}

    server:
      persistentVolume:
        enabled: false
      remoteWrite:
        - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
```

```
          sigv4:
            region: ${REGION}
          queue_config:
            max_samples_per_send: 1000
            max_shards: 200
            capacity: 2500
```

Instal Prometheus menggunakan penggantian dengan perintah berikut:

```
helm install prometheus-for-amp prometheus-community/prometheus \
                  -n prometheus \
                  -f amp_ingest_override_values.yaml
```

Perhatikan bahwa dalam konfigurasi bagan Helm kami menonaktifkan pengekspor node dan manajer peringatan serta menjalankan penyebaran server Prometheus.

Anda dapat memverifikasi instalasi dengan contoh kueri pengujian berikut.

```
$ awscurl --region region --service aps "https://aps-
workspaces.region_id.amazonaws.com/workspaces/workspace_id/api/v1/query?
query=prometheus_api_remote_read_queries"
    {"status":"success","data":{"resultType":"vector","result":[{"metric":
{"__name__":"prometheus_api_remote_read_queries","instance":"localhost:9090","job":"prometheus"
[1648461236.419,"0"]}]}}21
```

Menyiapkan Amazon Managed Service untuk Prometheus untuk data ketersediaan tinggi

Saat Anda mengirim data ke Amazon Managed Service untuk Prometheus, data akan direplikasi secara otomatis AWS di seluruh Availability Zone di Wilayah, dan disajikan kepada Anda dari sekelompok host yang menyediakan skalabilitas, ketersediaan, dan keamanan. Anda mungkin ingin menambahkan brankas kegagalan ketersediaan tinggi tambahan, tergantung pada pengaturan khusus Anda. Ada dua cara umum agar Anda dapat menambahkan keamanan ketersediaan tinggi ke pengaturan Anda:

- Jika Anda memiliki beberapa kontainer atau instans yang memiliki data yang sama, Anda dapat mengirim data tersebut ke Amazon Managed Service untuk Prometheus dan data secara otomatis di-de-duplikasi. Ini membantu memastikan bahwa data Anda akan dikirim ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Untuk informasi selengkapnya tentang menghilangkan duplikasi data ketersediaan tinggi, lihat. Mendeduplikasi metrik ketersediaan tinggi yang dikirim ke Amazon Managed Service untuk Prometheus

- Jika Anda ingin memastikan bahwa Anda memiliki akses ke data Anda, bahkan ketika AWS Wilayah tidak tersedia, Anda dapat mengirim metrik Anda ke ruang kerja kedua, di Wilayah lain.

  Untuk informasi selengkapnya tentang mengirim data metrik ke beberapa ruang kerja, lihat. Ketersediaan Lintas Wilayah

Topik

Mendeduplikasi metrik ketersediaan tinggi yang dikirim ke Amazon Managed Service untuk Prometheus

Anda dapat mengirim data dari beberapa agen Prometheus (instance Prometheus yang berjalan dalam mode Agen) ke Layanan Terkelola Amazon untuk ruang kerja Prometheus. Jika beberapa instans ini merekam dan mengirimkan metrik yang sama, data Anda akan memiliki ketersediaan yang lebih tinggi (meskipun salah satu agen berhenti mengirim data, Layanan Terkelola Amazon untuk ruang kerja Prometheus akan tetap menerima data dari instance lain). Namun, Anda ingin ruang kerja Amazon Managed Service for Prometheus secara otomatis menghapus duplikasi metrik sehingga Anda tidak melihat metrik beberapa kali, dan tidak dikenakan biaya untuk konsumsi dan penyimpanan data beberapa kali.

Agar Amazon Managed Service untuk Prometheus dapat secara otomatis menghapus duplikat data dari beberapa agen Prometheus, Anda memberikan kumpulan agen yang mengirimkan data duplikat satu nama cluster, dan setiap instance nama replika. Nama cluster mengidentifikasi instance sebagai memiliki data bersama, dan nama replika memungkinkan Amazon Managed Service untuk Prometheus mengidentifikasi sumber setiap metrik. Metrik terakhir yang disimpan mencakup label cluster, tetapi bukan replika, sehingga metrik tampaknya berasal dari satu sumber.

Topik berikut menunjukkan cara mengirim data dan menyertakan label cluster dan replika, sehingga Amazon Managed Service for Prometheus menghapus duplikasi data secara otomatis.

> **Important**
>
> Jika Anda tidak mengatur deduplikasi, Anda akan dikenakan biaya untuk semua sampel data yang dikirim ke Amazon Managed Service untuk Prometheus. Sampel data ini termasuk sampel duplikat.

Kirim data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus dengan Prometheus

Untuk menyiapkan konfigurasi ketersediaan tinggi dengan Prometheus, Anda harus menerapkan label eksternal pada semua instance grup ketersediaan tinggi, sehingga Amazon Managed Service for Prometheus dapat mengidentifikasinya. Gunakan `cluster` label untuk mengidentifikasi agen instance Prometheus sebagai bagian dari grup ketersediaan tinggi. Gunakan `__replica__` label untuk mengidentifikasi setiap replika dalam grup secara terpisah. Anda perlu menerapkan keduanya `__replica__` dan `cluster` label agar de-duplikasi berfungsi.

> **Note**
>
> `__replica__`Label diformat dengan dua simbol garis bawah sebelum dan sesudah kata. `replica`

Contoh: potongan kode

Dalam cuplikan kode berikut, `cluster` label mengidentifikasi agen `prom-team1` instance Prometheus, dan label mengidentifikasi replika dan. `_replica_` `replica1` `replica2`

```
cluster: prom-team1
```

```
__replica__: replica1
```

```
cluster: prom-team1
__replica__: replica2
```

Karena Amazon Managed Service untuk Prometheus menyimpan sampel data dari replika ketersediaan tinggi dengan label ini, itu menghapus label saat sampel diterima`replica`. Ini berarti bahwa Anda hanya akan memiliki pemetaan seri 1:1 untuk seri Anda saat ini, bukan seri per replika. `cluster`Label disimpan.

Kirim data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus dengan Operator Prometheus

Untuk menyiapkan konfigurasi ketersediaan tinggi dengan Operator Prometheus, Anda harus menerapkan label eksternal pada semua instance grup ketersediaan tinggi, sehingga Amazon Managed Service for Prometheus dapat mengidentifikasinya. Anda juga harus mengatur atribut `replicaExternalLabelName` dan `externalLabels` pada bagan Helm Operator Prometheus.

Contoh: header YAMM

Di header YAMM berikut, `cluster` ditambahkan `externalLabel` untuk mengidentifikasi agen instans Prometheus sebagai bagian dari grup ketersediaan tinggi, `replicaExternalLabels` dan mengidentifikasi setiap replika dalam grup.

```
replicaExternalLabelName: __replica__
externalLabels:
cluster: prom-dev
```

Kirim data ketersediaan tinggi ke Amazon Managed Service untuk AWS Prometheus dengan Distro untuk Open Telemetry

AWSDistro for Open Telemetry (ADOT) adalah distribusi proyek yang aman dan siap produksi. OpenTelemetry ADOT memberi Anda API sumber, pustaka, dan agen, sehingga Anda dapat mengumpulkan jejak dan metrik terdistribusi untuk pemantauan aplikasi. Untuk informasi tentang ADOT, lihat Tentang AWS Distro untuk Telemetri Terbuka.

Untuk mengatur ADOT dengan konfigurasi ketersediaan tinggi, Anda harus mengonfigurasi gambar kontainer kolektor ADOT dan menerapkan label eksternal `cluster` dan `__replica__` ke eksportir tulis jarak jauh PrometheusAWS. Eksportir ini mengirimkan metrik tergores Anda ke Layanan

Terkelola Amazon untuk ruang kerja Prometheus melalui titik akhir. `remote_write` Saat Anda menyetel label ini pada eksportir tulis jarak jauh, Anda mencegah metrik duplikat disimpan saat replika redundan berjalan. Untuk informasi lebih lanjut tentang eksportir tulis jarak jauh AWS Prometheus, lihat Memulai dengan eksportir tulis jarak jauh Prometheus untuk Layanan Terkelola Amazon untuk Prometheus.

Kirim data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus dengan bagan Helm komunitas Prometheus

Untuk menyiapkan konfigurasi ketersediaan tinggi dengan bagan Helm komunitas Prometheus, Anda harus menerapkan label eksternal pada semua instance grup ketersediaan tinggi, sehingga Layanan Terkelola Amazon untuk Prometheus dapat mengidentifikasinya. Berikut adalah contoh bagaimana Anda dapat menambahkan `external_labels` ke satu contoh Prometheus dari bagan Helm komunitas Prometheus.

```
server:
global:
  external_labels:
      cluster: monitoring-cluster
      __replica__: replica-1
```

> **Note**
>
> Jika Anda menginginkan beberapa replika, Anda harus menerapkan bagan beberapa kali dengan nilai replika yang berbeda, karena bagan Helm komunitas Prometheus tidak memungkinkan Anda mengatur nilai replika secara dinamis saat menambah jumlah replika langsung dari grup pengontrol. Jika Anda lebih suka `replica` label disetel secara otomatis, gunakan bagan Helm prometheus-operator.

FAQ: Konfigurasi ketersediaan tinggi

Haruskah saya memasukkan nilai **`__replica__`** ke label lain untuk melacak titik sampel?

Dalam pengaturan ketersediaan tinggi, Amazon Managed Service untuk Prometheus memastikan sampel data tidak diduplikasi dengan memilih pemimpin dalam cluster instance Prometheus. Jika replika pemimpin berhenti mengirim sampel data selama 30 detik, Layanan Terkelola Amazon untuk Prometheus secara otomatis menjadikan instance Prometheus lain sebagai replika pemimpin dan

menyerap data dari pemimpin baru, termasuk data yang terlewat. Karena itu, jawabannya tidak, tidak disarankan. Melakukannya dapat menyebabkan masalah seperti:

- Meminta a `count` di PromQL dapat mengembalikan nilai yang lebih tinggi dari yang diharapkan selama periode pemilihan pemimpin baru.
- Jumlah `active series` akan meningkat selama periode memilih pemimpin baru dan mencapai. `active series limits` Lihat Kuota AMP untuk info selengkapnya.

Ketersediaan Lintas Wilayah

Untuk menambahkan ketersediaan lintas wilayah ke data Anda, Anda dapat mengirim metrik ke beberapa ruang kerja di seluruh Wilayah. AWS Prometheus mendukung banyak penulis dan penulisan lintas wilayah.

Contoh berikut menunjukkan cara mengatur server Prometheus yang berjalan dalam mode Agen untuk mengirim metrik ke dua ruang kerja di Wilayah yang berbeda dengan Helm.

```
extensions:
      sigv4auth:
        service: "aps"

    receivers:
      prometheus:
        config:
          scrape_configs:
            - job_name: 'kubernetes-kubelet'
              scheme: https
              tls_config:
                ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
                insecure_skip_verify: true
              bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
              kubernetes_sd_configs:
              - role: node
              relabel_configs:
              - action: labelmap
                regex: __meta_kubernetes_node_label_(.+)
              - target_label: __address__
                replacement: kubernetes.default.svc.cluster.local:443
              - source_labels: [__meta_kubernetes_node_name]
                regex: (.+)
                target_label: __metrics_path__
```

```
                replacement: /api/v1/nodes/$${1}/proxy/metrics

    exporters:
      prometheusremotewrite/one:
        endpoint: "https://aps-workspaces.workspace_1_region.amazonaws.com/workspaces/
ws-workspace_1_id/api/v1/remote_write"
        auth:
          authenticator: sigv4auth
      prometheusremotewrite/two:
        endpoint: "https://aps-workspaces.workspace_2_region.amazonaws.com/workspaces/
ws-workspace_2_id/api/v1/remote_write"
        auth:
          authenticator: sigv4auth

    service:
      extensions: [sigv4auth]
      pipelines:
        metrics/one:
          receivers: [prometheus]
          exporters: [prometheusremotewrite/one]
        metrics/two:
          receivers: [prometheus]
          exporters: [prometheusremotewrite/two]
```

Memahami dan mengoptimalkan biaya

Pertanyaan umum berikut dan jawabannya dapat membantu dalam memahami dan mengoptimalkan biaya yang terkait dengan Amazon Managed Service untuk Prometheus.

Apa yang berkontribusi pada biaya saya?

> Bagi sebagian besar pelanggan, konsumsi metrik berkontribusi sebagian besar biaya. Pelanggan dengan penggunaan kueri tinggi juga akan melihat beberapa biaya berdasarkan sampel kueri yang diproses, dengan penyimpanan metrik menjadi pendorong kecil biaya keseluruhan. Untuk informasi selengkapnya tentang harga masing-masing, lihat Harga di halaman produk Layanan Terkelola Amazon untuk Prometheus.

Apa cara terbaik untuk menurunkan biaya saya? Bagaimana cara menurunkan biaya konsumsi?

> Tingkat konsumsi (bukan penyimpanan metrik) adalah sebagian besar biaya bagi sebagian besar pelanggan. Anda dapat mengurangi tingkat konsumsi dengan mengurangi frekuensi pengumpulan (meningkatkan interval pengumpulan) atau dengan mengurangi jumlah seri aktif yang dicerna.

Anda dapat meningkatkan interval pengumpulan (pengikisan) dari agen koleksi Anda: Server Prometheus (berjalan dalam mode Agen) dan kolektor AWS Distro for (ADOT) mendukung konfigurasi. OpenTelemetry `scrape_interval` Misalnya, meningkatkan interval pengumpulan dari 30 detik menjadi 60 detik akan mengurangi penggunaan konsumsi Anda hingga setengahnya.

Anda juga dapat memfilter metrik yang dikirim ke Amazon Managed Service untuk Prometheus dengan menggunakan. `<relabel_config>` Untuk informasi lebih lanjut tentang pelabelan ulang dalam konfigurasi agen Prometheus, lihat https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config di dokumentasi Prometheus.

Apa cara terbaik untuk menurunkan biaya kueri saya?

Biaya kueri didasarkan pada jumlah sampel yang diproses. Anda dapat mengurangi frekuensi kueri untuk mengurangi biaya kueri Anda.

Untuk mendapatkan lebih banyak visibilitas ke kueri yang berkontribusi paling besar terhadap biaya kueri Anda, Anda dapat menghubungi untuk mengajukan tiket dengan kontak dukungan Anda. Tim Amazon Managed Service untuk Prometheus dapat membantu Anda memahami pertanyaan yang berkontribusi paling besar terhadap biaya Anda.

Jika saya mengurangi periode retensi metrik saya, apakah itu akan membantu mengurangi total tagihan saya?

Anda dapat mengurangi periode retensi Anda, namun, ini tidak mungkin secara substansional mengurangi biaya Anda.

Jika Anda ingin mengurangi (atau menambah) periode retensi Anda, Anda dapat mengajukan permintaan batas layanan untuk mengubah `Retention time for ingested data` kuota.

Metrik apa yang dapat saya gunakan untuk memantau biaya saya?

Pantau `IngestionRate` di Amazon CloudWatch untuk melacak biaya konsumsi Anda. Untuk informasi selengkapnya tentang pemantauan Amazon Managed Service untuk metrik Prometheus, lihat. CloudWatch CloudWatch metrik

Bisakah saya memeriksa tagihan saya kapan saja?

AWS Cost and Usage ReportLacak AWS penggunaan Anda dan memberikan perkiraan biaya yang terkait dengan akun Anda dalam periode penagihan. Untuk informasi selengkapnya, lihat Apa itu Laporan AWS Biaya dan Penggunaan? dalam Panduan Pengguna Laporan AWS Biaya dan Penggunaan

Mengapa tagihan saya lebih tinggi di awal bulan daripada di akhir bulan?

Amazon Managed Service untuk Prometheus memiliki model penetapan harga berjenjang untuk konsumsi, yang mengakibatkan biaya penggunaan awal Anda menjadi lebih tinggi. Saat penggunaan Anda mencapai tingkat konsumsi yang lebih tinggi, dengan biaya lebih rendah, biaya Anda lebih rendah. Untuk informasi selengkapnya tentang harga, termasuk tingkatan konsumsi, lihat Harga di halaman produk Layanan Terkelola Amazon untuk Prometheus.

> **Note**
>
> Tingkatan adalah per akun pembayar, bukan per akun, jadi ketika total metrik yang dicerna untuk semua akun dalam organisasi mencapai tingkat berikutnya, semua akun kemudian dikenakan tarif yang lebih rendah.

Kueri metrik Prometheus Anda

Sekarang metrik sedang dicerna ke ruang kerja, Anda dapat menanyakannya. Anda dapat menggunakan layanan seperti Grafana untuk menanyakan metrik, atau Anda dapat menggunakan Amazon Managed Service for Prometheus API.

Anda melakukan kueri Anda menggunakan bahasa kueri Prometheus standar, PromQL. Untuk informasi selengkapnya tentang PromQL dan sintaksnya, lihat Meminta Prometheus dalam dokumentasi Prometheus.

Topik

Mengamankan kueri metrik Anda

Layanan Terkelola Amazon untuk Prometheus menyediakan cara untuk membantu Anda mengamankan kueri metrik Anda.

Menggunakan AWS PrivateLink dengan Amazon Managed Service untuk Prometheus

Lalu lintas jaringan untuk menanyakan metrik di Amazon Managed Service untuk Prometheus dapat dilakukan melalui titik akhir internet publik, atau melalui titik akhir VPC. AWS PrivateLink Saat Anda menggunakanAWS PrivateLink, lalu lintas jaringan dari VPC Anda diamankan di dalam AWS jaringan tanpa melalui internet publik. Untuk membuat titik akhir AWS PrivateLink VPC untuk Amazon Managed Service untuk Prometheus, lihat. Menggunakan Amazon Managed Service untuk Prometheus dengan titik akhir VPC antarmuka

Autentikasi dan otorisasi

AWS Identity and Access Managementadalah layanan web yang membantu Anda mengontrol akses ke AWS sumber daya dengan aman. Anda menggunakan IAM untuk mengontrol siapa yang diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya. Amazon Managed Service for Prometheus terintegrasi dengan IAM untuk membantu Anda menjaga keamanan data. Saat menyiapkan Amazon Managed Service untuk Prometheus, Anda harus membuat beberapa peran IAM yang memungkinkan server Grafana untuk menanyakan metrik yang disimpan di Amazon Managed Service untuk ruang kerja Prometheus. Untuk informasi selengkapnya tentang IAM, lihat Apa itu IAM?.

Fitur AWS keamanan lain yang dapat membantu Anda menyiapkan Amazon Managed Service untuk Prometheus adalah AWS proses penandatanganan Signature Version 4 (SigV4). AWS Signature Versi 4 adalah proses untuk menambahkan informasi autentikasi ke permintaan AWS yang dikirim oleh HTTP. Demi keamanan, sebagian besar permintaan untuk AWS harus ditandatangani dengan kunci akses, yang terdiri dari ID kunci akses dan kunci akses rahasia. Kedua kunci ini umumnya disebut sebagai kredensial keamanan Anda. Untuk informasi selengkapnya tentang SiGv4, lihat proses penandatanganan Sigv4 Versi Tanda Tangan 4.

Siapkan Grafana Terkelola Amazon untuk digunakan dengan Amazon Managed Service untuk Prometheus

Grafana Terkelola Amazon adalah layanan yang dikelola sepenuhnya untuk Grafana open-source yang menyederhanakan koneksi ke sumber terbuka, ISV pihak ketiga, AWS dan layanan untuk memvisualisasikan dan menganalisis sumber data Anda dalam skala besar.

Layanan Terkelola Amazon untuk Prometheus mendukung penggunaan Grafana Terkelola Amazon untuk menanyakan metrik di ruang kerja. Di konsol Grafana Terkelola Amazon, Anda dapat menambahkan Layanan Terkelola Amazon untuk ruang kerja Prometheus sebagai sumber data dengan menemukan Layanan Terkelola Amazon untuk akun Prometheus yang ada. Grafana yang Dikelola Amazon mengelola konfigurasi kredensyal otentikasi yang diperlukan untuk mengakses Layanan Terkelola Amazon untuk Prometheus. Untuk petunjuk mendetail tentang cara membuat sambungan ke Layanan Terkelola Amazon untuk Prometheus dari Grafana yang Dikelola Amazon, lihat petunjuk di Panduan Pengguna Grafana Terkelola Amazon.

Anda juga dapat melihat peringatan Layanan Terkelola Amazon untuk Prometheus di Grafana Terkelola Amazon. Untuk petunjuk mengatur integrasi dengan peringatan, lihatMengintegrasikan peringatan dengan Grafana Terkelola Amazon atau Grafana open source.

Menghubungkan ke Grafana yang Dikelola Amazon dalam VPC pribadi

Layanan Terkelola Amazon untuk Prometheus menyediakan titik akhir layanan untuk Grafana Terkelola Amazon untuk disambungkan saat menanyakan metrik dan peringatan.

Anda dapat mengonfigurasi Grafana Terkelola Amazon untuk menggunakan VPC pribadi (untuk detail tentang pengaturan VPC pribadi di Grafana, lihat Menyambung ke Amazon VPC di Panduan Pengguna Grafana Terkelola Amazon). Bergantung pada pengaturannya, VPC ini mungkin tidak memiliki akses ke titik akhir layanan Amazon Managed Service for Prometheus.

Untuk menambahkan Layanan Terkelola Amazon untuk Prometheus sebagai sumber data ke ruang kerja Grafana Terkelola Amazon yang dikonfigurasi untuk menggunakan VPC pribadi tertentu, Anda harus terlebih dahulu menghubungkan Layanan Terkelola Amazon untuk Prometheus ke VPC yang sama dengan membuat titik akhir VPC. Untuk informasi selengkapnya tentang membuat titik akhir VPC, lihat. Buat titik akhir VPC antarmuka untuk Amazon Managed Service untuk Prometheus

Siapkan open source Grafana atau Grafana Enterprise untuk digunakan dengan Amazon Managed Service for Prometheus

Amazon Managed Service untuk Prometheus mendukung penggunaan Grafana versi 7.3.5 dan yang lebih baru untuk menanyakan metrik di ruang kerja. Versi 7.3.5 dan yang lebih baru mencakup dukungan untuk otentikasi AWS Signature Version 4 (SigV4).

Untuk petunjuk untuk menyiapkan Grafana mandiri menggunakan file tar.gz atau zip, lihat Menginstal Grafana di dokumentasi Grafana. Jika Anda menginstal Grafana mandiri baru, Anda akan diminta untuk nama pengguna dan kata sandi. Default-nya adalah **`admin/admin`**. Anda akan diminta untuk mengubah kata sandi setelah Anda masuk untuk pertama kalinya. Untuk informasi lebih lanjut, lihat Memulai Grafana di dokumentasi Grafana.

Untuk memeriksa versi Grafana Anda, masukkan perintah berikut.

```
grafana_install_directory/bin/grafana-server -v
```

Untuk mengatur Grafana agar berfungsi dengan Layanan Terkelola Amazon untuk Prometheus, Anda harus masuk ke akun yang memiliki AmazonPrometheusQueryAccesskebijakan atau,,, dan izin. `aps:QueryMetrics` `aps:GetMetricMetadata` `aps:GetSeries` `aps:GetLabels` Untuk informasi selengkapnya, lihat Izin dan kebijakan IAM.

Mengatur AWS SiGv4

Amazon Managed Service for Prometheus bekerja AWS Identity and Access Management dengan (IAM) untuk mengamankan semua panggilan ke Prometheus API dengan kredensyal IAM. Secara default, sumber data Prometheus di Grafana mengasumsikan bahwa Prometheus tidak memerlukan otentikasi. Untuk mengaktifkan Grafana memanfaatkan Layanan Terkelola Amazon untuk kemampuan otentikasi dan otorisasi Prometheus, Anda harus mengaktifkan dukungan otentikasi SiGv4 di sumber data Grafana. Ikuti langkah-langkah di halaman ini saat Anda menggunakan sumber terbuka Grafana yang dikelola sendiri atau server perusahaan Grafana. Jika Anda menggunakan Grafana Terkelola Amazon, otentikasi SiGv4 sepenuhnya otomatis. Untuk informasi selengkapnya tentang Grafana yang Dikelola Amazon, lihat Apa itu Grafana yang Dikelola Amazon?

Untuk mengaktifkan SiGv4 di Grafana, mulai Grafana dengan variabel dan lingkungan yang disetel ke. `AWS_SDK_LOAD_CONFIG` `GF_AUTH_SIGV4_AUTH_ENABLED` `true` Variabel `GF_AUTH_SIGV4_AUTH_ENABLED` lingkungan mengganti konfigurasi default Grafana untuk mengaktifkan dukungan SiGv4. Untuk informasi selengkapnya, lihat Konfigurasi dalam dokumentasi Grafana.

Linux

Untuk mengaktifkan SiGv4 pada server Grafana mandiri di Linux, masukkan perintah berikut.

```
export AWS_SDK_LOAD_CONFIG=true
```

```
export GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

```
./bin/grafana-server
```

Windows

Untuk mengaktifkan SiGv4 pada Grafana mandiri di Windows menggunakan prompt perintah Windows, masukkan perintah berikut.

```
set AWS_SDK_LOAD_CONFIG=true
```

```
set GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

```
.\bin\grafana-server.exe
```

Tambahkan sumber data Prometheus di Grafana

Langkah-langkah berikut menjelaskan cara menyiapkan sumber data Prometheus di Grafana untuk menanyakan metrik Layanan Terkelola Amazon Anda untuk Prometheus.

Untuk menambahkan sumber data Prometheus di server Grafana Anda

1. Buka konsol Grafana.
2. Di bawah Konfigurasi, pilih Sumber data.
3. Pilih Tambahkan sumber data.
4. Pilih Prometheus.
5. Untuk URL HTTP, tentukan URL kueri Titik Akhir yang ditampilkan di halaman detail ruang kerja di konsol Amazon Managed Service for Prometheus.
6. Di URL HTTP yang baru saja Anda tentukan, hapus `/api/v1/query` string yang ditambahkan ke URL, karena sumber data Prometheus akan secara otomatis menambahkannya.

   URL yang benar akan terlihat mirip dengan https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-1234a5b6-78cd-901e-2fgh-3i45j6k178l9.
7. Di bawah Auth, pilih sakelar untuk SiGv4 Auth untuk mengaktifkannya.
8. Anda dapat mengonfigurasi otorisasi SigV4 dengan menentukan kredensyal jangka panjang Anda secara langsung di Grafana, atau dengan menggunakan rantai penyedia default. Menentukan kredensi jangka panjang Anda secara langsung membuat Anda memulai lebih cepat, dan langkah-langkah berikut memberikan instruksi tersebut terlebih dahulu. Setelah Anda lebih terbiasa menggunakan Grafana dengan Amazon Managed Service untuk Prometheus, kami sarankan Anda menggunakan rantai penyedia default, karena memberikan fleksibilitas dan keamanan yang lebih baik. Untuk informasi selengkapnya tentang menyiapkan rantai penyedia default, lihat Menentukan Kredensyal.
   - Untuk menggunakan kredensi jangka panjang Anda secara langsung, lakukan hal berikut:
     a. Di bawah Detail Auth SiGv4, untuk Penyedia Otentikasi pilih Kunci Akses & rahasia.
     b. Untuk ID Kunci Akses, masukkan ID kunci AWS akses Anda.

c. Untuk Kunci Akses Rahasia, masukkan kunci akses AWS rahasia Anda.

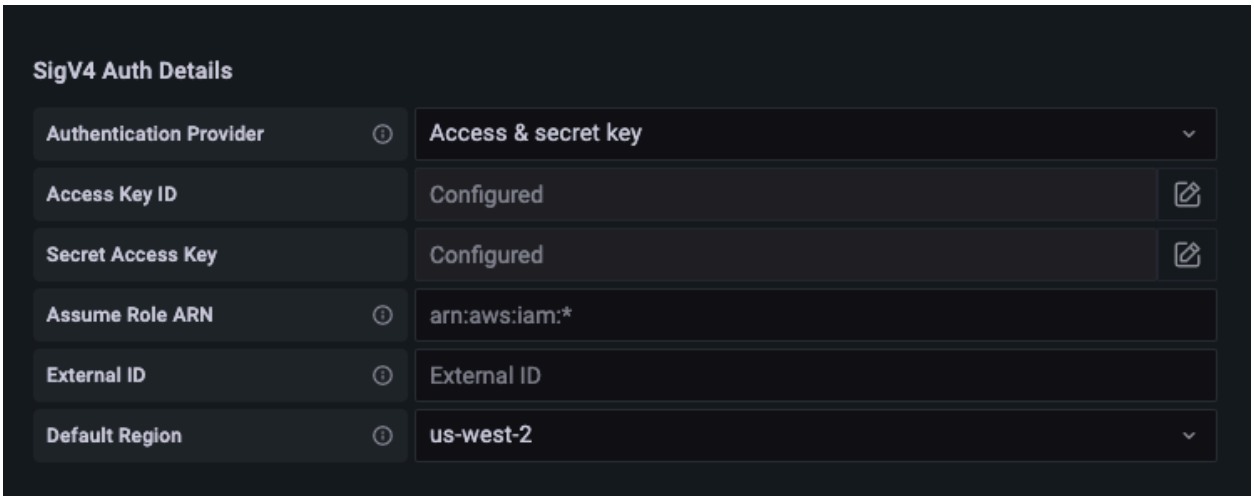
d. Biarkan kolom Assume Role ARN dan External ID kosong.

e. Untuk Wilayah Default, pilih Wilayah Layanan Terkelola Amazon untuk ruang kerja Prometheus. Wilayah ini harus cocok dengan Wilayah yang terdapat dalam URL yang Anda cantumkan di langkah 5.

f. Pilih Simpan & Uji.

Anda akan melihat pesan berikut: Sumber data berfungsi

Screenshot berikut menunjukkan tombol Access, Secret key SiGv4 pengaturan detail autentikasi.



- Untuk menggunakan rantai penyedia default sebagai gantinya (direkomendasikan untuk lingkungan produksi), lakukan hal berikut:

a. Di bawah Detail Auth SiGv4, untuk Penyedia Otentikasi pilih SDK Default. AWS

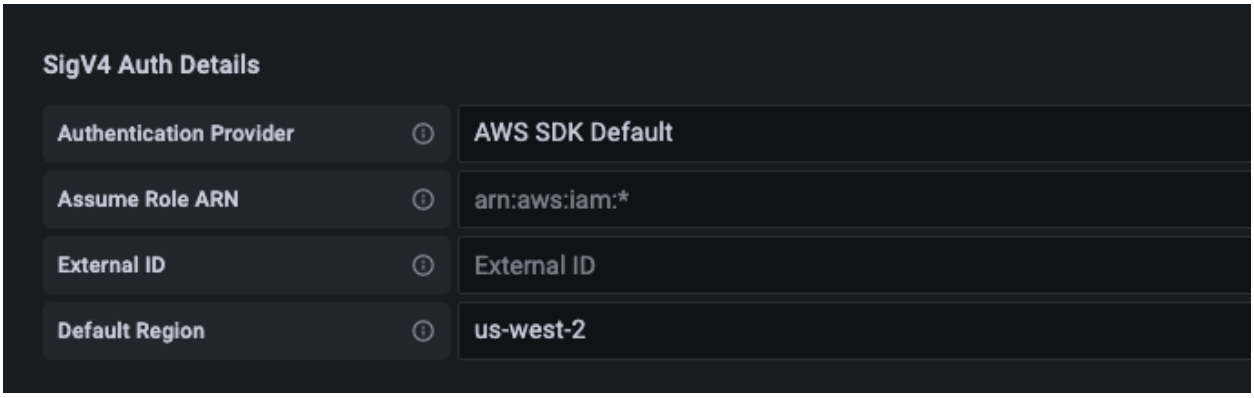
b. Biarkan kolom Assume Role ARN dan External ID kosong.

c. Untuk Wilayah Default, pilih Wilayah Layanan Terkelola Amazon untuk ruang kerja Prometheus. Wilayah ini harus cocok dengan Wilayah yang terdapat dalam URL yang Anda cantumkan di langkah 5.

d. Pilih Simpan & Uji.

Anda akan melihat pesan berikut: Sumber data berfungsi

Tangkapan layar berikut menunjukkan pengaturan detail autentikasi SiGv4 default SDK.



9. Uji kueri PromQL terhadap sumber data baru:

   a. Pilih Jelajahi.

   b. Jalankan contoh kueri PromQL seperti:

```
prometheus_tsdb_head_series
```

Pemecahan masalah jika Simpan & Uji tidak berfungsi

Pada prosedur sebelumnya, jika Anda melihat kesalahan saat memilih Simpan & Uji, periksa yang berikut ini.

Kesalahan HTTP Tidak Ditemukan

Pastikan bahwa ID ruang kerja di URL sudah benar.

Kesalahan HTTP Terlarang

Kesalahan ini berarti bahwa kredensialnya tidak valid. Periksa hal-hal berikut:

- Periksa apakah Wilayah yang ditentukan di Wilayah Default sudah benar.
- Periksa kredensi Anda untuk kesalahan ketik.
- Pastikan bahwa kredensi yang Anda gunakan memiliki AmazonPrometheusQueryAccesskebijakan. Untuk informasi selengkapnya, lihat Izin dan kebijakan IAM.
- Pastikan kredensi yang Anda gunakan memiliki akses ke Layanan Terkelola Amazon untuk ruang kerja Prometheus ini.

Kesalahan HTTP Gateway Buruk

Lihat log server Grafana untuk memecahkan masalah kesalahan ini. Untuk informasi selengkapnya, lihat Pemecahan Masalah di dokumentasi Grafana.

Jika Anda melihatnya**`Error http: proxy error: NoCredentialProviders: no valid providers in chain`**, rantai penyedia kredensyal default tidak dapat menemukan AWS kredensi yang valid untuk digunakan. Pastikan Anda telah menyiapkan kredensyal Anda seperti yang didokumentasikan dalam Menentukan Kredensyal. Jika Anda ingin menggunakan konfigurasi bersama, pastikan bahwa `AWS_SDK_LOAD_CONFIG` lingkungan diatur `ketrue`.

Kueri menggunakan Grafana yang berjalan di kluster Amazon EKS

Layanan Terkelola Amazon untuk Prometheus mendukung penggunaan Grafana versi 7.3.5 dan yang lebih baru untuk menanyakan metrik di Layanan Terkelola Amazon untuk ruang kerja Prometheus. Versi 7.3.5 dan yang lebih baru mencakup dukungan untuk otentikasi AWS Signature Version 4 (SigV4).

Untuk mengatur Grafana agar berfungsi dengan Layanan Terkelola Amazon untuk Prometheus, Anda harus masuk ke akun yang memiliki AmazonPrometheusQueryAccesskebijakan atau,,, dan izin. `aps:QueryMetrics` `aps:GetMetricMetadata` `aps:GetSeries` `aps:GetLabels` Untuk informasi selengkapnya, lihat Izin dan kebijakan IAM.

Mengatur AWS SiGv4

Grafana telah menambahkan fitur baru untuk mendukung otentikasi AWS Signature Version 4 (SiGv4). Untuk informasi selengkapnya, lihat proses penandatanganan Signature Version 4. Fitur ini tidak diaktifkan secara default di server Grafana. Instruksi berikut untuk mengaktifkan fitur ini mengasumsikan bahwa Anda menggunakan Helm untuk menyebarkan Grafana pada klaster Kubernetes.

Untuk mengaktifkan SiGv4 di server Grafana 7.3.5 atau yang lebih baru

1. Buat file pembaruan baru untuk mengganti konfigurasi Grafana Anda, dan beri nama. `amp_query_override_values.yaml`
2. Masukkan konten berikut ke dalam file, dan simpan file. Ganti *`account-id`* dengan ID AWS akun tempat server Grafana berjalan.

```
serviceAccount:
    name: "amp-iamproxy-query-service-account"
    annotations:
```

```
        eks.amazonaws.com/role-arn: "arn:aws:iam::account-id:role/amp-iamproxy-
query-role"
grafana.ini:
  auth:
    sigv4_auth_enabled: true
```

Dalam konten file YAMM `amp-iamproxy-query-role` itu, adalah nama peran yang akan Anda buat di bagian berikutnya,Menyiapkan peran IAM untuk akun layanan. Anda dapat mengganti peran ini dengan nama peran Anda sendiri jika Anda sudah memiliki peran yang dibuat untuk menanyakan ruang kerja Anda.

Anda akan menggunakan file ini nanti, diTingkatkan server Grafana menggunakan Helm.

Menyiapkan peran IAM untuk akun layanan

Jika Anda menggunakan server Grafana di kluster Amazon EKS, sebaiknya gunakan peran IAM untuk akun layanan, juga dikenal sebagai peran layanan, untuk kontrol akses Anda. Ketika Anda melakukan ini untuk mengaitkan peran IAM dengan akun layanan Kubernetes, akun layanan kemudian dapat memberikan AWS izin ke container di pod mana pun yang menggunakan akun layanan tersebut. Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan.

Jika Anda belum menyiapkan peran layanan ini untuk kueri, ikuti petunjuk di Menyiapkan peran IAM untuk akun layanan untuk kueri metrik untuk mengatur peran.

Anda kemudian perlu menambahkan akun layanan Grafana dalam kondisi hubungan kepercayaan.

Untuk menambahkan akun layanan Grafana dalam kondisi hubungan kepercayaan

1. Dari jendela terminal, tentukan namespace dan nama akun layanan untuk server Grafana Anda. Misalnya, Anda dapat menggunakan perintah berikut.

   ```
   kubectl get serviceaccounts -n grafana_namespace
   ```

2. Di konsol Amazon EKS, buka peran IAM untuk akun layanan yang terkait dengan kluster EKS.
3. Pilih Edit trust relationship (Edit Hubungan Kepercayaan).
4. Perbarui Kondisi untuk menyertakan namespace Grafana dan nama akun layanan Grafana yang Anda temukan di output perintah di langkah 1. Berikut sebuah contoh.

   ```
   {
   ```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::account-id:oidc-provider/
oidc.eks.aws_region.amazonaws.com/id/openid"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region.amazonaws.com/id/openid:sub": [
            "system:serviceaccount:aws-amp:amp-iamproxy-query-service-account",
            "system:serviceaccount:grafana-namespace:grafana-service-account-name"
          ]
        }
      }
    }
  ]
}
```

5. Pilih Perbarui Kebijakan Kepercayaan.

Tingkatkan server Grafana menggunakan Helm

Langkah ini meningkatkan server Grafana untuk menggunakan entri yang Anda tambahkan ke file di `amp_query_override_values.yaml` bagian sebelumnya.

Jalankan perintah berikut. Untuk informasi lebih lanjut tentang bagan Helm untuk Grafana, lihat Grafik Helm Kubernetes Komunitas Grafana.

```
helm repo add grafana https://grafana.github.io/helm-charts
```

```
helm upgrade --install grafana grafana/grafana -n grafana_namespace -f ./
amp_query_override_values.yaml
```

Tambahkan sumber data Prometheus di Grafana

Langkah-langkah berikut menjelaskan cara menyiapkan sumber data Prometheus di Grafana untuk menanyakan metrik Layanan Terkelola Amazon Anda untuk Prometheus.

Untuk menambahkan sumber data Prometheus di server Grafana Anda

1. Buka konsol Grafana.
2. Di bawah Konfigurasi, pilih Sumber data.
3. Pilih Tambahkan sumber data.
4. Pilih Prometheus.
5. Untuk URL HTTP, tentukan URL kueri Titik Akhir yang ditampilkan di halaman detail ruang kerja di konsol Amazon Managed Service for Prometheus.
6. Di URL HTTP yang baru saja Anda tentukan, hapus `/api/v1/query` string yang ditambahkan ke URL, karena sumber data Prometheus akan secara otomatis menambahkannya.
7. Di bawah Auth, pilih sakelar untuk SiGv4 Auth untuk mengaktifkannya.

   Biarkan kolom Assume Role ARN dan External ID kosong. Kemudian untuk Wilayah Default, pilih Wilayah tempat Amazon Managed Service untuk ruang kerja Prometheus berada.
8. Pilih Simpan & Uji.

   Anda akan melihat pesan berikut: Sumber data berfungsi
9. Uji kueri PromQL terhadap sumber data baru:

   a. Pilih Jelajahi.

   b. Jalankan contoh kueri PromQL seperti:

   ```
   prometheus_tsdb_head_series
   ```

Kueri menggunakan API yang kompatibel dengan Prometheus

Meskipun menggunakan alat seperti Amazon Managed Grafana adalah cara termudah untuk melihat dan menanyakan metrik Anda, Amazon Managed Service for Prometheus juga mendukung beberapa API yang kompatibel dengan Prometheus yang dapat Anda gunakan untuk menanyakan metrik Anda. Untuk informasi selengkapnya tentang semua API yang kompatibel dengan Prometheus yang tersedia, lihat. API yang kompatibel dengan Prometheus

Saat Anda menggunakan API ini untuk menanyakan metrik Anda, permintaan harus ditandatangani dengan proses penandatanganan Versi AWS Tanda Tangan 4. Anda dapat mengatur AWSSignature Version 4 untuk menyederhanakan proses penandatanganan. Untuk informasi selengkapnya, lihat aws-sigv4-proxy.

Penandatanganan melalui proxy AWS SiGv4 dapat dilakukan dengan menggunakan. `awscurl` Topik berikut Menggunakan awscurl untuk menanyakan API yang kompatibel dengan Prometheus memandu Anda menggunakan untuk menyiapkan Sigv4. `awscurl` AWS

Menggunakan awscurl untuk menanyakan API yang kompatibel dengan Prometheus

Permintaan API untuk Amazon Managed Service untuk Prometheus harus ditandatangani dengan SigV4. Anda dapat menggunakan awscurl untuk menyederhanakan proses kueri.

Untuk menginstal`awscurl`, Anda harus menginstal manajer paket Python 3 dan pip.

Pada instance berbasis Linux, perintah berikut diinstal`awscurl`.

```
$ pip3 install awscurl
```

Pada mesin macOS, perintah berikut diinstal. `awscurl`

```
$ brew install awscurl
```

Contoh berikut adalah contoh `awscurl` query. Ganti input *`Region`*, *`workspace-ID`*, dan *`QUERY`* dengan nilai yang sesuai untuk kasus penggunaan Anda:

```
# Define the Prometheus query endpoint URL. This can be found in the Amazon Managed
 Service for Prometheus console page
# under the respective workspace.

$ export AMP_QUERY_ENDPOINT=https://aps-workspaces.Region.amazonaws.com/
workspaces/Workspace-id/api/v1/query

# credentials are infered from the default profile
$ awscurl -X POST --region Region \
                  --service aps "${AMP_QUERY_ENDPOINT}" -d 'query=QUERY  --header
 'Content-Type: application/x-www-form-urlencoded'
```

> **Note**
>
> String kueri Anda harus dikodekan url.

Untuk kueri seperti`query=up`, Anda bisa mendapatkan hasil seperti:

```
{
  "status": "success",
  "data": {
    "resultType": "vector",
    "result": [
      {
        "metric": {
          "__name__": "up",
          "instance": "localhost:9090",
          "job": "prometheus",
          "monitor": "monitor"
        },
        "value": [
          1652452637.636,
          "1"
        ]
      },
    ]
  }
}
```

`awscurl`Untuk menandatangani permintaan yang diberikan, Anda harus melewati kredensyal yang valid dengan salah satu cara berikut:

- Berikan ID kunci akses dan kunci Rahasia untuk peran IAM. Anda dapat menemukan kunci akses dan kunci rahasia untuk peran di https://console.aws.amazon.com/iam/.

  Sebagai contoh:

  ```
  $ export AMP_QUERY_ENDPOINT=https://aps-workspaces.Region.amazonaws.com/
  workspaces/Workspace_id/api/v1/query

  $ awscurl -X POST --region <Region> \
                    --access_key <ACCESS_KEY> \
                    --secret_key <SECRET_KEY> \
                    --service aps "$AMP_QUERY_ENDPOINT?query=<QUERY>"
  ```

- Referensi file konfigurasi yang disimpan dalam `/aws/config` file `.aws/credentials` and. Anda juga dapat memilih untuk menentukan nama profil yang akan digunakan. Jika tidak ditentukan, `default` file akan digunakan. Sebagai contoh:

```
$ export AMP_QUERY_ENDPOINT=https://aps-workspaces.<Region>.amazonaws.com/workspaces/
<Workspace_ID>/api/v1/query
$ awscurl -X POST --region <Region> \
                  --profile <PROFILE_NAME>
                  --service aps "$AMP_QUERY_ENDPOINT?query=<QUERY>"
```

- Gunakan profil instans yang terkait dengan instans EC2.

Menjalankan permintaan kueri menggunakan wadah awscurl

Saat menginstal versi Python yang berbeda dan dependensi terkait tidak layak, wadah dapat digunakan untuk mengemas aplikasi dan dependensinya. `awscurl` Contoh berikut menggunakan runtime Docker untuk menerapkan`awscurl`, tetapi runtime dan gambar yang sesuai dengan OCI akan berfungsi.

```
$ docker pull okigan/awscurl
$ export AMP_QUERY_ENDPOINT=https://aps-workspaces.Region.amazonaws.com/
workspaces/Workspace_id/api/v1/query
$ docker run --rm -it okigan/awscurl --access_key $AWS_ACCESS_KEY_ID  --secret_key
 $AWS_SECRET_ACCESS_KEY \ --region Region --service aps "$AMP_QUERY_ENDPOINT?
query=QUERY"
```

Kueri informasi statistik dalam respons API kueri

Harga kueri didasarkan pada jumlah sampel kueri yang diproses dalam sebulan dari kueri yang dieksekusi. Respons kueri untuk `queryRange` API `query` atau menyertakan data statistik tentang sampel kueri yang diproses. Ketika parameter query `stats=all` dikirim dalam permintaan, `samples` objek dibuat dalam `stats` objek dan `stats` data dikembalikan dalam respon.

`samples`Objek terdiri dari atribut berikut:

Atribut	Deskripsi
`totalQueryableSamples`	Jumlah total sampel kueri yang diproses. Ini adalah informasi yang akan digunakan untuk penagihan.

Atribut	Deskripsi
`totalQueryableSamplesPerStep`	Jumlah sampel kueri yang diproses per setiap langkah. Ini disusun sebagai array array dengan stempel waktu dalam epoch dan jumlah sampel yang dimuat pada langkah tertentu.

Contoh permintaan dan tanggapan yang menyertakan `stats` informasi dalam tanggapan adalah sebagai berikut:

Contoh untuk`query`:

DAPATKAN

```
endpoint/api/v1/query?query=up&time=1652382537&stats=all
```

Respons

```
{
    "status": "success",
    "data": {
        "resultType": "vector",
        "result": [
            {
                "metric": {
                    "__name__": "up",
                    "instance": "localhost:9090",
                    "job": "prometheus"
                },
                "value": [
                    1652382537,
                    "1"
                ]
            }
        ],
        "stats": {
            "timings": {
                "evalTotalTime": 0.00453349,
                "resultSortTime": 0,
                "queryPreparationTime": 0.000019363,
                "innerEvalTime": 0.004508405,
```

```
                "execQueueTime": 0.000008786,
                "execTotalTime": 0.004554219
            },
            "samples": {
                "totalQueryableSamples": 1,
                "totalQueryableSamplesPerStep": [
                    [
                        1652382537,
                        1
                    ]
                ]
            }
        }
    }
}
```

Contoh untuk`queryRange`:

DAPATKAN

```
endpoint/api/v1/query_range?query=sum+%28rate+%28go_gc_duration_seconds_count%5B1m%5D
%29%29&start=1652382537&end=1652384705&step=1000&stats=all
```

Respons

```
{
    "status": "success",
    "data": {
        "resultType": "matrix",
        "result": [
            {
                "metric": {},
                "values": [
                    [
                        1652383000,
                        "0"
                    ],
                    [
                        1652384000,
                        "0"
                    ]
                ]
            }
```

```
        ],
        "stats": {
            "samples": {
                "totalQueryableSamples": 8,
                "totalQueryableSamplesPerStep": [
                    [
                        1652382000,
                        0
                    ],
                    [
                        1652383000,
                        4
                    ],
                    [
                        1652384000,
                        4
                    ]
                ]
            }
        }
    }
}
```

Merekam aturan dan aturan peringatan

Amazon Managed Service untuk Prometheus mendukung dua jenisaturanyang dievaluasi secara berkala:

- Aturan perekamanmemungkinkan Anda untuk menghitung ulang ekspresi yang sering dibutuhkan atau mahal secara komputasi dan menyimpan hasilnya sebagai rangkaian waktu baru. Menanyakan hasil yang telah dihitung sebelumnya seringkali jauh lebih cepat daripada menjalankan ekspresi asli setiap kali diperlukan.
- Aturan peringatanmemungkinkan Anda untuk menentukan kondisi peringatan berdasarkan PromQL dan ambang batas. Saat aturan memicu ambang batas, pemberitahuan dikirim ke manajer peringatan, yang meneruskan notifikasi ke hilir ke penerima seperti Amazon Simple Notification Service.

Untuk menggunakan aturan di Amazon Managed Service untuk Prometheus, Anda membuat satu atau beberapa file aturan YAMM yang menentukan aturan. File aturan Amazon Managed Service untuk Prometheus memiliki format yang sama dengan file aturan di Prometheus mandiri. Untuk informasi selengkapnya, lihatMendefinisikan aturan PerekamandanAturan peringatandalam dokumentasi Prometheus.

Anda dapat memiliki beberapa file aturan di ruang kerja. Setiap file aturan terpisah terkandung dalam file terpisahnamespace. Memiliki beberapa file aturan memungkinkan Anda mengimpor file aturan Prometheus yang ada ke ruang kerja tanpa harus mengubah atau menggabungkannya. Ruang nama grup aturan yang berbeda juga dapat memiliki tag yang berbeda.

Urutan aturan

Dalam file aturan, aturan terkandung di dalamaturan kelompok. Aturan dalam satu grup aturan dalam file aturan selalu dievaluasi secara berurutan dari atas ke bawah. Oleh karena itu, dalam aturan perekaman, hasil dari satu aturan perekaman dapat digunakan dalam perhitungan aturan perekaman nanti atau dalam aturan peringatan dalam kelompok aturan yang sama. Namun, karena Anda tidak dapat menentukan urutan untuk menjalankan file aturan terpisah, Anda tidak dapat menggunakan hasil dari satu aturan perekaman untuk menghitung aturan dalam grup aturan yang berbeda atau file aturan yang berbeda.

Topik

- Izin IAM yang diperlukan

- Membuat file aturan
- Mengunggah file konfigurasi aturan ke Amazon Managed Service untuk Prometheus
- Mengedit file konfigurasi aturan
- Pemecahan Masalah

Izin IAM yang diperlukan

Anda harus memberi pengguna izin untuk menggunakan aturan di Amazon Managed Service untuk Prometheus. Buat sebuahAWS Identity and Access Management(IAM) kebijakan dengan izin berikut, dan tetapkan kebijakan untuk pengguna, grup, atau peran Anda.

> **Note**
>
> Untuk informasi selengkapnya tentang IAM, lihat Identity and Access Management untuk Amazon Managed Service untuk Prometheus.

Kebijakan untuk memberikan akses ke aturan penggunaan

Kebijakan berikut memberikan akses untuk menggunakan aturan untuk semua sumber daya di akun Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps: CreateRuleGroupsNamespace",
                "aps: ListRuleGroupsNamespaces",
                "aps: DescribeRuleGroupsNamespace",
                "aps: PutRuleGroupsNamespace",
                "aps: DeleteRuleGroupsNamespace",
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan untuk memberikan akses ke hanya satu namespace

Anda juga dapat membuat kebijakan yang hanya memberikan akses ke kebijakan tertentu. Kebijakan sampel berikut hanya memberikan akses ke`RuleGroupNamespace`editentukan. Untuk menggunakan kebijakan ini, ganti*<account>*,*<region>*,*<workspace-id>*, dan*<namespace-name>*dengan nilai yang sesuai untuk akun Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:ListRules",
                "aps:ListTagsForResource",
                "aps:GetLabels",
                "aps:CreateRuleGroupsNamespace",
                "aps:ListRuleGroupsNamespaces",
                "aps:DescribeRuleGroupsNamespace",
                "aps:PutRuleGroupsNamespace",
                "aps:DeleteRuleGroupsNamespace"
            ],
            "Resource": [
                "arn:aws:aps:*:<account>:workspace/*",
                "arn:aws:aps:<region>:<account>:rulegroupnamespace/<workspace-
id>/<namespace-name>"
            ]
        }
    ]
}
```

Membuat file aturan

Untuk menggunakan aturan di Amazon Managed Service untuk Prometheus, Anda membuat file aturan yang menentukan aturan. File aturan Amazon Managed Service untuk Prometheus memiliki format yang sama dengan file aturan di Prometheus mandiri. Untuk informasi selengkapnya, lihatMendefinisikan aturan PerekamandanAturan peringatan.

Berikut ini adalah contoh dasar dari file aturan:

```
groups:
```

```
  - name: test
    rules:
    - record: metric:recording_rule
      expr: avg(rate(container_cpu_usage_seconds_total[5m]))
  - name: alert-test
    rules:
    - alert: metric:alerting_rule
      expr: avg(rate(container_cpu_usage_seconds_total[5m])) > 0
      for: 2m
```

Untuk contoh aturan peringatan lainnya, lihatContoh aturan peringatan.

Mengunggah file konfigurasi aturan ke Amazon Managed Service untuk Prometheus

Sekarang Anda harus mengunggah file konfigurasi aturan ini ke Amazon Managed Service untuk Prometheus. Anda dapat menggunakan salah satu dari konsol atauAWS CLIuntuk mengunggahnya.

Untuk menggunakan Amazon Managed Service for Prometheus console untuk mengunggah konfigurasi aturan dan membuat namespace

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Di sudut kiri atas halaman, pilih ikon menu, lalu pilihSemua ruang kerja.
3. Pilih ID ruang kerja ruang kerja, lalu pilihManajemen aturantab.
4. PilihTambahkan namespace.
5. PilihPilih berkas, dan pilih file definisi aturan.
6. (Opsional) Untuk menambahkan tag ke namespace, pilihTambahkan tag baru.

   Kemudian, untukKunci, masukkan nama untuk tag. Anda dapat menambahkan nilai opsional untuk tag di Nilai.

   Untuk menambahkan tag lain, pilihTambahkan tag baru.
7. Pilih Continue (Lanjutkan). Amazon Managed Service untuk Prometheus membuat namespace baru dengan nama yang sama dengan file aturan yang Anda pilih.

Untuk menggunakanAWS CLIuntuk mengunggah konfigurasi manajer peringatan ke ruang kerja di namespace baru

1. Base64 menyandikan konten file pengelola peringatan Anda. Di Linux, Anda dapat menggunakan perintah berikut:

   ```
   base64 input-file output-file
   ```

   Di macOS, Anda dapat menggunakan perintah berikut:

   ```
   openssl base64 input-file output-file
   ```

2. Masukkan salah satu perintah berikut untuk membuat namespace dan mengompasi file.

   PadaAWS CLIversi 2, masukkan:

   ```
   aws amp create-rule-groups-namespace --data file://path_to_base_64_output_file --
   name namespace-name  --workspace-id my-workspace-id --region region
   ```

   PadaAWS CLIversi 1, masukkan:

   ```
   aws amp create-rule-groups-namespace --data fileb://path_to_base_64_output_file --
   name namespace-name  --workspace-id my-workspace-id --region region
   ```

3. Dibutuhkan beberapa detik agar konfigurasi manajer peringatan Anda menjadi aktif. Untuk memeriksa status berikut, masukkan perintah berikut:

   ```
   aws amp describe-rule-groups-namespace --workspace-id workspace_id --
   name namespace-name --region region
   ```

   Jika`status`adalah`ACTIVE`, file aturan Anda telah berlaku.

Mengedit file konfigurasi aturan

Anda tidak dapat mengedit file konfigurasi aturan secara langsung di konsol. Sebagai gantinya, Anda mengunggah file aturan baru untuk menggantinya. Secara opsional, Anda dapat mengunduh file saat ini, mengeditnya di editor teks, lalu mengunggah versi baru.

Untuk menggunakan Amazon Managed Service for Prometheus console untuk mengedit konfigurasi aturan

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Di sudut kiri atas halaman, pilih ikon menu, lalu pilihSemua ruang kerja.
3. Pilih ID ruang kerja ruang kerja, lalu pilihManajemen aturantab.
4. (Opsional) Jika Anda ingin memulai dengan mengedit file konfigurasi aturan saat ini, pilihUnduhatauSalin.
5. Saat file aturan baru Anda siap, pilihGanti.
6. PilihPilih berkas, pilih file definisi aturan baru, dan pilihLanjutkan.

Untuk menggunakanAWS CLIuntuk mengedit file konfigurasi aturan

1. Base64 menyandikan isi file aturan Anda. Di Linux, Anda dapat menggunakan perintah berikut:

   ```
   base64 input-file output-file
   ```

   Di macOS, Anda dapat menggunakan perintah berikut:

   ```
   openssl base64 input-file output-file
   ```

2. Masukkan salah satu perintah berikut untuk mengompasi file baru.

   PadaAWS CLIversi 2, masukkan:

   ```
   aws amp put-rule-groups-namespace --data file://path_to_base_64_output_file --
   name namespace-name  --workspace-id my-workspace-id --region region
   ```

   PadaAWS CLIversi 1, masukkan:

   ```
   aws amp put-rule-groups-namespace --data fileb://path_to_base_64_output_file --
   name namespace-name  --workspace-id my-workspace-id --region region
   ```

3. Dibutuhkan beberapa detik agar file aturan Anda menjadi aktif. Untuk memeriksa status berikut, masukkan perintah berikut:

```
aws amp describe-rule-groups-namespace --workspace-id workspace_id --
name namespace-name --region region
```

Jika`status`adalah`ACTIVE`, file aturan Anda telah berlaku. Sampai saat itu, versi sebelumnya dari file aturan ini masih aktif.

Pemecahan Masalah

MenggunakanCloudWatch Log, Anda dapat memecahkan masalah terkait Manajer Peringatan dan Penggaris. Bagian ini berisi topik pemecahan masalah terkait penggaris.

Ketika log berisi kesalahan kegagalan penggaris berikut

```
{
    "workspaceId": "ws-12345c67-89c0-4d12-345b-f14db70f7a99",
    "message": {
        "log": "Evaluating rule failed, name=failure,
 group=canary_long_running_vl_namespace, namespace=canary_long_running_vl_namespace,
 err=found duplicate series for the match group {dimension1=\\\"1\\\"} on the right
 hand-side of the operation: [{__name__=\\\"fake_metric2\\\", dimension1=\\\"1\\
\", dimension2=\\\"b\\\"}, {__name__=\\\"fake_metric2\\\", dimension1=\\\"1\\\",
 dimension2=\\\"a\\\"}];many-to-many matching not allowed: matching labels must be
 unique on one side",
        "level": "ERROR",
        "name": "failure",
        "group": "canary_long_running_vl_namespace",
        "namespace": "canary_long_running_vl_namespace"
    },
    "component": "ruler"
}
```

Ini berarti bahwa beberapa kesalahan terjadi saat menjalankan aturan.

Tindakan yang harus dilakukan

Gunakan pesan kesalahan untuk memecahkan masalah eksekusi aturan.

Manajer Peringatan

Saat aturan peringatan yang dijalankan Amazon Managed Service untuk Prometheus diaktifkan, manajer peringatan menangani peringatan yang dikirim. Ini menghapus duplikasi, mengelompokkan, dan merutekan peringatan ke penerima hilir. Layanan Terkelola Amazon untuk Prometheus hanya mendukung Layanan Pemberitahuan Sederhana Amazon sebagai penerima, dan dapat merutekan pesan ke topik Amazon SNS di akun yang sama. Anda juga dapat menggunakan manajer peringatan untuk membungkam dan menghambat peringatan.

Manajer peringatan menyediakan fungsionalitas yang mirip dengan Alertmanager di Prometheus.

Anda dapat menggunakan file konfigurasi manajer peringatan untuk hal-hal berikut:

- Pengelompokan — Pengelompokan mengumpulkan peringatan serupa menjadi satu pemberitahuan. Ini sangat berguna selama pemadaman yang lebih besar ketika banyak sistem gagal sekaligus dan ratusan peringatan mungkin menyala secara bersamaan. Misalnya, kegagalan jaringan menyebabkan banyak node Anda gagal pada saat yang bersamaan. Jika jenis peringatan ini dikelompokkan, manajer peringatan mengirimi Anda satu pemberitahuan.

  Pengelompokan peringatan dan waktu untuk pemberitahuan yang dikelompokkan dikonfigurasi oleh pohon perutean di file konfigurasi manajer peringatan. Untuk informasi lebih lanjut, lihathttps://prometheus.io/docs/alerting/latest/configuration/#route<route>.
- Penghambatan — Penghambatan menekan pemberitahuan untuk peringatan tertentu jika peringatan tertentu lainnya sudah menyala. Misalnya, jika peringatan diaktifkan tentang klaster yang tidak dapat dijangkau, Anda dapat mengonfigurasi pengelola peringatan untuk membisukan semua peringatan lain mengenai kluster ini. Ini mencegah pemberitahuan untuk ratusan atau ribuan peringatan penembakan yang tidak terkait dengan masalah sebenarnya. <inhibit_rule>Untuk informasi lebih lanjut tentang cara menulis aturan penghambatan, lihathttps://prometheus.io/docs/alerting/latest/configuration/#inhibit_rule.
- Keheningan — Membungkam peringatan bisu untuk waktu tertentu, seperti selama jendela pemeliharaan. Peringatan yang masuk diperiksa apakah mereka cocok dengan semua persamaan atau pencocokan ekspresi reguler dari keheningan aktif. Jika ya, tidak ada pemberitahuan yang dikirim untuk peringatan itu.

  Untuk membuat keheningan, Anda menggunakan `PutAlertManagerSilences` API. Untuk informasi selengkapnya, lihat PutAlertManagerSilences.

Templat Prometheus

Prometheus mandiri mendukung templating, menggunakan memisahkan file template. Template dapat menggunakan kondisional dan memformat data, antara lain.

Di Amazon Managed Service untuk Prometheus, Anda menempatkan template Anda di file konfigurasi manajer peringatan yang sama dengan konfigurasi manajer peringatan Anda.

Topik

Izin IAM yang diperlukan

Anda harus memberi pengguna izin untuk menggunakan aturan di Amazon Managed Service untuk Prometheus. Buat kebijakan AWS Identity and Access Management (IAM) dengan izin berikut, dan tetapkan kebijakan tersebut ke pengguna, grup, atau peran Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps: CreateAlertManagerDefinition",
                "aps: DescribeAlertManagerSilence",
                "aps: DescribeAlertManagerDefinition",
                "aps: PutAlertManagerDefinition",
                "aps: DeleteAlertManagerDefinition",
                "aps: ListAlerts",
                "aps: ListRules",
                "aps: ListAlertManagerReceivers",
                "aps: ListAlertManagerSilences",
                "aps: ListAlertManagerAlerts",
```

```
                "aps: ListAlertManagerAlertGroups",
                "aps: GetAlertManagerStatus",
                "aps: GetAlertManagerSilence",
                "aps: PutAlertManagerSilences",
                "aps: DeleteAlertManagerSilence",
                "aps: CreateAlertManagerAlerts"
            ],
            "Resource": "*"
        }
    ]
}
```

Membuat file konfigurasi manajer peringatan

Untuk menggunakan pengelola peringatan dan template di Amazon Managed Service untuk Prometheus, Anda membuat file YAMM konfigurasi manajer peringatan. Layanan Terkelola Amazon untuk file manajer peringatan Prometheus memiliki dua bagian utama:

- `template_files:`berisi template yang digunakan untuk pesan yang dikirim oleh penerima. Untuk informasi selengkapnya, lihat Referensi Template dan Contoh Template di dokumentasi Prometheus.
- `alertmanager_config:`berisi konfigurasi manajer peringatan. Ini menggunakan struktur yang sama dengan file konfigurasi manajer peringatan di Prometheus mandiri. Untuk informasi selengkapnya, lihat Konfigurasi dalam dokumentasi Alertmanager.

> **Note**
>
> `repeat_interval`Konfigurasi yang dijelaskan dalam dokumentasi Prometheus di atas memiliki batasan tambahan di Amazon Managed Service untuk Prometheus. Nilai maksimum yang diizinkan adalah lima hari. Jika Anda mengaturnya lebih dari lima hari, itu akan diperlakukan sebagai lima hari dan pemberitahuan akan dikirim lagi setelah periode lima hari berlalu.

Di Amazon Managed Service untuk Prometheus, file konfigurasi manajer peringatan Anda harus memiliki semua konten konfigurasi manajer peringatan Anda di dalam kunci di root `alertmanager_config` file YAMB.

Berikut ini adalah contoh dasar file konfigurasi manajer peringatan:

```
alertmanager_config: |
  route:
    receiver: 'default'
  receivers:
    - name: 'default'
      sns_configs:
      - topic_arn: arn:aws:sns:us-east-2:123456789012:My-Topic
        sigv4:
          region: us-east-2
        attributes:
          key: key1
          value: value1
```

Satu-satunya penerima yang saat ini didukung adalah Amazon Simple Notification Service (Amazon SNS). Jika Anda memiliki jenis penerima lain yang tercantum dalam konfigurasi, itu akan ditolak.

Berikut adalah contoh file konfigurasi manajer peringatan lain yang menggunakan `template_files` blok dan `alertmanager_config` blok.

```
template_files:
  default_template: |
    {{ define "sns.default.subject" }}[{{ .Status | toUpper }}{{ if eq .Status
 "firing" }}:{{ .Alerts.Firing | len }}{{ end }}]{{ end }}
    {{ define "__alertmanager" }}AlertManager{{ end }}
    {{ define "__alertmanagerURL" }}{{ .ExternalURL }}/#/alerts?receiver={{ .Receiver |
 urlquery }}{{ end }}
alertmanager_config: |
  global:
  templates:
    - 'default_template'
  route:
    receiver: default
  receivers:
    - name: 'default'
      sns_configs:
      - topic_arn: arn:aws:sns:us-east-2:accountid:My-Topic
        sigv4:
          region: us-east-2
        attributes:
          key: severity
          value: SEV2
```

Blok template Amazon SNS default

Konfigurasi Amazon SNS default menggunakan templat berikut kecuali jika Anda secara eksplisit menggantinya.

```
{{ define "sns.default.message" }}{{ .CommonAnnotations.SortedPairs.Values | join "
 " }}
  {{ if gt (len .Alerts.Firing) 0 -}}
  Alerts Firing:
    {{ template "__text_alert_list" .Alerts.Firing }}
  {{- end }}
  {{ if gt (len .Alerts.Resolved) 0 -}}
  Alerts Resolved:
    {{ template "__text_alert_list" .Alerts.Resolved }}
  {{- end }}
{{- end }}
```

Menyiapkan penerima peringatan

Satu-satunya penerima peringatan yang saat ini didukung di Amazon Managed Service untuk Prometheus adalah Amazon Simple Notification Service (Amazon SNS). Untuk informasi lebih lanjut, lihat Apa itu Amazon SNS? .

Topik

- (Opsional) Membuat topik Amazon SNS baru
- Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan ke topik Amazon SNS Anda
- Menentukan topik Amazon SNS Anda di file konfigurasi manajer peringatan
- (Opsional) Mengonfigurasi manajer peringatan untuk mengeluarkan JSON ke Amazon SNS
- (Opsional) Mengirim dari Amazon SNS ke tujuan lain
- Aturan validasi dan pemotongan pesan penerima SNS

(Opsional) Membuat topik Amazon SNS baru

Anda dapat menggunakan topik Amazon SNS yang ada atau membuat yang baru. Kami menyarankan Anda menggunakan topik tipe Standar, sehingga Anda dapat meneruskan peringatan dari topik ke email, SMS, atau HTTP.

Untuk membuat topik Amazon SNS baru untuk digunakan sebagai penerima manajer peringatan, ikuti langkah-langkah di Langkah 1: Buat topik. Pastikan untuk memilih Standar untuk jenis topik.

Jika Anda ingin menerima email setiap kali pesan dikirim ke topik Amazon SNS itu, ikuti langkah-langkah di Langkah 2: Buat langganan ke topik tersebut.

Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan ke topik Amazon SNS Anda

Anda harus memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan ke topik Amazon SNS Anda. Pernyataan kebijakan berikut mencakup `Condition` pernyataan untuk membantu mencegah masalah keamanan wakil yang membingungkan. `Condition`Pernyataan tersebut membatasi akses ke topik Amazon SNS untuk mengizinkan hanya operasi yang berasal dari akun khusus ini dan Layanan Terkelola Amazon untuk ruang kerja Prometheus. Untuk informasi lebih lanjut tentang masalah wakil yang membingungkan, lihatPencegahan confused deputy lintas layanan.

Untuk memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan ke topik Amazon SNS Anda

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Di panel navigasi, pilih Pengguna.
3. Pilih nama topik yang Anda gunakan dengan Amazon Managed Service untuk Prometheus.
4. Pilih Edit.
5. Pilih Kebijakan akses dan tambahkan pernyataan kebijakan berikut ke kebijakan yang ada.

```
{
    "Sid": "Allow_Publish_Alarms",
    "Effect": "Allow",
    "Principal": {
        "Service": "aps.amazonaws.com"
    },
    "Action": [
        "sns:Publish",
        "sns:GetTopicAttributes"
    ],
    "Condition": {
        "ArnEquals": {
            "aws:SourceArn": "workspace_ARN"
        },
        "StringEquals": {
```

```
            "AWS:SourceAccount": "account_id"
        }
    },
    "Resource": "arn:aws:sns:region:account_id:topic_name"
}
```

[Opsional] Jika topik SNS Anda diaktifkan enkripsi sisi layanan (SSE), Anda perlu menambahkan izin berikut ke kebijakan kunci KMS Anda di blok. `"Action"` Untuk informasi selengkapnya, lihat Izin AWS KMS untuk Topik SNS.

```
    kms:GenerateDataKey
    kms:Decrypt
```

6. Pilih Simpan perubahan.

> **Note**
>
> Secara default, Amazon SNS membuat kebijakan akses dengan kondisi aktif. `AWS:SourceOwner` Untuk informasi selengkapnya, lihat Kebijakan Akses SNS.

> **Note**
>
> IAM mengikuti aturan pertama kebijakan yang paling membatasi. Dalam topik SNS Anda, jika ada blok kebijakan yang lebih ketat daripada blok kebijakan Amazon SNS yang didokumentasikan, izin untuk kebijakan topik tidak diberikan. Untuk mengevaluasi kebijakan Anda dan mencari tahu apa yang telah diberikan, lihat Logika evaluasi kebijakan.

Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. Pada tahunAWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS sediakan alat yang

membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan global dalam kebijakan sumber daya untuk membatasi izin yang diberikan Layanan Terkelola Amazon untuk Prometheus ke Amazon SNS ke sumber daya. Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Nilai `aws:SourceArn` harus ARN dari Amazon Managed Service untuk ruang kerja Prometheus.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan kunci konteks kondisi `aws:SourceArn` global dengan ARN penuh sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Sebagai contoh, `arn:aws:`*`servicename`*`::`*`123456789012`*`:*`.

Kebijakan yang ditampilkan di Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan ke topik Amazon SNS Anda menunjukkan cara Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global di Layanan Terkelola Amazon untuk Prometheus untuk mencegah masalah deputi yang membingungkan.

Menentukan topik Amazon SNS Anda di file konfigurasi manajer peringatan

Sekarang, Anda dapat menambahkan penerima Amazon SNS Anda ke konfigurasi manajer peringatan Anda. Untuk melakukan ini, Anda harus mengetahui Nama Sumber Daya Amazon (ARN) dari topik Amazon SNS Anda.

Untuk informasi selengkapnya tentang konfigurasi penerima Amazon SNS, lihathttps://prometheus.io/docs/alerting/latest/configuration/#sns_configs <sns_configs>di dokumentasi konfigurasi Prometheus.

Properti yang tidak didukung

Amazon Managed Service untuk Prometheus mendukung Amazon SNS sebagai penerima peringatan. Namun, karena kendala layanan, tidak semua properti penerima Amazon SNS didukung. Properti berikut tidak diizinkan dalam file konfigurasi manajer peringatan Prometheus Layanan Terkelola Amazon untuk Prometheus:

- `api_url:`— Layanan Terkelola Amazon untuk Prometheus menetapkan untuk Anda, jadi properti `api_url` ini tidak diizinkan.

- `Http_config`— Properti ini memungkinkan Anda untuk mengatur proxy eksternal. Amazon Managed Service untuk Prometheus saat ini tidak mendukung fitur ini.

Selain itu, pengaturan SiGv4 diperlukan untuk memiliki properti Region. Tanpa properti Wilayah, Amazon Managed Service untuk Prometheus tidak memiliki informasi yang cukup untuk membuat permintaan otorisasi.

Untuk mengonfigurasi pengelola peringatan dengan topik Amazon SNS Anda sebagai penerima

1. Jika Anda menggunakan file konfigurasi manajer peringatan yang ada, buka di editor teks.
2. Jika ada penerima saat ini selain Amazon SNS di `receivers` blok, hapus. Anda dapat mengonfigurasi beberapa topik Amazon SNS menjadi penerima dengan menempatkannya di `sns_config` blok terpisah di dalam blok. `receivers`
3. Tambahkan blok YAMM berikut di dalam `receivers` bagian.

```
- name: name_of_receiver
  sns_configs:
    - sigv4:
        region: region
      topic_arn: ARN_of_SNS_topic
      subject: somesubject
      attributes:
        key: somekey
        value: somevalue
```

Jika a tidak `subject` ditentukan, secara default, subjek akan dihasilkan dengan template default dengan nama label dan nilai, yang dapat menghasilkan nilai yang terlalu panjang untuk SNS. Untuk mengubah templat yang diterapkan pada subjek, lihat (Opsional) Mengonfigurasi manajer peringatan untuk mengeluarkan JSON ke Amazon SNS di panduan ini.

Sekarang Anda harus mengunggah file konfigurasi manajer peringatan Anda ke Amazon Managed Service untuk Prometheus. Untuk informasi selengkapnya, lihat Mengunggah file konfigurasi pengelola peringatan Anda ke Amazon Managed Service untuk Prometheus.

(Opsional) Mengonfigurasi manajer peringatan untuk mengeluarkan JSON ke Amazon SNS

Anda dapat mengonfigurasi pengelola peringatan untuk mengirim peringatan dalam format JSON, sehingga dapat diproses di hilir dari Amazon SNS di atau di AWS Lambda titik akhir penerima webhook. Template default yang disertakan dengan Amazon Managed Service for Prometheus alert manager menampilkan payload pesan dalam format daftar teks, yang mungkin tidak mudah diuraikan. Alih-alih menggunakan template default, Anda dapat menentukan template kustom untuk menampilkan konten pesan di JSON, sehingga lebih mudah untuk mengurai dalam fungsi hilir.

Untuk menampilkan pesan dari manajer peringatan ke Amazon SNS dalam format JSON, perbarui konfigurasi manajer peringatan Anda untuk memuat kode berikut di dalam bagian root Andatemplate_files:

```
default_template: |
   {{ define "sns.default.message" }}{{ "{" }}"receiver": "{{ .Receiver }}","status":
 "{{ .Status }}","alerts": [{{ range $alertIndex, $alerts := .Alerts }}{{ if
 $alertIndex }}, {{ end }}{{ "{" }}"status": "{{ $alerts.Status }}"{{ if
 gt (len $alerts.Labels.SortedPairs) 0 -}},"labels": {{ "{" }}{{ range
 $index, $label := $alerts.Labels.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $label.Name }}": "{{ $label.Value }}"{{ end }}
{{ "}" }}{{- end }}{{ if gt (len $alerts.Annotations.SortedPairs )
 0 -}},"annotations": {{ "{" }}{{ range $index, $annotations :=
 $alerts.Annotations.SortedPairs }}{{ if $index }}, {{ end }}"{{ $annotations.Name }}":
 "{{ $annotations.Value }}"{{ end }}{{ "}" }}{{- end }},"startsAt":
 "{{ $alerts.StartsAt }}","endsAt": "{{ $alerts.EndsAt }}","generatorURL":
 "{{ $alerts.GeneratorURL }}","fingerprint": "{{ $alerts.Fingerprint }}"{{ "}" }}
{{ end }}]{{ if gt (len .GroupLabels) 0 -}},"groupLabels": {{ "{" }}{{ range
 $index, $groupLabels := .GroupLabels.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $groupLabels.Name }}": "{{ $groupLabels.Value }}"{{ end }}
{{ "}" }}{{- end }}{{ if gt (len .CommonLabels) 0 -}},"commonLabels": {{ "{" }}
{{ range $index, $commonLabels := .CommonLabels.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $commonLabels.Name }}": "{{ $commonLabels.Value }}"{{ end }}{{ "}" }}{{-
 end }}{{ if gt (len .CommonAnnotations) 0 -}},"commonAnnotations": {{ "{" }}{{ range
 $index, $commonAnnotations := .CommonAnnotations.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $commonAnnotations.Name }}": "{{ $commonAnnotations.Value }}"{{ end }}
{{ "}" }}{{- end }}{{ "}" }}{{ end }}
   {{ define "sns.default.subject" }}[{{ .Status | toUpper }}{{ if eq .Status
 "firing" }}:{{ .Alerts.Firing | len }}{{ end }}]{{ end }}
```

> **Note**
>
> Template ini membuat JSON dari data alfanumerik. Jika data Anda memiliki karakter khusus, encode mereka sebelum menggunakan template ini.

Untuk memastikan bahwa template ini digunakan dalam notifikasi keluar, rujuk di `alertmanager_config` blok Anda sebagai berikut:

```
alertmanager_config: |
  global:
  templates:
    - 'default_template'
```

> **Note**
>
> Template ini untuk seluruh badan pesan sebagai JSON. Template ini menimpa seluruh isi pesan. Anda tidak dapat mengganti isi pesan jika Anda ingin menggunakan templat khusus ini. Setiap penggantian yang dilakukan secara manual akan diutamakan daripada template.

Untuk informasi lebih lanjut tentang:

- File konfigurasi manajer peringatan, lihatMembuat file konfigurasi manajer peringatan.
- Mengunggah file konfigurasi Anda, lihatMengunggah file konfigurasi pengelola peringatan Anda ke Amazon Managed Service untuk Prometheus.

(Opsional) Mengirim dari Amazon SNS ke tujuan lain

Saat ini, Amazon Managed Service untuk Prometheus dapat mengirim pesan peringatan langsung ke Amazon SNS saja. Anda dapat mengonfigurasi Amazon SNS untuk mengirim pesan tersebut ke tujuan lain seperti email, webhook, Slack, dan. OpsGenie

Email

Untuk mengonfigurasi topik Amazon SNS untuk menampilkan pesan ke email, buat langganan. Di konsol Amazon SNS, pilih tab Langganan untuk membuka halaman daftar Langganan. Pilih Buat

Langganan dan pilih Email. Amazon SNS mengirimkan email konfirmasi ke alamat email yang terdaftar. Setelah Anda menerima konfirmasi, Anda dapat menerima notifikasi Amazon SNS sebagai email dari topik yang Anda langgani. Untuk informasi selengkapnya, lihat Berlangganan topik Amazon SNS.

Webhook

Untuk mengonfigurasi topik Amazon SNS untuk menampilkan pesan ke titik akhir webhook, buat langganan. Di konsol Amazon SNS, pilih tab Langganan untuk membuka halaman daftar Langganan. Pilih Buat Langganan dan pilih HTTP/HTTPS. Setelah Anda membuat langganan, Anda harus mengikuti langkah-langkah konfirmasi untuk mengaktifkannya. Saat aktif, titik akhir HTTP Anda akan menerima notifikasi Amazon SNS. Untuk informasi selengkapnya, lihat Berlangganan topik Amazon SNS. Untuk informasi selengkapnya tentang menggunakan webhook Slack untuk mempublikasikan pesan ke berbagai tujuan, lihat Bagaimana cara menggunakan webhook untuk mempublikasikan pesan Amazon SNS ke Amazon Chime, Slack, atau Microsoft Teams?

Kendur

Untuk mengonfigurasi topik Amazon SNS untuk menampilkan pesan ke Slack, Anda memiliki dua opsi. Anda dapat mengintegrasikan dengan email-to-channel integrasi Slack, yang memungkinkan Slack menerima pesan email dan meneruskannya ke saluran Slack, atau Anda dapat menggunakan fungsi Lambda untuk menulis ulang notifikasi Amazon SNS ke Slack. Untuk informasi selengkapnya tentang meneruskan email ke saluran slack, lihat Mengonfirmasi Langganan Topik AWS SNS untuk Slack Webhook. Untuk informasi selengkapnya tentang membuat fungsi Lambda untuk mengonversi pesan Amazon SNS ke Slack, lihat Cara mengintegrasikan Layanan Terkelola Amazon untuk Prometheus dengan Slack.

OpsGenie

Untuk selengkapnya tentang cara mengonfigurasi topik Amazon SNS untuk menampilkan pesan OpsGenie, lihat Mengintegrasikan Oppgenie dengan Amazon SNS yang masuk.

Aturan validasi dan pemotongan pesan penerima SNS

Pesan SNS akan divalidasi, dipotong, atau dimodifikasi, jika perlu, oleh penerima SNS berdasarkan aturan berikut:

- Pesan berisi karakter non-utf.
  - Pesan akan diganti dengan “Kesalahan - bukan string yang dikodekan UTF-8 yang valid.”

  - Satu atribut pesan akan ditambahkan dengan kunci “terpotong” dan nilai “benar”
  - Satu atribut pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai “Pesan: Kesalahan - bukan string yang dikodekan UTF-8 yang valid.”
- Pesan kosong.
  - Pesan akan diganti dengan “Kesalahan - Pesan seharusnya tidak kosong.”
  - Satu atribut pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai “Pesan: Kesalahan - Pesan tidak boleh kosong.”
- Pesan telah terpotong.
  - Pesan akan memiliki konten terpotong.
  - Satu atribut pesan akan ditambahkan dengan kunci “terpotong” dan nilai “benar”
  - Satu atribut pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai “Pesan: Kesalahan - Pesan telah terpotong dari *X* KB, karena melebihi batas ukuran 256 KB.”
- Subjek bukan ASCII.
  - Subjek akan diganti dengan “Kesalahan - berisi karakter ASCII yang tidak dapat dicetak.”
  - Satu atribut pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai “Subjek: Kesalahan - berisi karakter ASCII yang tidak dapat dicetak.”
- Subjek telah terpotong.
  - Subjek akan memiliki konten terpotong.
  - Satu atribut pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai “Subjek: Kesalahan - Subjek telah terpotong dari karakter *X*, karena melebihi batas ukuran karakter 100.”
- Atribut pesan memiliki kunci/nilai yang tidak valid.
  - Atribut pesan tidak valid akan dihapus.
  - Satu atribusi pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai "MessageAttribute: Kesalahan - *X* atribut pesan telah dihapus karena tidak valid MessageAttributeKey atau.” MessageAttributeValue
- Atribut pesan telah terpotong.
  - Atribut pesan tambahan akan dihapus.
  - Satu atribut pesan akan ditambahkan dengan kunci “dimodifikasi” dan nilai "MessageAttribute: Kesalahan - *X* atribut pesan telah dihapus, karena melebihi batas ukuran 256KB.

Mengunggah file konfigurasi pengelola peringatan Anda ke Amazon Managed Service untuk Prometheus

Sekarang, Anda harus mengunggah file konfigurasi manajer peringatan Anda ke Amazon Managed Service untuk Prometheus. Anda dapat menggunakan konsol atau AWS CLI untuk mengunggahnya.

Untuk menggunakan Amazon Managed Service untuk konsol Prometheus untuk mengunggah konfigurasi manajer peringatan

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di https://console.aws.amazon.com/prometheus/.
2. Di sudut kiri atas halaman, pilih ikon menu, lalu pilih Semua ruang kerja.
3. Pilih ID ruang kerja ruang kerja, lalu pilih tab Manajer peringatan.
4. Jika ruang kerja belum memiliki definisi manajer peringatan, pilih Tambahkan definisi. Jika ruang kerja memiliki definisi manajer peringatan yang ingin Anda ganti, pilih Ganti definisi.
5. Pilih Pilih file, pilih file definisi manajer peringatan, dan pilih Lanjutkan.

Untuk menggunakan konfigurasi manajer peringatan AWS CLI untuk mengunggah ke ruang kerja untuk pertama kalinya

1. Base64 menyandikan konten file pengelola peringatan Anda. Di Linux, Anda dapat menggunakan perintah berikut:

   ```
   base64 input-file output-file
   ```

   Di macOS, Anda dapat menggunakan perintah berikut:

   ```
   openssl base64 input-file output-file
   ```

2. Untuk mengunggah file, masukkan salah satu perintah berikut.

   Pada AWS CLI versi 2, masukkan:

   ```
   aws amp create-alert-manager-definition --data file://path_to_base_64_output_file
    --workspace-id my-workspace-id --region region
   ```

   Pada AWS CLI versi 1, masukkan:

```
aws amp create-alert-manager-definition --data fileb://path_to_base_64_output_file
 --workspace-id my-workspace-id --region region
```

3. Dibutuhkan beberapa detik agar konfigurasi manajer peringatan Anda menjadi aktif. Untuk memeriksa status, masukkan perintah berikut:

```
aws amp describe-alert-manager-definition --workspace-id workspace_id --
region region
```

Jika `status` ya`ACTIVE`, definisi manajer peringatan baru Anda telah berlaku.

Untuk menggunakan AWS CLI untuk mengganti konfigurasi manajer peringatan ruang kerja dengan yang baru

1. Base64 menyandikan konten file pengelola peringatan Anda. Di Linux, Anda dapat menggunakan perintah berikut:

```
base64 input-file output-file
```

Di macOS, Anda dapat menggunakan perintah berikut:

```
openssl base64 input-file output-file
```

2. Untuk mengunggah file, masukkan salah satu perintah berikut.

Pada AWS CLI versi 2, masukkan:

```
aws amp put-alert-manager-definition --data file://path_to_base_64_output_file --
workspace-id my-workspace-id --region region
```

Pada AWS CLI versi 1, masukkan:

```
aws amp put-alert-manager-definition --data fileb://path_to_base_64_output_file --
workspace-id my-workspace-id --region region
```

3. Dibutuhkan beberapa detik agar konfigurasi manajer peringatan baru Anda menjadi aktif. Untuk memeriksa status, masukkan perintah berikut:

```
aws amp describe-alert-manager-definition --workspace-id workspace_id --
region region
```

Jika `status` ya`ACTIVE`, definisi manajer peringatan baru Anda telah berlaku. Hingga saat itu, konfigurasi manajer peringatan Anda sebelumnya masih aktif.

Mengintegrasikan peringatan dengan Grafana Terkelola Amazon atau Grafana open source

Aturan peringatan yang telah Anda buat di Alertmanager dalam Layanan Terkelola Amazon untuk Prometheus dapat diteruskan dan dilihat di Grafana dan Grafana yang Dikelola Amazon, menyatukan aturan peringatan dan peringatan Anda dalam satu lingkungan. Dalam Grafana Terkelola Amazon, Anda dapat melihat aturan peringatan dan peringatan yang dihasilkan.

Prasyarat

Sebelum mulai mengintegrasikan Amazon Managed Service untuk Prometheus ke Amazon Managed Grafana, Anda harus telah menyelesaikan prasyarat berikut:

- Anda harus memiliki kredensional yang ada Akun AWS dan IAM untuk membuat Layanan Terkelola Amazon untuk peran Prometheus dan IAM secara terprogram.

  Untuk informasi selengkapnya tentang membuat Akun AWS kredensil IAM dan IAM, lihat. Mengatur
- Anda harus memiliki Layanan Terkelola Amazon untuk ruang kerja Prometheus, dan memasukkan data ke dalamnya. Untuk menyiapkan ruang kerja baru, lihatBuat ruang kerja. Anda juga harus terbiasa dengan konsep Prometheus seperti Alertmanager dan Ruler. Untuk informasi lebih lanjut tentang topik ini, lihat dokumentasi Prometheus.
- Anda memiliki konfigurasi Alertmanager dan file aturan yang sudah dikonfigurasi di Amazon Managed Service untuk Prometheus. Untuk informasi selengkapnya tentang Alertmanager di Amazon Managed Service for Prometheus, lihat. Manajer Peringatan Untuk informasi selengkapnya tentang aturan, lihatMerekam aturan dan aturan peringatan.
- Anda harus menyiapkan Grafana Terkelola Amazon, atau Grafana versi open source yang berjalan.

- Jika Anda menggunakan Grafana Terkelola Amazon, Anda harus menggunakan peringatan Grafana. Untuk informasi selengkapnya, lihat Memigrasi lansiran dasbor lama ke peringatan Grafana.
- Jika Anda menggunakan Grafana versi open source, Anda harus menjalankan versi 9.1 atau lebih tinggi.

> **Note**
>
> Anda dapat menggunakan Grafana versi sebelumnya, tetapi Anda harus mengaktifkan fitur peringatan terpadu (peringatan Grafana), dan Anda mungkin harus menyiapkan proxy sigv4 untuk melakukan panggilan dari Grafana ke Layanan Terkelola Amazon untuk Prometheus. Untuk informasi selengkapnya, lihat Siapkan open source Grafana atau Grafana Enterprise untuk digunakan dengan Amazon Managed Service for Prometheus.

- Grafana yang Dikelola Amazon harus memiliki izin berikut untuk sumber daya Prometheus Anda. Anda harus menambahkannya ke kebijakan yang dikelola layanan atau yang dikelola pelanggan yang dijelaskan dalam. https://docs.aws.amazon.com/grafana/latest/userguide/AMG-manage-permissions.html
  - `aps:ListRules`
  - `aps:ListAlertManagerSilences`
  - `aps:ListAlertManagerAlerts`
  - `aps:GetAlertManagerStatus`
  - `aps:ListAlertManagerAlertGroups`
  - `aps:PutAlertManagerSilences`
  - `aps:DeleteAlertManagerSilence`

Menyiapkan Grafana yang Dikelola Amazon

Jika Anda telah menyiapkan aturan dan peringatan di Instans Layanan Terkelola Amazon untuk Prometheus, konfigurasi untuk menggunakan Grafana Terkelola Amazon sebagai dasbor untuk peringatan tersebut dilakukan sepenuhnya dalam Grafana yang Dikelola Amazon.

Untuk mengonfigurasi Grafana Terkelola Amazon sebagai dasbor peringatan

1. Buka konsol Grafana untuk ruang kerja Anda.

2. Di bawah Konfigurasi, pilih Sumber data.
3. Buat atau buka sumber data Prometheus Anda. Jika sebelumnya Anda belum menyiapkan sumber data Prometheus, lihat untuk informasi lebih lanjut. Tambahkan sumber data Prometheus di Grafana
4. Di sumber data Prometheus, pilih Kelola peringatan melalui UI Alertmanager.
5. Kembali ke antarmuka Sumber data.
6. Buat sumber data Alertmanager baru.
7. Di halaman konfigurasi sumber data Alertmanager, tambahkan pengaturan berikut:
   - Setel Implementasi ke`Prometheus`.
   - Untuk pengaturan URL, gunakan URL untuk ruang kerja Prometheus Anda, hapus semuanya setelah ID ruang kerja, dan tambahkan ke akhir. `/alertmanager` Misalnya, *`https://aps-workspaces.us-east1.amazonaws.com/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager`*.
   - Di bawah Auth, nyalakan Sigv4auth. Ini memberitahu Grafana untuk menggunakan AWSotentikasi untuk permintaan.
   - Di bawah Detail Sigv4Auth, untuk Wilayah Default, berikan wilayah instance Prometheus Anda, misalnya. `us-east-1`
   - Setel opsi Default ke`true`.
8. Pilih Simpan dan uji.
9. Layanan Terkelola Amazon Anda untuk peringatan Prometheus sekarang harus dikonfigurasi agar berfungsi dengan instans Grafana Anda. Pastikan Anda dapat melihat aturan Peringatan, grup Peringatan (termasuk lansiran aktif), dan Pembungkaman dari Layanan Terkelola Amazon untuk instance Prometheus di halaman Peringatan Grafana.

Pemecahan Masalah Manajer Peringatan

Dengan menggunakanCloudWatch Log, Anda dapat memecahkan masalah terkait Pengelola Peringatan dan Penggaris. Bagian ini berisi topik pemecahan masalah terkait Alert Manager.

Topik

- Peringatan konten kosong
- Peringatan non ASCII
- Peringatan tidak valid key/value

- Peringatan batas pesan
- Tidak ada kesalahan kebijakan berbasis sumber daya

Peringatan konten kosong

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
   "message": {
       "log": "Message has been modified because the content was empty."
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

Ini berarti bahwa template manajer Alert menyelesaikan peringatan keluar ke pesan kosong.

Tindakan yang harus diambil

Validasi template manajer Alert Anda dan pastikan bahwa Anda memiliki template yang valid untuk semua jalur penerima.

Peringatan non ASCII

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
   "message": {
       "log": "Subject has been modified because it contains control or non-ASCII
 characters."
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

Ini berarti bahwa subjek memiliki karakter non-ASCII.

Tindakan yang harus diambil

Hapus referensi di bidang subjek template Anda ke label yang mungkin berisi karakter non-ASCII.

Peringatan tidak valid **`key/value`**

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
   "message": {
       "log": "MessageAttributes has been removed because of invalid key/value,
 numberOfRemovedAttributes=1"
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

Ini berarti bahwa beberapa atribut pesan telah dihapus karena kunci/nilai tidak valid.

Tindakan yang harus diambil

Evaluasi ulang template yang Anda gunakan untuk mengisi atribut pesan, dan pastikan itu menyelesaikan atribut pesan SNS yang valid. Untuk informasi selengkapnya tentang memvalidasi pesan ke topik Amazon SNS, lihat Memvalidasi topik SNS

Peringatan batas pesan

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
   "message": {
       "log": "Message has been truncated because it exceeds size limit,
 originSize=266K, truncatedSize=12K"
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

Ini berarti bahwa beberapa ukuran pesan terlalu besar.

Tindakan yang harus diambil

Lihatlah template pesan penerima Peringatan dan kerjakan ulang agar sesuai dengan batas ukuran.

Tidak ada kesalahan kebijakan berbasis sumber daya

Ketika log berisi kesalahan berikut

```
{
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
   "message": {
       "log": "Notify for alerts failed, AMP is not authorized to perform: SNS:Publish
 on resource: arn:aws:sns:us-west-2:12345:testSnsReceiver because no resource-based
 policy allows the SNS:Publish action"
       "level": "ERROR"
   },
   "component": "alertmanager"
}
```

Ini berarti bahwa Amazon Managed Service untuk Prometheus tidak memiliki izin untuk mengirimkan peringatan ke topik SNS yang ditentukan.

Tindakan yang harus diambil

Validasi bahwa kebijakan akses pada topik Amazon SNS Anda memberi Layanan Terkelola Amazon untuk Prometheus kemampuan untuk mengirim pesan SNS ke topik tersebut. Anda dapat memvalidasi kebijakan topik terhadap simulator kebijakan IAM dengan simulator kebijakan IAM. Pastikan Anda memiliki izin dan kebijakan yang diperlukan dalam peran IAM Anda. Untuk mengetahui lebih lanjut tentang izin dan kebijakan IAM, lihat Izin dan kebijakan IAM.

Pencatatan dan pemantauan

Anda dapat mengelola Layanan Terkelola Amazon untuk penggunaan sumber daya Prometheus dengan fitur pencatatan dan pemantauan CloudWatch Amazon.

- Gunakan CloudWatch metrik untuk memantau Amazon Managed Service untuk Prometheus.
- Gunakan CloudWatch Log untuk menanyakan dan melihat Layanan Terkelola Amazon untuk manajer peringatan Prometheus dan peristiwa penggaris.

CloudWatch metrik

Layanan Terkelola Amazon untuk Prometheus menjual metrik penggunaan ke. CloudWatch Metrik ini memberikan visibilitas tentang pemanfaatan ruang kerja Anda. Metrik vended dapat ditemukan di `AWS/Usage` dan `AWS/Prometheus` namespace di. CloudWatch Metrik ini tersedia tanpa CloudWatch biaya. Untuk informasi selengkapnya tentang metrik penggunaan, lihat metrik penggunaan CloudWatch .

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
ResourceCount	IngestionRate	`AWS/Usage`	Tingkat konsumsi sampel Satuan: hitung per detik Statistik yang Valid: rata-rata, maks, min, jumlah
ResourceCount	ActiveSeries	`AWS/Usage`	Jumlah seri aktif per ruang kerja Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
ResourceCount	ActiveAlerts	`AWS/Usage`	Jumlah peringatan aktif per ruang kerja

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
			Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
ResourceCount	SizeOfAlerts	AWS/Usage	Ukuran total semua peringatan di ruang kerja, dalam byte Unit: byte Statistik yang Valid: rata-rata, maks, min, jumlah
ResourceCount	SuppressedAlerts	AWS/Usage	Jumlah peringatan dalam keadaan ditekan per ruang kerja. Peringatan dapat ditekan oleh keheningan atau penghambatan. Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
ResourceCount	UnprocessedAlerts	`AWS/Usage`	Jumlah peringatan dalam keadaan belum diproses per ruang kerja. Peringatan dalam keadaan belum diproses setelah diterima oleh AlertManager, tetapi sedang menunggu evaluasi grup agregasi berikutnya. Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
ResourceCount	AllAlerts	`AWS/Usage`	Jumlah peringatan di negara bagian mana pun per ruang kerja. Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
AlertManagerAlerts Received	-	`AWS/Prometheus`	Total lansiran berhasil dikirimkan Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
AlertManagerNotificationsFailed	-	`AWS/Prometheus`	Jumlah pengiriman peringatan yang gagal Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
AlertManagerNotificationsThrottled	-	`AWS/Prometheus`	Jumlah peringatan yang dibatasi Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
DiscardedSamples*	-	`AWS/Prometheus`	Jumlah sampel yang dibuang dengan alasan Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
RuleEvaluations	-	`AWS/Prometheus`	Jumlah total evaluasi aturan Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
RuleEvaluationFailures	-	`AWS/Prometheus`	Jumlah kegagalan evaluasi aturan dalam interval Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah
RuleGroupIterationsMissed	-	`AWS/Prometheus`	Jumlah iterasi Grup Aturan yang terlewatkan dalam interval. Unit: hitung Statistik yang Valid: rata-rata, maks, min, jumlah

* Beberapa alasan yang menyebabkan sampel dibuang adalah sebagai berikut.

Alasan	Arti
greater_than_max_sample_age	Membuang baris log yang lebih tua dari waktu saat ini
new-value-for-timestamp	Sampel duplikat dikirim dengan stempel waktu yang berbeda dari yang direkam sebelumnya
per_metric_series_limit	Pengguna telah mencapai seri aktif per batas metrik
per_user_series_limit	Pengguna telah mencapai jumlah total batas seri aktif
rate_limited	Tingkat konsumsi terbatas
sample-out-of-order	Sampel dikirim keluar dari pesanan dan tidak dapat diproses

Alasan	Arti
label_value_too_long	Nilai label lebih panjang dari batas karakter yang diizinkan
max_label_names_per_series	Pengguna telah menekan nama label per metrik
hilang_metric_name	Nama metrik tidak disediakan
metric_name_invalid	Nama metrik yang diberikan tidak valid
label_invalid	Label tidak valid yang diberikan
duplikate_label_names	Nama label duplikat yang disediakan

> **Note**
>
> Metrik yang tidak ada atau hilang sama dengan nilai metrik itu menjadi 0.

> **Note**
>
> `RuleGroupIterationsMissed`, `RuleEvaluations`, dan `RuleEvaluationFailures` memiliki `RuleGroup` dimensi struktur berikut: *`RuleGroupNamespace`*;*`RuleGroup`*

Menyetel CloudWatch alarm pada metrik penjual Prometheus

Anda dapat memantau penggunaan sumber daya Prometheus menggunakan alarm. CloudWatch

Untuk mengatur alarm pada jumlah ActiveSeriesdi Prometheus

1. Pilih tab Graphed metrics dan gulir ke bawah ke label. ActiveSeries

   Dalam tampilan metrik Grafik, hanya metrik yang saat ini sedang dicerna yang akan muncul.
2. Pilih ikon notifikasi di kolom Tindakan.

3. Di Tentukan metrik dan kondisi, masukkan kondisi ambang batas di bidang Nilai kondisi dan pilih Berikutnya.
4. Di Mengkonfigurasi tindakan, pilih topik SNS yang ada atau buat topik SNS baru untuk mengirim notifikasi.
5. Di Tambahkan nama dan deskripsi, tambahkan nama alarm dan deskripsi opsional.
6. Pilih Buat alarm.

CloudWatch Log

Layanan Terkelola Amazon untuk Prometheus mencatat kesalahan Pengelola Peringatan dan Penggaris dan peristiwa peringatan di grup log di Log Amazon. CloudWatch Untuk informasi selengkapnya tentang Pengelola Peringatan dan Penguasa, lihat topik Pengelola Peringatan di panduan ini. Anda dapat mempublikasikan data log ruang kerja untuk mencatat aliran di CloudWatch Log. Anda dapat mengonfigurasi log yang ingin Anda pantau di Amazon Managed Service untuk konsol Prometheus atau dengan menggunakan file. AWS CLI Anda dapat melihat atau menanyakan log ini di CloudWatch konsol. Untuk informasi selengkapnya tentang melihat aliran CloudWatch log log di konsol, lihat Bekerja dengan grup log dan aliran log CloudWatch dalam panduan CloudWatch pengguna.

Tingkat CloudWatch gratis memungkinkan hingga 5Gb log untuk dipublikasikan di CloudWatch Log. Log yang melebihi tunjangan tingkat gratis akan dibebankan berdasarkan paket CloudWatch harga.

Topik

- Mengkonfigurasi Log CloudWatch

Mengkonfigurasi Log CloudWatch

Layanan Terkelola Amazon untuk Prometheus mencatat kesalahan Pengelola Peringatan dan Penggaris dan peristiwa peringatan di grup log di Log Amazon. CloudWatch

Anda dapat menyetel konfigurasi logging CloudWatch Log di Amazon Managed Service untuk konsol Prometheus atau dengan memanggil permintaan APIAWS CLI. `create-logging-configuration`

Prasyarat

Sebelum meneleponcreate-logging-configuration, lampirkan kebijakan berikut atau izin yang setara ke ID atau peran Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups",
                "aps:CreateLoggingConfiguration",
                "aps:UpdateLoggingConfiguration",
                "aps:DescribeLoggingConfiguration",
                "aps:DeleteLoggingConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

Untuk mengkonfigurasi CloudWatch Log

Anda dapat mengonfigurasi logging di Amazon Managed Service untuk Prometheus menggunakan konsol atau. AWS AWS CLI

Console

Untuk mengonfigurasi logging di Amazon Managed Service untuk konsol Prometheus

1. Arahkan ke tab Log di panel detail ruang kerja Anda.
2. Pilih Kelola log di sisi kanan atas panel Log.
3. Pilih semua dalam daftar dropdown tingkat Log.
4. Pilih grup log yang ingin Anda publikasikan log Anda di daftar dropdown Grup Log.

   Anda juga dapat membuat grup log baru di CloudWatch konsol.

5. Pilih Simpan perubahan.

AWS CLI

Anda dapat mengatur konfigurasi logging menggunakan fileAWS CLI.

Untuk mengkonfigurasi logging menggunakan AWS CLI

- MenggunakanAWS CLI, jalankan perintah berikut.

```
aws amp create-logging-configuration --workspace-id my_workspace_ID
                                  --log-group-arn my-log-group-arn
```

Batasan

- Tidak semua peristiwa dicatat

  Layanan Terkelola Amazon untuk Prometheus hanya mencatat peristiwa yang ada di tingkat atau. `warning` `error`
- Batas ukuran kebijakan

  CloudWatch Kebijakan sumber daya log dibatasi hingga 5120 karakter. Ketika CloudWatch Log mendeteksi bahwa kebijakan mendekati batas ukuran ini, secara otomatis mengaktifkan grup log yang memulai`/aws/vendedlogs/`.

  Bila Anda membuat aturan peringatan dengan logging diaktifkan, Amazon Managed Service untuk Prometheus harus memperbarui kebijakan sumber daya Log CloudWatch Anda dengan grup log yang Anda tentukan. Agar tidak mencapai batas ukuran kebijakan sumber daya CloudWatch Log, awali nama grup CloudWatch log Log Anda dengan`/aws/vendedlogs/`. Saat Anda membuat grup log di Amazon Managed Service untuk konsol Prometheus, nama grup log akan diawali dengan. `/aws/vendedlogs/` Untuk informasi selengkapnya, lihat Mengaktifkan Logging dari AWS Layanan Tertentu di Panduan Pengguna CloudWatch Log.

Integrasi dengan layanan lain AWS

Amazon Managed Service untuk Prometheus terintegrasi dengan layanan lain. AWS Bagian ini menjelaskan integrasi dengan pemantauan biaya Amazon Elastic Kubernetes Service (Amazon EKS) (dengan Kubecost), dan menggunakan modul Terraform untuk membuat solusi observabilitas lengkap untuk proyek EKS Anda dengan Observability Accelerator. AWS

Topik

Mengintegrasikan dengan pemantauan biaya Amazon EKS

Amazon Managed Service for Prometheus terintegrasi dengan pemantauan biaya Amazon Elastic Kubernetes Service (Amazon EKS) (dengan Kubecost) untuk melakukan perhitungan alokasi biaya dan memberikan wawasan untuk mengoptimalkan cluster Kubernetes Anda. Dengan menggunakan Amazon Managed Service for Prometheus dengan Kubecost, Anda dapat menskalakan pemantauan biaya secara andal untuk mendukung klaster yang lebih besar.

Mengintegrasikan dengan Kubecost memberi Anda visibilitas granular ke dalam biaya klaster Amazon EKS Anda. Anda dapat mengumpulkan biaya berdasarkan sebagian besar konteks Kubernetes, dari level container hingga level cluster, dan bahkan level multi-cluster. Anda dapat membuat laporan di seluruh kontainer atau cluster untuk melacak biaya untuk tujuan pertunjukan kembali atau tolak bayar.

Berikut ini memberikan instruksi untuk mengintegrasikan dengan Kubecost dalam skenario tunggal atau multi-cluster:

- Integrasi kluster tunggal — Untuk mempelajari cara mengintegrasikan pemantauan biaya Amazon EKS dengan satu cluster, lihat posting AWS blog Mengintegrasikan Kubecost dengan Amazon Managed Service untuk Prometheus.
- Integrasi multi-cluster — Untuk mempelajari cara mengintegrasikan pemantauan biaya Amazon EKS dengan beberapa cluster, lihat posting AWS blog Pemantauan biaya multi-cluster untuk Amazon EKS menggunakan Kubecost dan Amazon Managed Service untuk Prometheus.

> **Note**
>
> Untuk informasi selengkapnya tentang penggunaan Kubecost, lihat Pemantauan biaya di Panduan Pengguna Amazon EKS.

Menggunakan AWS Observability Accelerator

AWSmenyediakan alat observabilitas, termasuk pemantauan, pencatatan, peringatan, dan dasbor, untuk proyek Amazon Elastic Kubernetes Service (Amazon EKS) Anda. Ini termasuk Layanan Terkelola Amazon untuk Prometheus, Grafana Terkelola Amazon, Distro untuk, dan alat AWSlainnya. OpenTelemetry Untuk membantu Anda menggunakan alat ini bersama-sama, AWS sediakan modul Terraform yang mengonfigurasi observabilitas dengan layanan ini, yang disebut Observability Accelerator. AWS

AWSObservability Accelerator memberikan contoh untuk memantau infrastruktur, penerapan NGINX, dan skenario lainnya. Bagian ini memberikan contoh infrastruktur pemantauan dalam klaster Amazon EKS Anda.

Template Terraform dan instruksi terperinci dapat ditemukan di halaman AWSObservability Accelerator for Terraform. GitHub Anda juga dapat membaca posting blog yang mengumumkan AWS Observability Accelerator.

Prasyarat

Untuk menggunakan AWS Observability Accelerator, Anda harus memiliki kluster Amazon EKS yang sudah ada, dan prasyarat berikut:

- AWS CLI— digunakan untuk memanggil AWS fungsionalitas dari baris perintah.
- kubectl — digunakan untuk mengontrol kluster EKS Anda dari baris perintah.
- Terraform — digunakan untuk mengotomatiskan pembuatan sumber daya untuk solusi ini. Anda harus menyiapkan AWS penyedia dengan peran IAM yang memiliki akses untuk membuat dan mengelola Layanan Terkelola Amazon untuk Prometheus, Grafana Terkelola Amazon, dan IAM dalam akun Anda. AWS Untuk informasi selengkapnya tentang cara mengonfigurasi AWS penyedia untuk Terraform, lihat AWSpenyedia di dokumentasi Terraform.

Menggunakan contoh pemantauan infrastruktur

AWSObservability Accelerator menyediakan contoh templat yang menggunakan modul Terraform yang disertakan untuk menyiapkan dan mengonfigurasi observabilitas untuk klaster Amazon EKS Anda. Contoh ini menunjukkan penggunaan AWS Observability Accelerator untuk mengatur pemantauan infrastruktur. Untuk detail selengkapnya tentang penggunaan template ini dan kemampuan tambahan yang disertakan, lihat Existing Cluster with the AWS Observability Accelerator base and Infrastructure monitoring page on. GitHub

Untuk menggunakan modul pemantauan infrastruktur Terraform

1. Dari folder tempat Anda ingin membuat proyek, kloning repo menggunakan perintah berikut.

```
git clone https://github.com/aws-observability/terraform-aws-observability-
accelerator.git
```

2. Inisialisasi Terraform dengan perintah berikut.

```
cd examples/existing-cluster-with-base-and-infra

terraform init
```

3. Buat `terraform.tfvars` file baru, seperti pada contoh berikut. Gunakan AWS Region dan ID cluster untuk klaster Amazon EKS Anda.

```
# (mandatory) AWS Region where your resources will be located
aws_region = "eu-west-1"

# (mandatory) EKS Cluster name
eks_cluster_id = "my-eks-cluster"
```

4. Buat ruang kerja Grafana Terkelola Amazon, jika Anda belum memilikinya yang ingin Anda gunakan. Untuk informasi tentang cara membuat ruang kerja baru, lihat Membuat ruang kerja pertama Anda di Panduan Pengguna Grafana Terkelola Amazon.

5. Buat dua variabel untuk Terraform untuk menggunakan ruang kerja Grafana Anda dengan menjalankan perintah berikut di baris perintah. Anda harus mengganti *grafana-workspace-id*dengan ID dari ruang kerja Grafana Anda.

```
export TF_VAR_managed_grafana_workspace_id=grafana-workspace-id
```

```
export TF_VAR_grafana_api_key=`aws grafana create-workspace-api-key --key-name
 "observability-accelerator-$(date +%s)" --key-role ADMIN --seconds-to-live 1200 --
workspace-id $TF_VAR_managed_grafana_workspace_id --query key --output text`
```

6. [Opsional] Untuk menggunakan Layanan Terkelola Amazon yang ada untuk ruang kerja Prometheus, tambahkan ID ke `terraform.tfvars` file, seperti pada contoh berikut, ganti dengan ID ruang kerja Prometheus Anda. *`prometheus-workspace-id`* Jika Anda tidak menentukan ruang kerja yang ada, maka ruang kerja Prometheus baru akan dibuat untuk Anda.

```
# (optional) Leave it empty for a new workspace to be created
managed_prometheus_workspace_id = "prometheus-workspace-id"
```

7. Terapkan solusi dengan perintah berikut.

```
terraform apply -var-file=terraform.tfvars
```

Ini akan membuat sumber daya di AWS akun Anda, termasuk yang berikut:

- Layanan Terkelola Amazon baru untuk ruang kerja Prometheus (kecuali Anda memilih untuk menggunakan ruang kerja yang ada).
- Konfigurasi, peringatan, dan aturan manajer peringatan di ruang kerja Prometheus Anda.
- Sumber data Grafana yang Dikelola Amazon baru dan dasbor di ruang kerja Anda saat ini. Sumber data akan dipanggil`aws-observability-accelerator`. Dasbor akan terdaftar di bawah Observability Accelerator Dashboards.
- AWSDistro untuk OpenTelemetry operator yang disiapkan di kluster Amazon EKS yang disediakan, untuk mengirim metrik ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Untuk melihat dasbor baru Anda, buka dasbor tertentu di ruang kerja Grafana Terkelola Amazon Anda. Untuk informasi selengkapnya tentang menggunakan Grafana Terkelola Amazon, lihat Bekerja di ruang kerja Grafana Anda, di Panduan Pengguna Grafana Terkelola Amazon.

Integrasi dengan AWS Controller untuk Kubernetes

Amazon Managed Service for Prometheus terintegrasi AWS dengan Controllers for Kubernetes (ACK), dengan dukungan untuk mengelola ruang kerja, Alert Manager, dan sumber daya Ruler di Amazon EKS. Anda dapat menggunakan AWS Controller untuk definisi sumber daya kustom

Kubernetes (CRD) dan objek Kubernetes asli tanpa harus mendefinisikan sumber daya apa pun di luar klaster Anda.

Bagian ini menjelaskan cara menyiapkan AWS Controller untuk Kubernetes dan Amazon Managed Service untuk Prometheus di cluster Amazon EKS yang ada.

Anda juga dapat membaca posting blog yang memperkenalkan AWS Controller untuk Kubernetes dan memperkenalkan ACK controller untuk Amazon Managed Service untuk Prometheus.

Prasyarat

Sebelum mulai mengintegrasikan AWS Controller untuk Kubernetes dan Amazon Managed Service untuk Prometheus dengan cluster Amazon EKS Anda, Anda harus memiliki prasyarat berikut.

- Anda harus memiliki izin Akun AWS dan izin untuk membuat Layanan Terkelola Amazon untuk peran Prometheus dan IAM secara terprogram.
- Anda harus memiliki kluster Amazon EKS yang sudah ada dengan OpenID Connect (OIDC) diaktifkan.

  Jika Anda tidak mengaktifkan OIDC, Anda dapat menggunakan perintah berikut untuk mengaktifkannya. Ingatlah untuk mengganti *YOUR_CLUSTER_NAME dan AWS_REGION* dengan nilai yang benar untuk akun Anda.

  ```
  eksctl utils associate-iam-oidc-provider \
      --cluster ${YOUR_CLUSTER_NAME} --region ${AWS_REGION} \
      --approve
  ```

  Untuk informasi selengkapnya tentang penggunaan OIDC dengan Amazon EKS, lihat otentikasi penyedia identitas OIDC dan Membuat penyedia IAM OIDC di Panduan Pengguna Amazon EKS.
- Anda harus menginstal driver Amazon EBS CSI di cluster Amazon EKS Anda.
- Anda harus AWS CLImenginstal. AWS CLI Ini digunakan untuk memanggil AWS fungsionalitas dari baris perintah.
- Helm, manajer paket untuk Kubernetes, harus diinstal.
- Metrik bidang kontrol dengan Prometheus harus disiapkan di klaster Amazon EKS Anda.
- Anda harus memiliki topik Amazon Simple Notification Service (Amazon SNS) tempat Anda ingin mengirim peringatan dari ruang kerja baru Anda. Pastikan Anda telah memberikan izin Amazon Managed Service for Prometheus untuk mengirim pesan ke topik tersebut.

Jika kluster Amazon EKS Anda dikonfigurasi dengan tepat, Anda seharusnya dapat melihat metrik yang diformat untuk Prometheus dengan menelepon. `kubectl get --raw /metrics` Sekarang Anda siap untuk menginstal AWS Controllers for Kubernetes service controller dan menggunakannya untuk menyebarkan Amazon Managed Service untuk sumber daya Prometheus.

Menerapkan ruang kerja dengan AWS Controller untuk Kubernetes

Untuk menerapkan Amazon Managed Service baru untuk ruang kerja Prometheus, Anda akan menginstal Controllers for Kubernetes controller, dan kemudian menggunakannya untuk AWS membuat ruang kerja.

Untuk menerapkan Amazon Managed Service baru untuk ruang kerja Prometheus dengan Controller untuk Kubernetes AWS

1. Gunakan perintah berikut untuk menggunakan Helm untuk menginstal Amazon Managed Service for Prometheus service controller. Untuk informasi selengkapnya, lihat Menginstal ACK Controller di dokumentasi AWS Controllers for Kubernetes. GitHub Gunakan *`wilayah`* yang tepat untuk sistem Anda, sepertius`-east-1`.

   ```
   export SERVICE=prometheusservice
   export RELEASE_VERSION=`curl -sL https://api.github.com/repos/aws-controllers-k8s/
   $SERVICE-controller/releases/latest | grep '"tag_name":' | cut -d'"' -f4`
   export ACK_SYSTEM_NAMESPACE=ack-system
   export AWS_REGION=region

   aws ecr-public get-login-password --region us-east-1 | helm registry login --
   username AWS --password-stdin public.ecr.aws
   helm install --create-namespace -n $ACK_SYSTEM_NAMESPACE ack-$SERVICE-controller \
     oci://public.ecr.aws/aws-controllers-k8s/$SERVICE-chart --version=
   $RELEASE_VERSION --set=aws.region=$AWS_REGION
   ```

   Setelah beberapa saat, Anda akan melihat respons yang mirip dengan yang menunjukkan keberhasilan berikut.

   ```
   You are now able to create Amazon Managed Service for Prometheus (AMP) resources!
   The controller is running in "cluster" mode.
   The controller is configured to manage AWS resources in region: "us-east-1"
   ```

   Anda dapat secara opsional memverifikasi bahwa AWS Controllers for Kubernetes controller telah berhasil diinstal dengan perintah berikut.

```
helm list --namespace $ACK_SYSTEM_NAMESPACE -o yaml
```

Ini akan mengembalikan informasi tentang pengontrol`ack-prometheusservice-controller`, termasuk file`status: deployed`.

2. Buat file yang disebut `workspace.yaml` dengan teks berikut. Ini akan digunakan sebagai konfigurasi untuk ruang kerja yang Anda buat.

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: Workspace
metadata:
  name: my-amp-workspace
spec:
  alias: my-amp-workspace
  tags:
    ClusterName: EKS-demo
```

3. Jalankan perintah berikut untuk membuat ruang kerja Anda (perintah ini tergantung pada variabel sistem yang Anda atur di langkah 1).

```
kubectl apply -f workspace.yaml -n $ACK_SYSTEM_NAMESPACE
```

Dalam beberapa saat, Anda akan dapat melihat ruang kerja baru, yang dipanggil `my-amp-workspace` di akun Anda.

Menjalankan perintah berikut untuk melihat detail dan status ruang kerja Anda termasuk ID ruang kerja. Sebagai alternatif, Anda dapat melihat ruang kerja baru di Amazon Managed Service untuk konsol Prometheus.

```
kubectl describe workspace my-amp-workspace -n $ACK_SYSTEM_NAMESPACE
```

> **Note**
>
> Anda juga dapat menggunakan ruang kerja yang ada daripada membuat yang baru.

4. Buat dua file yaml baru sebagai konfigurasi untuk Rulegroups dan AlertManager yang akan Anda buat selanjutnya menggunakan konfigurasi berikut.

Simpan konfigurasi ini sebagai`rulegroup.yaml`. Ganti *`WORKSPACE-ID dengan ID`* ruang kerja dari langkah sebelumnya.

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: RuleGroupsNamespace
metadata:
  name: default-rule
spec:
  workspaceID: WORKSPACE-ID
  name: default-rule
  configuration: |
    groups:
    - name: example
      rules:
      - alert: HostHighCpuLoad
        expr: 100 - (avg(rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 60
        for: 5m
        labels:
          severity: warning
          event_type: scale_up
        annotations:
          summary: Host high CPU load (instance {{ $labels.instance }})
          description: "CPU load is > 60%\n  VALUE = {{ $value }}\n  LABELS =
 {{ $labels }}"
      - alert: HostLowCpuLoad
        expr: 100 - (avg(rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) < 30
        for: 5m
        labels:
          severity: warning
          event_type: scale_down
        annotations:
          summary: Host low CPU load (instance {{ $labels.instance }})
          description: "CPU load is < 30%\n  VALUE = {{ $value }}\n  LABELS =
 {{ $labels }}"
```

Simpan konfigurasi berikut sebagai`alertmanager.yaml`. Ganti *`WORKSPACE-ID dengan ID`* ruang kerja dari langkah sebelumnya. *`Ganti TOPIC-ARN dengan ARN untuk topik Amazon SNS untuk mengirim notifikasi, dan REGION dengan yang Anda gunakan.`* Wilayah AWS Ingat bahwa Amazon Managed Service untuk Prometheus harus memiliki izin untuk topik Amazon SNS.

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: AlertManagerDefinition
metadata:
  name: alert-manager
spec:
  workspaceID: WORKSPACE-ID
  configuration: |
    alertmanager_config: |
      route:
         receiver: default_receiver
      receivers:
        - name: default_receiver
          sns_configs:
          - topic_arn: TOPIC-ARN
            sigv4:
              region: REGION
            message: |
              alert_type: {{ .CommonLabels.alertname }}
              event_type: {{ .CommonLabels.event_type }}
```

> **Note**
>
> Untuk mempelajari lebih lanjut tentang format file konfigurasi ini, lihat RuleGroupsNamespaceDatadan AlertManagerDefinitionData.

5. Jalankan perintah berikut untuk membuat grup aturan dan konfigurasi manajer peringatan (perintah ini bergantung pada variabel sistem yang Anda atur di langkah 1).

```
kubectl apply -f rulegroup.yaml -n $ACK_SYSTEM_NAMESPACE
kubectl apply -f alertmanager.yaml -n $ACK_SYSTEM_NAMESPACE
```

Perubahan akan tersedia dalam beberapa saat.

> **Note**
>
> Untuk memperbarui sumber daya, daripada membuatnya, Anda cukup memperbarui file yaml, dan menjalankan `kubectl apply` perintah lagi.
> Untuk menghapus sumber daya, jalankan perintah berikut. Ganti *ResourceType*dengan jenis sumber daya yang ingin Anda hapus`Workspace`,`AlertManagerDefinition`,

atauRuleGroupNamespace. Ganti *ResourceName*dengan nama sumber daya yang akan dihapus.

```
kubectl delete ResourceType ResourceName -n $ACK_SYSTEM_NAMESPACE
```

Itu menyelesaikan penerapan ruang kerja baru. Bagian selanjutnya menjelaskan konfigurasi klaster Anda untuk mengirim metrik ke ruang kerja tersebut.

Mengonfigurasi klaster Amazon EKS Anda untuk menulis ke Layanan Terkelola Amazon untuk ruang kerja Prometheus

Bagian ini menjelaskan cara menggunakan Helm untuk mengonfigurasi Prometheus yang berjalan di klaster Amazon EKS Anda untuk menulis metrik jarak jauh ke Amazon Managed Service untuk ruang kerja Prometheus yang Anda buat di bagian sebelumnya.

Untuk prosedur ini, Anda akan memerlukan nama peran IAM yang telah Anda buat untuk digunakan untuk menelan metrik. Jika Anda belum melakukan ini, lihat Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS untuk informasi dan instruksi lebih lanjut. Jika Anda mengikuti instruksi tersebut, peran IAM akan dipanggil`amp-iamproxy-ingest-role`.

Untuk mengonfigurasi klaster Amazon EKS Anda untuk penulisan jarak jauh

1. Gunakan perintah berikut untuk mendapatkan ruang `prometheusEndpoint` kerja Anda. Ganti *WORKSPACE-ID dengan ID* ruang kerja dari bagian sebelumnya.

   ```
   aws amp describe-workspace --workspace-id WORKSPACE-ID
   ```

   PromeTheUsendPoint akan berada di hasil pengembalian, dan diformat seperti ini:

   ```
   https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-a1b2c3d4-a123-b456-c789-ac1234567890/
   ```

   Simpan URL ini untuk digunakan dalam beberapa langkah berikutnya.

2. Buat file baru dengan teks berikut dan sebut saja`prometheus-config.yaml`. Ganti *akun* dengan ID akun Anda, *WorkspaceURL/* dengan URL yang baru saja Anda temukan, dan *wilayah* dengan yang sesuai Wilayah AWS untuk sistem Anda.

```
serviceAccounts:
        server:
            name: "amp-iamproxy-ingest-service-account"
            annotations:
                eks.amazonaws.com/role-arn: "arn:aws:iam::account:role/amp-
iamproxy-ingest-role"
server:
    remoteWrite:
        - url: workspaceURL/api/v1/remote_write
          sigv4:
            region: region
          queue_config:
            max_samples_per_send: 1000
            max_shards: 200
            capacity: 2500
```

3. Temukan bagan Prometheus dan namespace nama serta versi bagan dengan perintah Helm berikut.

```
helm ls --all-namespaces
```

Berdasarkan langkah-langkah sejauh ini, bagan Prometheus dan namespace keduanya harus diberi nama, dan versi bagan mungkin `prometheus 15.2.0`

4. Jalankan perintah berikut, menggunakan *PrometheusChartNamePrometheusNamespace*,, dan *PrometheusChartVersion*ditemukan di langkah sebelumnya.

```
helm upgrade PrometheusChartName prometheus-community/prometheus -
n PrometheusNamespace -f prometheus-config.yaml --version PrometheusChartVersion
```

Setelah beberapa menit, Anda akan melihat pesan bahwa peningkatan berhasil.

5. Secara opsional, validasi bahwa metrik berhasil dikirim dengan menanyakan Layanan Terkelola Amazon untuk titik akhir Prometheus melalui. `awscurl` Ganti *Region* dengan Wilayah AWS yang Anda gunakan, dan *WorkspaceURL/* dengan URL yang Anda temukan di langkah 1.

```
awscurl --service="aps" --region="Region" "workspaceURL/api/v1/query?
query=node_cpu_seconds_total"
```

Anda sekarang telah membuat Amazon Managed Service untuk ruang kerja Prometheus dan terhubung dengannya dari klaster Amazon EKS Anda, menggunakan file YAMAL sebagai konfigurasi. File-file ini, yang disebut definisi sumber daya khusus (CRD), tinggal di dalam kluster Amazon EKS Anda. Anda dapat menggunakan AWS Controller for Kubernetes controller untuk mengelola semua Amazon Managed Service untuk sumber daya Prometheus langsung dari cluster.

Mengintegrasikan CloudWatch metrik dengan Firehose

Bagian ini menjelaskan cara menginstrumentasikan aliran CloudWatch metrik Amazon dan menggunakan Amazon Data Firehose dan AWS Lambdauntuk memasukkan metrik ke dalam Layanan Terkelola Amazon untuk Prometheus.

Anda akan menyiapkan tumpukan menggunakan AWS Cloud Development Kit (CDK) untuk membuat Firehose Delivery Stream, Lambda, dan bucket Amazon S3 untuk mendemonstrasikan skenario lengkap.

Infrastruktur

Hal pertama yang harus Anda lakukan adalah mengatur infrastruktur untuk resep ini.

CloudWatch aliran metrik memungkinkan penerusan data metrik streaming ke titik akhir HTTP atau bucket Amazon S3.

Menyiapkan infrastruktur akan terdiri dari 4 langkah:

- Mengkonfigurasi prasyarat
- Membuat Layanan Terkelola Amazon untuk ruang kerja Prometheus
- Menginstal dependensi
- Menyebarkan tumpukan

Prasyarat

- AWS CLI Itu diinstal dan dikonfigurasi di lingkungan Anda.
- AWS CDK TypeScript diinstal di lingkungan Anda.
- Node.js dan Go diinstal di lingkungan Anda.
- Repositori github eksportir CloudWatch metrik AWS observabilitas (`CWMetricsStreamExporter`) telah dikloning ke mesin lokal Anda.

Untuk membuat Amazon Managed Service untuk ruang kerja Prometheus

1. Aplikasi demo dalam resep ini akan berjalan di atas Amazon Managed Service untuk Prometheus. Buat Amazon Managed Service untuk Prometheus Workspace melalui perintah berikut:

   ```
   aws amp create-workspace --alias prometheus-demo-recipe
   ```

2. Pastikan ruang kerja Anda telah dibuat dengan perintah berikut:

   ```
   aws amp list-workspaces
   ```

   Untuk informasi selengkapnya tentang Layanan Terkelola Amazon untuk Prometheus, lihat Panduan Pengguna Layanan Terkelola Amazon untuk Prometheus.

Untuk menginstal dependensi

1. Instal dependensi

   Dari root `aws-o11y-recipes` repositori, ubah direktori Anda untuk `CWMetricStreamExporter` menggunakan perintah:

   ```
   cd sandbox/CWMetricStreamExporter
   ```

   Ini sekarang akan dianggap sebagai akar repo, ke depan.

2. Ubah direktori ke `/cdk` melalui perintah berikut:

   ```
   cd cdk
   ```

3. Instal dependensi CDK melalui perintah berikut:

   ```
   npm install
   ```

4. Ubah direktori kembali ke root repo, dan kemudian ubah direktori untuk `/lambda` menggunakan perintah berikut:

   ```
   cd lambda
   ```

5. Setelah berada di `/lambda` folder, instal dependensi Go menggunakan:

```
go get
```

Semua dependensi sekarang diinstal.

Untuk menyebarkan tumpukan

1. Di root repo, buka `config.yaml` dan ubah URL Layanan Terkelola Amazon untuk ruang kerja Prometheus dengan mengganti `{workspace}` dengan id ruang kerja yang baru dibuat, dan wilayah tempat Anda berada Layanan Terkelola Amazon untuk ruang kerja Prometheus.

   Misalnya, ubah yang berikut ini dengan:

   ```
   AMP:
       remote_write_url: "https://aps-workspaces.us-east-2.amazonaws.com/workspaces/
   {workspaceId}/api/v1/remote_write"
       region: us-east-2
   ```

   Ubah nama aliran pengiriman Firehose dan bucket Amazon S3 sesuai keinginan Anda.
2. Untuk membangun kode AWS CDK dan Lambda, di root repo jalankan pujian berikut:

   ```
   npm run build
   ```

   Langkah pembuatan ini memastikan bahwa biner Go Lambda dibangun, dan menyebarkan CDK ke. CloudFormation
3. Untuk menyelesaikan penerapan, tinjau dan terima perubahan IAM yang dibutuhkan tumpukan.
4. (Opsional) Anda bisa sangat jika tumpukan telah dibuat dengan menjalankan perintah berikut.

   ```
   aws cloudformation list-stacks
   ```

   Sebuah tumpukan bernama `CDK Stack` akan ada dalam daftar.

Membuat CloudWatch aliran Amazon

Sekarang setelah Anda memiliki fungsi lambda untuk menangani metrik, Anda dapat membuat aliran metrik dari Amazon. CloudWatch

Untuk membuat aliran CloudWatch metrik

1. Arahkan ke CloudWatch konsol, di https://console.aws.amazon.com/cloudwatch/home#metric-streams:streamsList dan pilih Buat aliran metrik.
2. Pilih metrik yang diperlukan, baik semua metrik, atau hanya dari ruang nama yang dipilih.
3. Di bawah`Configuration`, pilih Pilih Firehose yang sudah ada yang dimiliki oleh akun Anda.
4. Anda akan menggunakan Firehose yang dibuat sebelumnya oleh CDK. Dalam menu drop-down Select your Kinesis data Firehose stream, pilih stream yang dibuat sebelumnya. Itu akan memiliki nama seperti`CdkStack-KinesisFirehoseStream123456AB-sample1234`.
5. Ubah format output ke JSON.
6. Beri nama aliran metrik yang berarti bagi Anda.
7. Pilih Buat stream metrik.
8. (Opsional) Untuk memverifikasi pemanggilan fungsi Lambda, navigasikan ke konsol Lambda dan pilih fungsinya. `KinesisMessageHandler` Pilih tab Monitor dan subtab Log, dan di bawah Pemanggilan Terbaru harus ada entri fungsi Lambda yang dipicu.

   > **Note**
   >
   > Mungkin diperlukan waktu hingga 5 menit sebelum pemanggilan mulai ditampilkan di tab Monitor.

Metrik Anda sekarang sedang dialirkan dari Amazon ke CloudWatch Amazon Managed Service untuk Prometheus.

Pembersihan

Anda mungkin ingin membersihkan sumber daya yang digunakan dalam contoh ini. Prosedur berikut menjelaskan cara melakukannya. Ini akan menghentikan aliran metrik yang Anda buat.

Untuk membersihkan sumber daya

1. Mulailah dengan menghapus CloudFormation tumpukan dengan perintah berikut:

   ```
   cd cdk
   cdk destroy
   ```

2. Hapus Layanan Terkelola Amazon untuk ruang kerja Prometheus:

```
aws amp delete-workspace --workspace-id \
    `aws amp list-workspaces --alias prometheus-sample-app --query
 'workspaces[0].workspaceId' --output text`
```

3. Terakhir, hapus aliran CloudWatch metrik Amazon menggunakan CloudWatch konsol Amazon.

Keamanan di Amazon Managed Service untuk Prometheus

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. Model tanggung jawab bersama menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan–layanan AWS di dalam AWS Cloud. AWS juga memberikan Anda layanan yang dapat digunakan dengan aman. Auditor pihak ketiga melakukan pengujian dan verifikasi secara berkala terhadap efektivitas keamanan kami sebagai bagian dari Program Kepatuhan AWS . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Layanan Terkelola Amazon untuk Prometheus, AWSlihat Layanan dalam Lingkup oleh Layanan Program Kepatuhan dalam Lingkup oleh Kepatuhan.
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon Managed Service for Prometheus. Topik berikut menunjukkan cara mengonfigurasi Layanan Terkelola Amazon untuk Prometheus agar memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan Layanan Terkelola Amazon untuk sumber daya Prometheus.

Topik

- Logging Amazon Managed Service untuk panggilan API Prometheus menggunakan AWS CloudTrail
- Mengatur peran IAM untuk akun layanan
- Menggunakan Amazon Managed Service untuk Prometheus dengan titik akhir VPC antarmuka

Perlindungan data di Amazon Managed Service untuk Prometheus

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Amazon Managed Service untuk Prometheus. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk memelihara kendali atas isi yang dihost pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya lindungi kredensial Akun AWS dan siapkan untuk masing-masing pengguna AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya AWS. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pengelogan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama semua kontrol keamanan bawaan dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon Managed Service untuk Prometheus

atau Layanan AWS lainnya menggunakan konsol, API, atau SDK. AWS CLI AWS Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Topik

Data yang dikumpulkan oleh Amazon Managed Service untuk Prometheus

Amazon Managed Service for Prometheus mengumpulkan dan menyimpan metrik operasional yang Anda konfigurasikan untuk dikirim dari server Prometheus yang berjalan di akun Anda ke Amazon Managed Service for Prometheus. Data ini mencakup yang berikut:

- Nilai metrik
- Label metrik (atau pasangan nilai kunci arbitrer) yang membantu mengidentifikasi dan mengklasifikasikan data
- Stempel waktu untuk sampel data

ID penyewa unik mengisolasi data dari pelanggan yang berbeda. ID ini membatasi data pelanggan yang dapat diakses. Pelanggan tidak dapat mengubah ID penyewa.

Amazon Managed Service for Prometheus mengenkripsi data yang disimpan dengan kunci (). AWS Key Management Service AWS KMS Amazon Managed Service untuk Prometheus mengelola kunci-kunci ini.

> **Note**
>
> Amazon Managed Service untuk Prometheus tidak mendukung pembuatan kunci yang dikelola pelanggan. Amazon Managed Service untuk Prometheus tidak dimaksudkan untuk menyimpan data yang sangat sensitif. Data sisi server dienkripsi atas nama Anda menggunakan kunci terkelola. AWS Untuk informasi selengkapnya tentang kunci ini, lihat kunci AWS terkelola di Panduan AWS Key Management Service Pengembang.

Data dalam perjalanan dienkripsi dengan HTTPS secara otomatis. Layanan Terkelola Amazon untuk Prometheus mengamankan koneksi antara Availability Zone dalam Wilayah menggunakan HTTPS secara internal. AWS

Enkripsi diam

Secara default, Amazon Managed Service untuk Prometheus secara otomatis memberi Anda enkripsi saat istirahat dan melakukan ini menggunakan kunci enkripsi yang dimiliki. AWS

- AWSkunci yang dimiliki — Amazon Managed Service untuk Prometheus menggunakan kunci ini untuk secara otomatis mengenkripsi data yang diunggah ke ruang kerja Anda. Anda tidak dapat melihat, mengelola, atau menggunakan kunci yang AWS dimiliki, atau mengaudit penggunaannya. Namun, Anda tidak perlu mengambil tindakan apa pun atau mengubah program apa pun untuk melindungi kunci yang mengenkripsi data Anda. Untuk informasi selengkapnya, lihat kunci yang AWS dimiliki di Panduan AWS Key Management Service Pengembang.

Enkripsi data saat istirahat membantu mengurangi overhead operasional dan kompleksitas yang digunakan untuk melindungi data pelanggan yang sensitif, seperti informasi yang dapat diidentifikasi secara pribadi. Ini memungkinkan Anda untuk membangun aplikasi aman yang memenuhi kepatuhan enkripsi yang ketat dan persyaratan peraturan.

Anda dapat memilih untuk menggunakan kunci yang dikelola pelanggan saat membuat ruang kerja:

- Kunci terkelola pelanggan — Layanan Terkelola Amazon untuk Prometheus mendukung penggunaan kunci terkelola pelanggan simetris yang Anda buat, miliki, dan kelola untuk mengenkripsi data di ruang kerja Anda. Karena Anda memiliki kontrol penuh atas enkripsi ini, Anda dapat melakukan tugas-tugas seperti:
  - Menetapkan dan memelihara kebijakan utama
  - Menetapkan dan memelihara kebijakan dan hibah IAM
  - Mengaktifkan dan menonaktifkan kebijakan utama
  - Memutar bahan kriptografi kunci
  - Menambahkan tanda
  - Membuat alias kunci
  - Kunci penjadwalan untuk penghapusan

  Untuk informasi selengkapnya, lihat kunci terkelola pelanggan di Panduan AWS Key Management Service Pengembang.

Pilih apakah akan menggunakan kunci yang dikelola pelanggan atau kunci AWS yang dimiliki dengan hati-hati. Ruang kerja yang dibuat dengan kunci yang dikelola pelanggan tidak dapat dikonversi untuk menggunakan kunci yang AWS dimiliki nanti (dan sebaliknya).

> **Note**
>
> Layanan Terkelola Amazon untuk Prometheus secara otomatis mengaktifkan enkripsi saat istirahat AWS menggunakan kunci yang dimiliki untuk melindungi data Anda tanpa biaya. Namun, AWS KMS biaya berlaku untuk menggunakan kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang harga, silakan lihat harga AWS Key Management Service.

Untuk informasi lebih lanjut tentangAWS KMS, lihat Apa ituAWS Key Management Service?

> **Note**
>
> Ruang kerja yang dibuat dengan kunci terkelola pelanggan tidak dapat menggunakan kolektor AWS terkelola untuk konsumsi.

Bagaimana Amazon Managed Service untuk Prometheus menggunakan hibah di AWS KMS

Amazon Managed Service untuk Prometheus memerlukan tiga hibah untuk menggunakan kunci terkelola pelanggan Anda.

Saat Anda membuat Layanan Terkelola Amazon untuk ruang kerja Prometheus yang dienkripsi dengan kunci yang dikelola pelanggan, Layanan Terkelola Amazon untuk Prometheus membuat tiga hibah atas nama Anda dengan mengirimkan permintaan ke. CreateGrantAWS KMS Hibah AWS KMS digunakan untuk memberikan Amazon Managed Service untuk Prometheus akses ke kunci KMS di akun Anda, bahkan ketika tidak dipanggil langsung atas nama Anda (misalnya, saat menyimpan data metrik yang telah dikikis dari kluster Amazon EKS.

Layanan Terkelola Amazon untuk Prometheus memerlukan hibah untuk menggunakan kunci terkelola pelanggan Anda untuk operasi internal berikut:

- Kirim DescribeKeypermintaan AWS KMS untuk memverifikasi bahwa kunci KMS terkelola pelanggan simetris yang diberikan saat membuat ruang kerja valid.

- Kirim GenerateDataKeypermintaan AWS KMS untuk menghasilkan kunci data yang dienkripsi oleh kunci terkelola pelanggan Anda.
- Kirim permintaan Dekripsi ke AWS KMS untuk mendekripsi kunci data terenkripsi sehingga mereka dapat digunakan untuk mengenkripsi data Anda.

Layanan Terkelola Amazon untuk Prometheus membuat tiga hibah ke AWS KMS kunci yang memungkinkan Layanan Dikelola Amazon untuk Prometheus menggunakan kunci atas nama Anda. Anda dapat menghapus akses ke kunci dengan mengubah kebijakan kunci, dengan menonaktifkan kunci, atau dengan mencabut hibah. Anda harus memahami konsekuensi dari tindakan ini sebelum melakukannya. Hal ini dapat menyebabkan hilangnya data di ruang kerja Anda.

Jika Anda menghapus akses ke salah satu hibah dengan cara apa pun, Layanan Terkelola Amazon untuk Prometheus tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, atau menyimpan data baru yang dikirim ke ruang kerja, yang memengaruhi operasi yang bergantung pada data tersebut. Data baru yang dikirim ke ruang kerja tidak akan dapat diakses dan mungkin hilang secara permanen.

⚠ Warning

- Jika Anda menonaktifkan kunci, atau menghapus Layanan Terkelola Amazon untuk akses Prometheus dalam kebijakan kunci, data ruang kerja tidak lagi dapat diakses. Data baru yang dikirim ke ruang kerja tidak akan dapat diakses dan mungkin hilang secara permanen.

  Anda bisa mendapatkan akses ke data ruang kerja dan mulai menerima data baru lagi dengan memulihkan akses Amazon Managed Service untuk Prometheus ke kunci.
- Jika Anda mencabut hibah, hibah tidak dapat dibuat ulang, dan data di ruang kerja akan hilang secara permanen.

Langkah 1: Buat kunci yang dikelola pelanggan

Anda dapat membuat kunci terkelola pelanggan simetris dengan menggunakanAWS Management Console, atau AWS KMS API. Kuncinya tidak harus berada di akun yang sama dengan Layanan Terkelola Amazon untuk ruang kerja Prometheus, selama Anda memberikan akses yang benar melalui kebijakan, seperti yang dijelaskan di bawah ini.

Untuk membuat kunci terkelola pelanggan simetris

Ikuti langkah-langkah untuk Membuat kunci terkelola pelanggan simetris di Panduan AWS Key Management Service Pengembang.

Kebijakan utama

Kebijakan utama mengontrol akses ke kunci yang dikelola pelanggan Anda. Setiap kunci yang dikelola pelanggan harus memiliki persis satu kebijakan utama, yang berisi pernyataan yang menentukan siapa yang dapat menggunakan kunci dan bagaimana mereka dapat menggunakannya. Saat membuat kunci terkelola pelanggan, Anda dapat menentukan kebijakan kunci. Untuk informasi selengkapnya, lihat Mengelola akses ke kunci yang dikelola pelanggan di Panduan AWS Key Management Service Pengembang.

Untuk menggunakan kunci terkelola pelanggan dengan Layanan Terkelola Amazon untuk ruang kerja Prometheus, operasi API berikut harus diizinkan dalam kebijakan kunci:

- `kms:CreateGrant`— Menambahkan hibah ke kunci yang dikelola pelanggan. Memberikan akses kontrol ke kunci KMS tertentu, yang memungkinkan akses untuk memberikan operasi yang diperlukan Amazon Managed Service untuk Prometheus. Untuk informasi selengkapnya, lihat Menggunakan Hibah di Panduan AWS Key Management Service Pengembang.

  Hal ini memungkinkan Amazon Managed Service untuk Prometheus untuk melakukan hal berikut:

  - Panggilan `GenerateDataKey` untuk menghasilkan kunci data terenkripsi dan menyimpannya, karena kunci data tidak segera digunakan untuk mengenkripsi.
  - Panggilan `Decrypt` untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.
- `kms:DescribeKey`— Memberikan detail kunci yang dikelola pelanggan untuk memungkinkan Layanan Terkelola Amazon untuk Prometheus memvalidasi kunci.

Berikut ini adalah contoh pernyataan kebijakan yang dapat Anda tambahkan untuk Amazon Managed Service for Prometheus:

```
 "Statement" : [
   {
     "Sid" : "Allow access to Amazon Managed Service for Prometheus principal within
your account",
     "Effect" : "Allow",
     "Principal" : {
```

```
        "AWS" : "*"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "kms:ViaService" : "aps.region.amazonaws.com",
          "kms:CallerAccount" : "111122223333"
        }
    },
    {
      "Sid": "Allow access for key administrators - not required for Amazon Managed
Service for Prometheus",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action" : [
        "kms:*"
       ],
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    <other statements needed for other non-Amazon Managed Service for Prometheus
scenarios>
 ]
```

- Untuk informasi selengkapnya tentang menentukan izin dalam kebijakan, lihat Panduan AWS Key Management ServicePengembang.
- Untuk informasi selengkapnya tentang akses kunci pemecahan masalah, lihat Panduan AWS Key Management ServicePengembang.

Langkah 2: Menentukan kunci yang dikelola pelanggan untuk Amazon Managed Service untuk Prometheus

Saat membuat ruang kerja, Anda dapat menentukan kunci yang dikelola pelanggan dengan memasukkan ARN Kunci KMS, yang digunakan Amazon Managed Service for Prometheus untuk mengenkripsi data yang disimpan oleh ruang kerja.

Langkah 3: Mengakses data dari layanan lain, seperti Grafana yang Dikelola Amazon

Langkah ini opsional — hanya diperlukan jika Anda perlu mengakses Layanan Terkelola Amazon untuk data Prometheus dari layanan lain.

Data terenkripsi Anda tidak dapat diakses dari layanan lain, kecuali mereka juga memiliki akses untuk menggunakan kunci. AWS KMS Misalnya, jika Anda ingin menggunakan Grafana Terkelola Amazon untuk membuat dasbor atau peringatan pada data Anda, Anda harus memberi Amazon Managed Grafana akses ke kunci tersebut.

Untuk memberi Amazon Managed Grafana akses ke kunci terkelola pelanggan Anda

1. Di daftar ruang kerja Amazon Managed Grafana, pilih nama untuk ruang kerja yang ingin Anda akses ke Amazon Managed Service for Prometheus. Ini menunjukkan kepada Anda informasi ringkasan tentang ruang kerja Grafana Terkelola Amazon Anda.
2. Perhatikan nama peran IAM yang digunakan oleh ruang kerja Anda. Namanya ada dalam format`AmazonGrafanaServiceRole-<unique-id>`. Konsol menunjukkan ARN lengkap untuk peran tersebut. Anda akan menentukan nama ini di AWS KMS konsol di langkah selanjutnya.
3. Dalam daftar kunci terkelola AWS KMS Pelanggan Anda, pilih kunci terkelola pelanggan yang Anda gunakan selama pembuatan Layanan Terkelola Amazon untuk ruang kerja Prometheus. Ini membuka halaman detail konfigurasi utama.
4. Di sebelah Pengguna utama, pilih tombol Tambah.
5. Dari daftar nama, pilih peran IAM Grafana Terkelola Amazon yang Anda sebutkan di atas. Untuk membuatnya lebih mudah ditemukan, Anda dapat mencari berdasarkan nama, juga.
6. Pilih Tambah untuk menambahkan peran IAM ke daftar pengguna Kunci.

Ruang kerja Grafana Terkelola Amazon Anda sekarang dapat mengakses data di Layanan Terkelola Amazon untuk ruang kerja Prometheus. Anda dapat menambahkan pengguna atau peran lain ke pengguna utama untuk mengaktifkan layanan lain mengakses ruang kerja Anda.

Layanan Terkelola Amazon untuk konteks enkripsi Prometheus

Konteks enkripsi adalah kumpulan opsional pasangan kunci-nilai yang berisi informasi kontekstual tambahan tentang data.

AWS KMSmenggunakan konteks enkripsi sebagai data otentikasi tambahan untuk mendukung enkripsi yang diautentikasi. Bila Anda menyertakan konteks enkripsi dalam permintaan untuk mengenkripsi data, AWS KMS mengikat konteks enkripsi ke data terenkripsi. Untuk mendekripsi data, Anda menyertakan konteks enkripsi yang sama dalam permintaan.

Layanan Terkelola Amazon untuk konteks enkripsi Prometheus

Amazon Managed Service untuk Prometheus menggunakan konteks enkripsi yang sama di AWS KMS semua operasi kriptografi, di mana kuncinya `aws:amp:arn` dan nilainya adalah Nama Sumber Daya Amazon (ARN) ruang kerja.

Example

```
"encryptionContext": {
    "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-sample-1234-
abcd-56ef-7890abcd12ef"
}
```

Menggunakan konteks enkripsi untuk pemantauan

Bila Anda menggunakan kunci terkelola pelanggan simetris untuk mengenkripsi data ruang kerja Anda, Anda juga dapat menggunakan konteks enkripsi dalam catatan audit dan log untuk mengidentifikasi bagaimana kunci yang dikelola pelanggan digunakan. Konteks enkripsi juga muncul di log yang dihasilkan oleh AWS CloudTrail atau Amazon CloudWatch Logs.

Menggunakan konteks enkripsi untuk mengontrol akses ke kunci terkelola pelanggan Anda

Anda dapat menggunakan konteks enkripsi dalam kebijakan utama dan kebijakan IAM `conditions` untuk mengontrol akses ke kunci terkelola pelanggan simetris Anda. Anda juga dapat menggunakan kendala konteks enkripsi dalam hibah.

Layanan Terkelola Amazon untuk Prometheus menggunakan batasan konteks enkripsi dalam hibah untuk mengontrol akses ke kunci yang dikelola pelanggan di akun atau wilayah Anda. Batasan hibah mengharuskan operasi yang diizinkan oleh hibah menggunakan konteks enkripsi yang ditentukan.

Example

Berikut ini adalah contoh pernyataan kebijakan kunci untuk memberikan akses ke kunci yang dikelola pelanggan untuk konteks enkripsi tertentu. Kondisi dalam pernyataan kebijakan ini mengharuskan hibah memiliki batasan konteks enkripsi yang menentukan konteks enkripsi.

```
{
    "Sid": "Enable DescribeKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:DescribeKey",
     "Resource": "*"
},
{
     "Sid": "Enable CreateGrant",
     "Effect": "Allow",
     "Principal": {
         "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:CreateGrant",
     "Resource": "*",
     "Condition": {
         "StringEquals": {
             "kms:EncryptionContext:aws:aps:arn": "arn:aws:aps:us-
west-2:111122223333:workspace/ws-sample-1234-abcd-56ef-7890abcd12ef"
          }
     }
}
```

Memantau kunci enkripsi Anda untuk Amazon Managed Service untuk Prometheus

Saat Anda menggunakan kunci terkelola AWS KMS pelanggan dengan Layanan Terkelola Amazon untuk ruang kerja Prometheus, Anda dapat menggunakan AWS CloudTrailatau CloudWatch Log Amazon untuk melacak permintaan yang dikirimkan oleh Layanan Terkelola Amazon untuk Prometheus. AWS KMS

Contoh berikut adalah AWS CloudTrail peristiwa untuk`CreateGrant`,, `GenerateDataKeyDecrypt`, dan `DescribeKey` untuk memantau operasi KMS yang dipanggil oleh Amazon Managed Service untuk Prometheus untuk mengakses data yang dienkripsi oleh kunci terkelola pelanggan Anda:

CreateGrant

Saat Anda menggunakan kunci yang dikelola AWS KMS pelanggan untuk mengenkripsi ruang kerja Anda, Amazon Managed Service for Prometheus mengirimkan tiga `CreateGrant` permintaan atas nama Anda untuk mengakses kunci KMS yang Anda tentukan. Hibah yang dibuat oleh Amazon Managed Service for Prometheus khusus untuk sumber daya yang terkait dengan kunci yang dikelola pelanggan. AWS KMS

Contoh peristiwa berikut mencatat `CreateGrant` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "TESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-KEY-ID1",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "TESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "aps.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "retiringPrincipal": "aps.region.amazonaws.com",
        "operations": [
```

```
            "GenerateDataKey",
            "Decrypt",
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "granteePrincipal": "aps.region.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey

Saat Anda mengaktifkan kunci terkelola AWS KMS pelanggan untuk ruang kerja Anda, Amazon Managed Service untuk Prometheus akan membuat kunci unik. Ini mengirimkan `GenerateDataKey` permintaan ke AWS KMS yang menentukan kunci yang dikelola AWS KMS pelanggan untuk sumber daya.

Contoh peristiwa berikut mencatat `GenerateDataKey` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "aps.amazonaws.com"
```

```
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-
sample-1234-abcd-56ef-7890abcd12ef"
        },
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "57f5dbee-16da-413e-979f-2c4c6663475e"
}
```

Decrypt

Saat kueri dibuat di ruang kerja terenkripsi, Amazon Managed Service for Prometheus memanggil `Decrypt` operasi untuk menggunakan kunci data terenkripsi yang disimpan untuk mengakses data terenkripsi.

Contoh peristiwa berikut mencatat `Decrypt` operasi:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "aps.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:10:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-
sample-1234-abcd-56ef-7890abcd12ef"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```

DescribeKey

Layanan Terkelola Amazon untuk Prometheus menggunakan operasi untuk memverifikasi apakah kunci `DescribeKey` terkelola pelanggan AWS KMS yang terkait dengan ruang kerja Anda ada di akun dan wilayah.

Contoh peristiwa berikut mencatat `DescribeKey` operasi:


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "TESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-KEY-ID1",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "TESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "aps.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
```

```
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

Pelajari selengkapnya

Sumber daya berikut memberikan informasi lebih lanjut tentang enkripsi data saat istirahat.

- Untuk informasi selengkapnya tentang konsep AWS Key Management Service dasar, lihat Panduan AWS Key Management Service Pengembang.
- Untuk informasi selengkapnya tentang praktik terbaik Keamanan AWS Key Management Service, lihat Panduan AWS Key Management Service Pengembang.

Identity and Access Management untuk Amazon Managed Service untuk Prometheus

(IAM) AWS Identity and Access Management adalah Layanan AWS yang membantu seorang administrator dalam mengendalikan akses ke sumber daya AWS secara aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan Layanan Terkelola Amazon untuk sumber daya Prometheus. IAM adalah sebuah layanan Layanan AWS yang dapat Anda gunakan tanpa dikenakan biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas

- Mengelola kebijakan menggunakan akses
- Bagaimana Amazon Managed Service untuk Prometheus bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus
- AWSkebijakan terkelola untuk Amazon Managed Service untuk Prometheus
- Memecahkan masalah Amazon Managed Service untuk identitas dan akses Prometheus

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon Managed Service untuk Prometheus.

Pengguna layanan - Jika Anda menggunakan Layanan Terkelola Amazon untuk layanan Prometheus untuk melakukan pekerjaan Anda, administrator akan memberi Anda kredensyal dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Layanan Terkelola Amazon untuk Prometheus untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon Managed Service untuk Prometheus, lihat. Memecahkan masalah Amazon Managed Service untuk identitas dan akses Prometheus

Administrator layanan - Jika Anda bertanggung jawab atas Layanan Terkelola Amazon untuk sumber daya Prometheus di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amazon Managed Service untuk Prometheus. Tugas Anda adalah menentukan fitur dan sumber daya Layanan Terkelola Amazon untuk Prometheus mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari selengkapnya tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Amazon Managed Service for Prometheus, lihat. Bagaimana Amazon Managed Service untuk Prometheus bekerja dengan IAM

Administrator IAM - Jika Anda administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Amazon Managed Service for Prometheus. Untuk melihat contoh Layanan Terkelola Amazon untuk kebijakan berbasis identitas Prometheus yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Mengautentikasi dengan identitas

Autentikasi merupakan cara Anda untuk masuk ke AWS dengan menggunakan kredensial identitas Anda. Anda harus terautentikasi (masuk keAWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk ke AWS sebagai identitas terfederasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Para pengguna (Pusat Identitas IAM), otentikasi sign-on tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda merupakan contoh identitas terfederasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas dengan menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil suatu peran.

Tergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal akses AWS. Untuk informasi selengkapnya tentang masuk ke AWS, silakan lihat Cara masuk ke Akun AWS Anda di Panduan Pengguna AWS Sign-In.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan peralatan AWS, maka Anda harus menandatangani sendiri permintaan tersebut. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, silakan lihat Menandatangani permintaan API AWS di Panduan Pengguna IAM.

Terlepas dari metode autentikasi yang Anda gunakan, Anda mungkin juga diminta untuk menyediakan informasi keamanan tambahan. Sebagai contoh, AWS menyarankan supaya Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, silakan lihat Autentikasi multi-faktor di Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS di Panduan Pengguna IAM.

Pengguna root Akun AWS

Ketika Anda membuat Akun AWS, Anda memulai dengan satu identitas masuk yang memiliki akses ke semua Layanan AWS dan sumber daya di akun tersebut. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk ke alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang

mengharuskan Anda masuk sebagai pengguna root, silakan lihat Tugas yang memerlukan kredensial pengguna root di Panduan Pengguna IAM.

Identitas terfederasi

Praktik terbaiknya berupa, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial temporer.

Identitas terfederasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, dikenal sebagai AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran memberikan kredensial temporer.

Untuk pengelolaan akses terpusat, kami sarankan Anda menggunakanAWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS Anda dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, silakan lihat Apakah Pusat Identitas IAM itu? di User Guide AWS IAM Identity Center.

Pengguna dan Grup IAM

Pengguna IAM adalah identitas dalam Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Apabila memungkinkan, kami menyarankan untuk mengandalkan pada kredensial temporer alih-alih membuat pengguna IAM yang memiliki kredensyal jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami menyarankan Anda memutar kunci akses. Untuk informasi selengkapnya, silakan lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang di Panduan Pengguna IAM.

Grup IAM adalah identitas yang menerangkan secara spesifik kumpulan pengguna IAM. Anda tidak dapat masuk sebagai kelompok. Anda dapat menggunakan grup untuk menerangkan secara spesifik izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Sebagai contoh, Anda dapat memiliki grup yang diberi nama AdminIAM dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran tersebut dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial

temporer. Untuk mempelajari selengkapnya, silakan lihat Kapan harus membuat pengguna IAM (alih-alih peran) di Panduan Pengguna IAM.

Peran IAM

Peran IAM merupakan identitas dalam Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat menggunakan peran IAM untuk sementara dalam AWS Management Console dengan berganti peran. Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, silakan lihat menggunakan peran IAM di Panduan Pengguna IAM.

IAM role dengan kredensial temporer berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas terfederasi, Anda harus membuat sebuah peran dan menentukan izin untuk peran tersebut. Ketika identitas gabungan terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran-peran untuk federasi, silakan lihat Membuat sebuah peran untuk Penyedia Identitas pihak ketiga di Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi serangkain izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengkorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, silakan lihat Rangkaian izin di Panduan Pengguna AWS IAM Identity Center.
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM untuk sementara mengambil izin berbeda untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) di akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, pada beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan suatu peran sebagai proksi). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat Bagaimana peran IAM role berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan – Sebagian Layanan AWS menggunakan fitur di Layanan AWS lainnya. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, lazim pada layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran tertaut layanan.

- Sesi akses maju (FAS) – Ketika Anda menggunakan pengguna IAM atau peran IAM untuk melakukan tindakan-tindakan di AWS, Anda akan dianggap sebagai seorang pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat Meneruskan sesi akses.
- Peran layanan – Sebuah peran layanan adalah sebuah peran IAM yang dijalankan oleh suatu layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, silakan lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran tertaut layanan – Peran tertaut layanan adalah tipe peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.

- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan peran IAM untuk mengelola kredensial temporer untuk aplikasi yang berjalan di instans EC2 dan mengajukan permintaan AWS CLI atau API AWS. Cara ini lebih baik daripada menyimpan kunci akses dalam instans EC2. Untuk menugaskan sebuah peran AWS ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda dapat membuat sebuah profil instans yang dilampirkan ke instans. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, silakan lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 di Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, silakan lihat Kapan harus membuat peran IAM (alih-alih pengguna) di Panduan Pengguna IAM.

Mengelola kebijakan menggunakan akses

Anda mengendalikan akses di AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya AWS. Kebijakan adalah objek di AWS yang, ketika terkait dengan identitas atau

sumber daya, akan menentukan izinnya. AWS mengevaluasi kebijakan-kebijakan tersebut ketika seorang pengguna utama (pengguna, root user, atau sesi peran) mengajukan permintaan. Izin dalam kebijakan menentukan apakah permintaan diberikan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, silakan lihat Gambaran Umum kebijakan JSON di Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Secara bawaan, para pengguna dan peran tidak memiliki izin. Untuk mengabulkan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk pengoperasiannya. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole` . Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau APIAWS.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline ditanam secara langsung ke pengguna tunggal, grup, atau peran. Kebijakan terkelola adalah kebijakan yang berdiri sendiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS Anda. Kebijakan terkelola mencakup kebijakan terkelola AWS dan kebijakan terkelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, silakan lihat Memilih antara kebijakan terkelola dan kebijakan inline diPanduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan

kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentkan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan terkelola AWS dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh-contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Ringkas Amazon.

Tipe-tipe kebijakan lain

AWS mendukung tipe kebijakan tambahan, yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batas izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini menindihi izin. Untuk informasi selengkapnya tentang batasan izin, silakan lihat Batasan izin untuk entitas IAM di Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan secara terpusat mengelola beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau ke semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun

AWS. Untuk informasi selengkapnya tentang Organisasi dan SCP, silakan lihat Cara kerja SCP di Panduan Pengguna AWS Organizations.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini menindihi izin. Untuk informasi selengkapnya, silakan lihat Kebijakan sesi di Panduan Pengguna IAM.

Berbagai tipe kebijakan

Ketika beberapa tipe kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan ketika beberapa tipe kebijakan dilibatkan, silakan lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana Amazon Managed Service untuk Prometheus bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon Managed Service untuk Prometheus, pelajari fitur IAM yang tersedia untuk digunakan dengan Amazon Managed Service for Prometheus.

Fitur IAM yang dapat Anda gunakan dengan Amazon Managed Service untuk Prometheus

Fitur IAM	Layanan Terkelola Amazon untuk dukungan Prometheus
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Tidak

Fitur IAM	Layanan Terkelola Amazon untuk dukungan Prometheus
ACL	Tidak
ABAC (tag dalam kebijakan)	Ya
Kredensial temporer	Ya
Sesi akses teruskan (FAS)	Tidak
Peran layanan	Tidak
Peran tertaut layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Layanan Terkelola Amazon untuk Prometheus dan layanan AWS lainnya dengan sebagian besar fitur IAM, AWSlihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Mendukung kebijakan berbasis identitas	Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta persyaratan yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik pengguna utama dalam sebuah kebijakan berbasis identitas karena pengguna utama berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, silakan lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Untuk melihat contoh Layanan Terkelola Amazon untuk kebijakan berbasis identitas Prometheus, lihat. Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Kebijakan berbasis sumber daya dalam Amazon Managed Service untuk Prometheus

Mendukung kebijakan berbasis sumber daya	Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentkan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika pengguna utama dan sumber daya berada dalam Akun AWS yang berbeda, Administrator IAM di akun tepercaya juga harus memberikan izin kepada entitas pengguna utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, silakan lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon Managed Service untuk Prometheus

Mendukung tindakan kebijakan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan-tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan-tindakan kebijakan biasanya memiliki nama yang sama sebagaimana operasi API AWS yang dikaitkan padanya. Ada beberapa pengecualian, misalnya tindakan yang memiliki izin saja yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam sebuah kebijakan. Tindakan-tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Untuk melihat daftar Layanan Terkelola Amazon untuk tindakan Prometheus, lihat Tindakan yang ditentukan oleh Amazon Managed Service for Prometheus di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon Managed Service untuk Prometheus menggunakan awalan berikut sebelum tindakan:

```
aps
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut dengan koma.

```
"Action": [
      "aps:action1",
      "aps:action2"
         ]
```

Untuk melihat contoh Layanan Terkelola Amazon untuk kebijakan berbasis identitas Prometheus, lihat. Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Sumber daya kebijakan untuk Amazon Managed Service untuk Prometheus

Mendukung sumber daya kebijakan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan entah elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan-tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk mengindikasikan bahwa pernyataan tersebut berlaku bagi semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar Layanan Terkelola Amazon untuk jenis sumber daya Prometheus dan ARNnya, lihat Sumber daya yang ditentukan oleh Amazon Managed Service for Prometheus di Referensi Otorisasi Layanan. Untuk mempelajari tindakan yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang ditentukan oleh Amazon Managed Service for Prometheus.

Untuk melihat contoh Layanan Terkelola Amazon untuk kebijakan berbasis identitas Prometheus, lihat. Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Kunci kondisi kebijakan untuk Amazon Managed Service untuk Prometheus

Mendukung kunci-kunci persyaratan kebijakan spesifik layanan	Tidak

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan syarat yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator syarat, misalnya sama dengan atau kurang dari, untuk mencocokkan syarat dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya dengan menggunakan

operasi AND yang logis. Jika Anda menentukan beberapa niilai untuk satu kunci persyaratan, maka AWS akan mengevaluasi syarat tersebut menggunakan operasi OR yang logis. Semua persyaratan harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan syarat. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat Elemen kebijakan IAM: variabel dan tag di Panduan Pengguna IAM.

AWS mendukung kunci-kunci syarat global dan kunci-kunci syarat spesifik layanan. Untuk melihat semua kunci persyaratan global AWS, silakan lihat kunci konteks syarat global AWS di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Layanan Terkelola Amazon untuk Prometheus, lihat Kunci kondisi untuk Layanan Terkelola Amazon untuk Prometheus di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang ditentukan oleh Amazon Managed Service for Prometheus.

Untuk melihat contoh Layanan Terkelola Amazon untuk kebijakan berbasis identitas Prometheus, lihat. Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Daftar kontrol akses (ACL) di Amazon Managed Service untuk Prometheus

Mendukung ACL	Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan Amazon Managed Service untuk Prometheus

Mendukung ABAC (tanda dalam kebijakan)	Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Di AWS, atribut-atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna

atau peran) dan ke banyak sumber daya AWS. Pemberian tag ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi-operasi ketika tag milik pengguna utama cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi dimana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen syarat dari sebuh kebijakan dengan menggunakan kunci-kunci persyaratan `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci-kunci persyaratan untuk setiap jenis sumber daya, maka nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci persyaratan untuk hanya beberapa jenis sumber daya, maka nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, silakan lihat Apa itu ABAC? di Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, silakan lihat Menggunakan kontrol akses berbasis atribut (ABAC) di Panduan Pengguna IAM.

Menggunakan kredensyal sementara dengan Amazon Managed Service untuk Prometheus

Mendukung kredensial temporer	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk dengan menggunakan kredensial temporer. Sebagai informasi tambahan, termasuk tentang Layanan AWS mana saja yang berfungsi dengan kredensial temporer, silakan lihat Layanan AWS yang berfungsi dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensial temporer jika Anda masuk ke AWS Management Console dengan menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Sebagai contoh, ketika Anda mengakses AWS dengan menggunakan tautan masuk tunggal (SSO) milik perusahaan Anda, proses itu secara otomatis akan membuat kredensial temporer. Anda juga akan secara otomatis membuat kredensial temporer ketika Anda masuk ke konsol sebagai seorang pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang peralihan peran, silakan lihat Peralihan peran (konsol) di Panduan Pengguna IAM.

Anda dapat secara manual membuat kredensial temporer menggunakan AWS CLI atau API AWS. Anda kemudian dapat menggunakan kredensial temporer tersebut untuk mengakses AWS. AWS menyarankan agar Anda secara dinamis membuat kredensial temporer alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, silakan lihat Kredensial keamanan temporer di IAM.

Teruskan sesi akses untuk Amazon Managed Service untuk Prometheus

Mendukung sesi akses maju (FAS)	Tidak

Saat Anda menggunakan pengguna IAM atau peran IAM untuk mengerjakan tindakan di AWS, Anda akan dianggap sebagai pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat Meneruskan sesi akses.

Peran layanan untuk Amazon Managed Service untuk Prometheus

Mendukung peran layanan	Tidak

Peran layanan adalah sebuah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas Layanan Terkelola Amazon untuk Prometheus. Edit peran layanan hanya jika Amazon Managed Service untuk Prometheus memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Mendukung peran yang terhubung dengan layanan	Ya

Peran yang tertaut layanan adalah jenis peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.

Untuk detail tentang membuat atau mengelola Layanan Terkelola Amazon untuk peran terkait layanan Prometheus, lihat. Menggunakan peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Contoh kebijakan berbasis identitas untuk Amazon Managed Service untuk Prometheus

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi Layanan Terkelola Amazon untuk sumber daya Prometheus. Pengguan dan peran tersebut juga tidak dapat melakukan tugas dengan menggunakan API AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS. Untuk mengabulkan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, silakan lihat Membuat kebijakan IAM di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Layanan Terkelola Amazon untuk Prometheus, termasuk format ARN untuk setiap jenis sumber daya, lihat Tindakan, sumber daya, dan kunci kondisi untuk Layanan Terkelola Amazon untuk Prometheus dalam Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan Amazon Managed Service untuk konsol Prometheus
- Perbolehkan pengguna untuk melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus Layanan Terkelola Amazon untuk sumber daya Prometheus di akun Anda. Tindakan ini mengenakan biaya kepada Anda Akun AWS. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan terkelola AWS dan beralih ke izin dengan hak akses paling rendah – Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan terkelola AWS yang memberikan izin untuk banyak kasus penggunaan umum. Kebijakan tedapat di Akun AWS Anda. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan AWS yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, silakan lihat kebijakan-kebijakan terkelola AWS atau kebijakan-kebijakan terkelola AWS untuk fungsi tugas di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan pengguna IAM untuk mengajukan izin, silakan lihat Kebijakan dan izin di IAM di Panduan Pengguna IAM.
- Gunakan syarat dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu syarat ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan syarat untuk memberi akses ke tindakan layanan jika digunakan melalui Layanan AWS yang spesifk, sepertiAWS CloudFormation. Untuk informasi selengkapnya, silakan lihat Elemen kebijakan JSON IAM: Syarat di Panduan Pengguna IAM.
- Gunakan Analizer Akses IAM untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – Analizer Akses IAM memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. Analizer Akses IAM menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, silakan lihat validasi kebijakan Analizer Akses IAM di Panduan Pengguna IAM.
- Memerlukan autentikasi multi-faktor (MFA) – Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Akun AWS Anda, aktifkan MFA untuk keamanan tambahan.

Untuk meminta MFA ketika operasi API dipanggil, tambahkan syarat MFA pada kebijakan Anda. Untuk informasi selengkapnya, silakan lihat Mengonfigurasi akses API yang diproteksi MFA di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, silakan lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Menggunakan Amazon Managed Service untuk konsol Prometheus

Untuk mengakses Amazon Managed Service untuk konsol Prometheus, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang Layanan Terkelola Amazon untuk sumber daya Prometheus di sumber daya Anda. Akun AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu meloloskan izin konsol minimum bagi pengguna yang hanya melakukan panggilan ke API AWS CLI atau AWS. Jika tidak, akses hanya diizinkan ke tindakan-tindakan yang sesuai dengan operasi API yang sedang mereka coba lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan Layanan Terkelola Amazon untuk konsol Prometheus, lampirkan juga Layanan Terkelola Amazon untuk `ConsoleAccess` Prometheus atau kebijakan terkelola ke entitas. `ReadOnly` AWS Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna dalam Panduan Pengguna IAM.

Perbolehkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda dapat membuat kebijakan yang mengizinkan para pengguna IAM untuk melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan pada konsol atau secara terprogram menggunakan API AWS CLI atau AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
```

```
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWSkebijakan terkelola untuk Amazon Managed Service untuk Prometheus

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

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

Anda tidak dapat mengubah izin yang ada dalam kebijakan-kebijakan terkelola AWS. Jika AWS memperbarui izin yang ditentukan dalam sebuah kebijakan terkelola AWS, maka pembaruan itu akan mempengaruhi semua identitas pengguna utama (pengguna, grup, dan peran) yang terkait dengan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan terkelola AWS

saat sebuah Layanan AWS baru diluncurkan atau operasi API baru tersedia untuk layanan yang sudah ada.

Untuk informasi selengkapnya, silakan lihat kebijakan terkelola AWS di Panduan Pengguna IAM.

AmazonPrometheusFullAccess

Anda dapat melampirkan kebijakan `AmazonPrometheusFullAccess` ke identitas-identitas IAM Anda.

Detail izin

Kebijakan ini mencakup izin berikut.

- `aps`— Memungkinkan akses penuh ke Amazon Managed Service untuk Prometheus
- `eks`— Memungkinkan Layanan Terkelola Amazon untuk layanan Prometheus membaca informasi tentang kluster Amazon EKS Anda. Ini diperlukan untuk memungkinkan pembuatan pencakar terkelola dan menemukan metrik di cluster Anda.
- `ec2`— Memungkinkan Layanan Terkelola Amazon untuk layanan Prometheus membaca informasi tentang jaringan Amazon EC2 Anda. Ini diperlukan untuk memungkinkan pembuatan pencakar terkelola dengan akses ke metrik Amazon EKS Anda.
- `iam`— Memungkinkan kepala sekolah untuk membuat peran terkait layanan untuk pencakar metrik terkelola.

Isi dari AmazonPrometheusFullAccessadalah sebagai berikut:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllPrometheusActions",
   "Effect": "Allow",
   "Action": [
    "aps:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DescribeCluster",
   "Effect": "Allow",
   "Action": [
```

```
    "eks:DescribeCluster",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "aps.amazonaws.com"
     ]
    }
   },
   "Resource": "*"
  },
  {
   "Sid": "CreateServiceLinkedRole",
   "Effect": "Allow",
   "Action": "iam:CreateServiceLinkedRole",
   "Resource": "arn:aws:iam::*:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper*",
   "Condition": {
    "StringEquals": {
     "iam:AWSServiceName": "scraper.aps.amazonaws.com"
    }
   }
  }
 ]
}
```

AmazonPrometheusConsoleFullAccess

Anda dapat melampirkan kebijakan `AmazonPrometheusConsoleFullAccess` ke identitas-identitas IAM Anda.

Detail izin

Kebijakan ini mencakup izin berikut.

- `aps`— Memungkinkan akses penuh ke Amazon Managed Service untuk Prometheus
- `tag`— Memungkinkan kepala sekolah melihat saran tag di Amazon Managed Service untuk konsol Prometheus.

```
{
```

```
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "TagSuggestions",
   "Effect": "Allow",
   "Action": [
    "tag:GetTagValues",
    "tag:GetTagKeys"
   ],
   "Resource": "*"
  },
  {
   "Sid": "PrometheusConsoleActions",
   "Effect": "Allow",
   "Action": [
    "aps:CreateWorkspace",
    "aps:DescribeWorkspace",
    "aps:UpdateWorkspaceAlias",
    "aps:DeleteWorkspace",
    "aps:ListWorkspaces",
    "aps:DescribeAlertManagerDefinition",
    "aps:DescribeRuleGroupsNamespace",
    "aps:CreateAlertManagerDefinition",
    "aps:CreateRuleGroupsNamespace",
    "aps:DeleteAlertManagerDefinition",
    "aps:DeleteRuleGroupsNamespace",
    "aps:ListRuleGroupsNamespaces",
    "aps:PutAlertManagerDefinition",
    "aps:PutRuleGroupsNamespace",
    "aps:TagResource",
    "aps:UntagResource",
    "aps:CreateLoggingConfiguration",
    "aps:UpdateLoggingConfiguration",
    "aps:DeleteLoggingConfiguration",
    "aps:DescribeLoggingConfiguration"
   ],
   "Resource": "*"
  }
 ]
}
```

AmazonPrometheusRemoteWriteAccess

Isi dari AmazonPrometheusRemoteWriteAccessadalah sebagai berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aps:RemoteWrite"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AmazonPrometheusQueryAccess

Isi dari AmazonPrometheusQueryAccessadalah sebagai berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aps:GetLabels",
                "aps:GetMetricMetadata",
                "aps:GetSeries",
                "aps:QueryMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWSkebijakan terkelola: AmazonPrometheusScraperServiceLinkedRolePolicy

Anda tidak dapat melampirkan AmazonPrometheusScraperServiceLinkedRolePolicy ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Layanan Terkelola

Amazon untuk Prometheus melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat Menggunakan peran untuk mengikis metrik dari EKS.

Kebijakan ini memberikan izin kontributor yang memungkinkan membaca dari klaster Amazon EKS Anda dan menulis ke ruang kerja Layanan Terkelola Amazon untuk Prometheus.

Detail izin

Kebijakan ini mencakup izin berikut.

- `aps`— Memungkinkan kepala layanan untuk menulis metrik ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.
- `ec2`— Memungkinkan kepala layanan untuk membaca dan memodifikasi konfigurasi jaringan untuk terhubung ke jaringan yang berisi kluster Amazon EKS Anda.
- `eks`— Memungkinkan kepala layanan untuk mengakses kluster Amazon EKS Anda. Ini diperlukan agar dapat secara otomatis mengikis metrik.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DeleteSLR",
   "Effect": "Allow",
   "Action": [
    "iam:DeleteRole"
   ],
   "Resource": "arn:aws:iam::*:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper*"
  },
  {
   "Sid": "NetworkDiscovery",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ENIManagement",
```

```
    "Effect": "Allow",
    "Action": "ec2:CreateNetworkInterface",
    "Resource": "*",
    "Condition": {
     "ForAllValues:StringEquals": {
      "aws:TagKeys": [
       "AMPAgentlessScraper"
      ]
     }
    }
   },
   {
    "Sid": "TagManagement",
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:*:ec2:*:*:network-interface/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
     },
     "Null": {
      "aws:RequestTag/AMPAgentlessScraper": "false"
     }
    }
   },
   {
    "Sid": "ENIUpdating",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteNetworkInterface",
     "ec2:ModifyNetworkInterfaceAttribute"
    ],
    "Resource": "*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AMPAgentlessScraper": "false"
     }
    }
   },
   {
    "Sid": "EKSAccess",
    "Effect": "Allow",
    "Action": "eks:DescribeCluster",
    "Resource": "arn:*:eks:*:*:cluster/*"
```

```
  },
  {
   "Sid": "APSWriting",
   "Effect": "Allow",
   "Action": "aps:RemoteWrite",
   "Resource": "arn:*:aps:*:*:workspace/*",
   "Condition": {
    "StringEquals": {
     "aws:PrincipalAccount": "${aws:ResourceAccount}"
    }
   }
  }
 ]
}
```

Layanan Terkelola Amazon untuk Prometheus memperbarui kebijakan terkelola AWS

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Layanan Terkelola Amazon untuk Prometheus sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Amazon Managed Service for Prometheus Document.

Perubahan	Deskripsi	Tanggal
AmazonPrometheusFullAccess — Permbaruan ke kebijakan yang sudah ada	Layanan Terkelola Amazon untuk Prometheus menambahkan izin baru untuk mendukung pembuatan pencakar terkelola AmazonPrometheusFullAccessuntuk metrik di kluster Amazon EKS. Termasuk izin untuk menghubungkan ke kluster Amazon EKS, membaca jaringan Amazon EC2, dan membuat peran terkait layanan untuk pencakar.	26 November 2023

Perubahan	Deskripsi	Tanggal
AmazonPrometheusScraperServiceLinkedRolePolicy – Kebijakan baru	Layanan Terkelola Amazon untuk Prometheus menambahkan kebijakan peran terkait layanan baru untuk dibaca dari kontainer Amazon EKS, untuk memungkinkan pengikisan metrik secara otomatis. Termasuk izin untuk menghubungkan ke kluster Amazon EKS, membaca jaringan Amazon EC2, dan membuat dan menghapus jaringan yang diberi tag, `AMPAgentlessScraper` serta untuk menulis ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.	26 November 2023

Perubahan	Deskripsi	Tanggal
AmazonPrometheusConsoleFullAccess – Pembaruan ke kebijakan yang sudah ada	Amazon Managed Service untuk Prometheus menambahkan izin baru untuk mendukung pengelola peringatan pencatatan dan AmazonPrometheusConsoleFullAccessperistiwa penggaris di Log. CloudWatch `aps:DescribeLoggingConfiguration` Izin `aps:CreateLoggingConfiguration` `aps:UpdateLoggingConfigurat ion aps:DeleteLoggingConfiguration` ,, ditambahkan.	Oktober 24, 2022

Perubahan	Deskripsi	Tanggal
AmazonPrometheusConsoleFullAccess – Pembaruan ke kebijakan yang sudah ada	Layanan Terkelola Amazon untuk Prometheus menambahkan izin baru untuk mendukung Layanan Terkelola Amazon baru AmazonPrometheusConsoleFullAccessuntuk fitur Prometheus dan agar pengguna dengan kebijakan ini dapat melihat daftar saran tag saat mereka menerapkan tag ke Layanan Terkelola Amazon untuk sumber daya Prometheus. `aps:UntagResource` Izin `tag:GetTagKeys` `tag:GetTagValues` `aps:CreateAlertManagerDefinition` ,`aps:CreateRuleGroupsNamespace` ,`aps:DeleteAlertManagerDefinition` ,`aps:DeleteRuleGroupsNamespace` , `aps:DescribeAlertManagerDefinition` `aps:DescribeRuleGroupsNamespace` ,`aps:ListRuleGroupsNamespaces` ,`aps:PutAlertManagerDefinition` ,`aps:PutRuleGroupsN`	29 September 2021

Perubahan	Deskripsi	Tanggal
	`amespace ,aps:TagRe source ,,,` dan ditambahkan.	
Amazon Managed Service untuk Prometheus mulai melacak perubahan	Layanan Terkelola Amazon untuk Prometheus mulai melacak perubahan untuk kebijakan yang dikelola. AWS	15 September 2021

Memecahkan masalah Amazon Managed Service untuk identitas dan akses Prometheus

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon Managed Service untuk Prometheus dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di Amazon Managed Service untuk Prometheus
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses Layanan Terkelola Amazon saya untuk sumber daya Prometheus

Saya tidak berwenang untuk melakukan tindakan di Amazon Managed Service untuk Prometheus

Jika Anda menerima pesan galat bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh galat berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *`my-example-widget`* rekaan, tetapi tidak memiliki izin `aps:`*`GetWidget`* rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aps:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya *`my-example-widget`* dengan menggunakan tindakan `aps:GetWidget`.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon Managed Service for Prometheus.

Sebagian Layanan AWS mengizinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut-layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon Managed Service untuk Prometheus. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses Layanan Terkelola Amazon saya untuk sumber daya Prometheus

Anda dapat membuat peran yang dapat digunakan para pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses kepada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mengetahui apakah Amazon Managed Service for Prometheus mendukung fitur-fitur ini, lihat. Bagaimana Amazon Managed Service untuk Prometheus bekerja dengan IAM
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, silakan lihat Menyediakan akses ke pengguna IAM di Akun AWS lainnya yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke pihak ketiga Akun AWS, silakan lihat Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, silakan lihat Menyediakan akses ke pengguna terautentikasi eksternal (gabungan identitas) di Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan IAM role dan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Izin dan kebijakan IAM

Akses ke Layanan Terkelola Amazon untuk tindakan dan data Prometheus memerlukan kredensil. Kredensil tersebut harus memiliki izin untuk melakukan tindakan dan mengakses AWS sumber daya, seperti mengambil data Amazon Managed Service untuk Prometheus tentang sumber daya cloud Anda. Bagian berikut memberikan detail tentang bagaimana Anda dapat menggunakan AWS Identity and Access Management (IAM) dan Layanan Terkelola Amazon untuk Prometheus untuk membantu mengamankan sumber daya Anda, dengan mengontrol siapa yang dapat mengaksesnya. Untuk informasi selengkapnya, lihat Kebijakan dan izin di IAM.

Layanan Terkelola Amazon untuk izin Prometheus

Tabel berikut menampilkan kemungkinan Layanan Terkelola Amazon untuk tindakan Prometheus dan izin yang diperlukan. Tindakan mungkin juga memerlukan izin dari layanan lain, tidak dirinci di sini.

Tindakan	Izin yang diperlukan
Buat peringatan.	`aps:CreateAlertManagerAlerts`

Tindakan	Izin yang diperlukan
Buat definisi manajer peringatan di ruang kerja. Untuk informasi selengkapnya, lihat Manajer Peringatan.	`aps:CreateAlertManagerDefinition`
Buat namespace grup aturan di ruang kerja. Untuk informasi selengkapnya, lihat Merekam aturan dan aturan peringatan.	`aps:CreateRuleGroupsNamespace`
Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus. Ruang kerja adalah ruang logis yang didedikasikan untuk penyimpanan dan kueri metrik Prometheus.	`aps:CreateWorkspace`
Hapus definisi manajer peringatan dari ruang kerja.	`aps:DeleteAlertManagerDefinition`
Hapus keheningan peringatan.	`aps:DeleteAlertManagerSilence`
Hapus Layanan Terkelola Amazon untuk ruang kerja Prometheus.	`aps:DeleteWorkspace`
Ambil informasi rinci tentang definisi manajer peringatan.	`aps:DescribeAlertManagerDefinition`
Ambil informasi rinci tentang ruang nama grup aturan.	`aps:DescribeRuleGroupsNamespace`
Ambil informasi terperinci tentang Layanan Terkelola Amazon untuk ruang kerja Prometheus.	`aps:DescribeWorkspace`
Ambil informasi rinci tentang keheningan peringatan.	`aps:GetAlertManagerSilence`
Ambil status manajer peringatan di ruang kerja.	`aps:GetAlertManagerStatus`

Tindakan	Izin yang diperlukan
Ambil label.	`aps:GetLabels`
Ambil metadata untuk Amazon Managed Service untuk metrik Prometheus.	`aps:GetMetricMetadata`
Ambil data deret waktu.	`aps:GetSeries`
Ambil daftar grup peringatan yang ditentukan dalam definisi manajer peringatan.	`aps:ListAlertManagerAlertGroups`
Ambil daftar peringatan yang ditentukan di manajer peringatan.	`aps:ListAlertManagerAlerts`
Ambil daftar penerima yang ditentukan dalam definisi manajer peringatan.	`aps:ListAlertManagerReceivers`
Ambil daftar keheningan peringatan yang ditentukan.	`aps:ListAlertManagerSilences`
Ambil daftar peringatan aktif.	`aps:ListAlerts`
Ambil daftar aturan di ruang nama grup aturan di ruang kerja Anda.	`aps:ListRules`
Ambil daftar ruang nama grup aturan di ruang kerja Anda.	`aps:ListRuleGroupsNamespaces`
Ambil tag yang terkait dengan Layanan Terkelola Amazon Anda untuk sumber daya Prometheus.	`aps:ListTagsForResource`
Ambil daftar Layanan Terkelola Amazon untuk ruang kerja Prometheus yang ada di akun.	`aps:ListWorkspaces`
Perbarui definisi manajer peringatan yang ada di ruang kerja.	`aps:PutAlertManagerDefinition`

Tindakan	Izin yang diperlukan
Buat keheningan peringatan.	`aps:PutAlertManagerSilences`
Perbarui namespace grup aturan yang ada.	`aps:PutRuleGroupsNamespace`
Jalankan kueri di Amazon Managed Service untuk metrik Prometheus.	`aps:QueryMetrics`
Lakukan operasi penulisan jarak jauh untuk memulai streaming metrik dari server Prometheus ke Amazon Managed Service untuk Prometheus.	`aps:RemoteWrite`
Tetapkan tag ke Amazon Managed Service untuk sumber daya Prometheus.	`aps:TagResource`
Hapus tag dari Amazon Managed Service untuk sumber daya Prometheus.	`aps:UntagResource`
Ubah alias ruang kerja yang ada.	`aps:UpdateWorkspaceAlias`
Buat konfigurasi logging.	`aps:CreateLoggingConfiguration`
Hapus konfigurasi logging.	`aps:DeleteLoggingConfiguration`
Jelaskan konfigurasi pencatatan ruang kerja.	`aps:DescribeLoggingConfiguration`
Perbarui konfigurasi logging.	`aps:UpdateLoggingConfiguration`

Contoh kebijakan IAM

Bagian ini memberikan contoh kebijakan lain yang dikelola sendiri yang dapat Anda buat.

Kebijakan IAM berikut memberikan akses penuh ke Amazon Managed Service untuk Prometheus dan juga memungkinkan pengguna untuk menemukan kluster Amazon EKS dan melihat detailnya.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:*",
                "eks:DescribeCluster",
                "eks:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

Validasi Kepatuhan untuk Layanan Terkelola Amazon untuk Prometheus

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan khusus, lihat Layanan AWS di Scope oleh Program Program Kepatuhan yang Anda minati. Untuk informasi umum, silakan lihat Program Kepatuhan AWS .

Anda bisa mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.
- Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

> **Note**
>
> Tidak semua Layanan AWS memenuhi syarat HIPAA. Untuk informasi selengkapnya, lihat Referensi Layanan yang Memenuhi Syarat HIPAA.

- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Panduan Kepatuhan Pelanggan AWS – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- AWS Security Hub – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dadlam AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda dan untuk memeriksa kepatuhan terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- AWS Audit Manager – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan bagaimana Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

Ketahanan dalam Layanan Terkelola Amazon untuk Prometheus

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan jaringan berlatensi rendah, throughput yang tinggi, dan sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Availability Zone lebih tersedia, memiliki toleransi kesalahan, dan dapat diskalakan dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Zona Ketersediaan, silakan lihat Infrastruktur Global AWS.

Selain infrastruktur AWS global, Amazon Managed Service for Prometheus menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan pencadangan Anda, termasuk dukungan untuk data ketersediaan tinggi.

Keamanan Infrastruktur di Amazon Managed Service untuk Prometheus

Sebagai layanan terkelola, Amazon Managed Service untuk Prometheus dilindungi oleh keamanan jaringan global. AWS Untuk informasi tentang layanan keamanan AWS dan cara AWS melindungi infrastruktur, lihat Keamanan Cloud AWS. Guna mendesain lingkungan AWS Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja AWS Well-Architected Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon Managed Service untuk Prometheus melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Cipher cocok dengan perfect forward secrecy (PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Menggunakan peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Layanan Terkelola Amazon untuk Prometheus AWS Identity and Access Management menggunakan peran terkait layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Amazon Managed Service untuk Prometheus. Peran terkait layanan telah ditentukan sebelumnya oleh Amazon Managed Service untuk Prometheus dan menyertakan semua izin yang diperlukan layanan untuk memanggil layanan lain atas nama Anda. AWS

Peran terkait layanan membuat pengaturan Amazon Managed Service untuk Prometheus lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Layanan Terkelola Amazon untuk Prometheus mendefinisikan izin peran terkait layanannya, dan kecuali ditentukan lain, hanya Layanan Terkelola Amazon untuk Prometheus yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Menggunakan peran untuk mengikis metrik dari EKS

Saat secara otomatis mengikis metrik menggunakan Amazon Managed Service untuk kolektor terkelola Prometheus, peran AWSServiceRoleForAmazonPrometheusScraper terkait layanan digunakan untuk mempermudah pengaturan kolektor terkelola, karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Layanan Terkelola Amazon untuk Prometheus mendefinisikan izin, dan hanya Layanan Terkelola Amazon untuk Prometheus yang dapat mengambil peran tersebut.

Untuk informasi tentang layanan lain yang mendukung peran tertaut layanan, silakan lihat layanan AWS yang bisa digunakan dengan IAM dan carilah layanan yang memiliki opsi Ya di kolom Peran tertaut layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Layanan Terkelola Amazon untuk Prometheus menggunakan peran terkait layanan yang diberi nama dengan awalan untuk AWSServiceRoleForAmazonPrometheusScrapermemungkinkan Layanan Terkelola Amazon untuk Prometheus mengikis metrik secara otomatis di kluster Amazon EKS Anda.

Peran AWSServiceRoleForAmazonPrometheusScraper terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `scraper.aps.amazonaws.com`

Kebijakan izin peran bernama AmazonPrometheusScraperServiceLinkedRolePolicymemungkinkan Layanan Terkelola Amazon untuk Prometheus menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Siapkan dan ubah konfigurasi jaringan untuk terhubung ke jaringan yang berisi kluster Amazon EKS Anda.

- Baca metrik dari kluster Amazon EKS dan tulis metrik ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat peran terkait layanan. Untuk informasi selengkapnya, silakan lihat Izin peran tertaut layanan di Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Anda tidak perlu membuat peran tertaut layanan secara manual. Saat Anda membuat instance kolektor terkelola menggunakan Amazon EKS atau Amazon Managed Service untuk Prometheus di, AWS Management Console the, atau AWS API, AWS CLI Amazon Managed Service for Prometheus membuat peran terkait layanan untuk Anda.

> **Important**
>
> Peran tertaut layanan ini dapat muncul di akun Anda jika Anda menyelesaikan tindakan di layanan lain yang menggunakan fitur yang disupport oleh peran ini. Untuk mempelajari lebih lanjut, lihat Peran baru muncul di saya Akun AWS.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat instance kolektor terkelola menggunakan Amazon EKS atau Amazon Managed Service untuk Prometheus, Amazon Managed Service for Prometheus membuat peran terkait layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Layanan Terkelola Amazon untuk Prometheus tidak memungkinkan Anda mengedit peran terkait layanan. AWSServiceRoleForAmazonPrometheusScraper Setelah membuat peran tertaut layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, silakan lihat Menyunting peran tertaut layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Anda tidak perlu menghapus AWSServiceRoleForAmazonPrometheusScraper peran secara manual. Saat Anda menghapus semua instance kolektor terkelola yang terkait dengan peran diAWS

Management Console, APIAWS CLI, atau AWS API, Amazon Managed Service for Prometheus membersihkan sumber daya dan menghapus peran terkait layanan untuk Anda.

Wilayah yang Didukung untuk Layanan Terkelola Amazon untuk peran terkait layanan Prometheus

Layanan Terkelola Amazon untuk Prometheus mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi selengkapnya, lihat Wilayah yang didukung.

Logging Amazon Managed Service untuk panggilan API Prometheus menggunakan AWS CloudTrail

Amazon Managed Service untuk Prometheus terintegrasi AWS CloudTrail dengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan di Amazon Managed AWS Service untuk Prometheus. CloudTrail menangkap semua panggilan API untuk Amazon Managed Service untuk Prometheus sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari Layanan Terkelola Amazon untuk konsol Prometheus dan panggilan kode ke Layanan Terkelola Amazon untuk operasi API Prometheus. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman acara secara berkelanjutan ke bucket Amazon S3, termasuk CloudTrail peristiwa untuk Layanan Terkelola Amazon untuk Prometheus. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke Amazon Managed Service untuk Prometheus, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Layanan Dikelola Amazon untuk informasi Prometheus di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Amazon Managed Service untuk Prometheus, aktivitas tersebut direkam dalam CloudTrail suatu peristiwa bersama dengan peristiwa layanan AWS lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Layanan Terkelola Amazon untuk Prometheus, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak

tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi, dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Amazon Managed Service untuk Prometheus mendukung pencatatan tindakan berikut:

- CreateAlertManagerAlerts
- CreateAlertManagerDefinition
- CreateRuleGroupsNamespace
- CreateWorkspace
- DeleteAlertManagerDefinition
- DeleteAlertManagerSilence
- DeleteWorkspace
- DeleteRuleGroupsNamespace
- DescribeAlertManagerDefinition
- DescribeRulesGroupsNamespace
- DescribeWorkspace
- ListRuleGroupsNamespaces
- ListWorkspaces
- PutAlertManagerDefinition
- PutAlertManagerSilences
- PutRuleGroupsNamespace
- UpdateWorkspaceAlias

Setiap peristiwa atau entri log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk suatu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Memahami Layanan Terkelola Amazon untuk entri file log Prometheus

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh: CreateWorkspace

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateWorkspace tindakan.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE123EXAMPLE123-1234567890616",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {

            },
            "attributes": {
```

```
                "mfaAuthenticated": "false",
                "creationDate": "2020-11-30T23:39:29Z"
            }
        }
    },
    "eventTime": "2020-11-30T23:43:21Z",
    "eventSource": "aps.amazonaws.com",
    "eventName": "CreateWorkspace",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.1",
    "userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
    "requestParameters": {
        "alias": "alias-example",
        "clientToken": "12345678-1234-abcd-1234-12345abcd1"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
trace-id,x-amzn-errormessage,x-amz-apigw-id,date",
        "arn": "arn:aws:aps:us-west-2:123456789012:workspace/ws-abc123456-
abcd-1234-5678-1234567890",
        "status": {
            "statusCode": "CREATING"
        },
        "workspaceId": "ws-12345678-1234-abcd-1234-1234567890"
    },
    "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
    "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```

Contoh: CreateAlertManagerDefinition

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateAlertManagerDefinition tindakan.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
```

```
        "principalId": "EXAMPLE123EXAMPLE123-1234567890616",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {

            },
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-09-23T20:20:14Z"
            }
        }
    },
    "eventTime": "2021-09-23T20:22:43Z",
    "eventSource": "aps.amazonaws.com",
    "eventName": "CreateAlertManagerDefinition",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.1",
    "userAgent": "Boto3/1.17.46 Python/3.6.14 Linux/4.14.238-182.422.amzn2.x86_64 exec-
env/AWS_ECS_FARGATE Botocore/1.20.46",
    "requestParameters": {
        "data":
 "YWxlcnRtYW5hZ2VyX2NvbmZpZzogfAogIGdsb2JhbDoKICAgIHNtdHBfc21hcnRob3N0OiAnbG9jYWxob3N0OjI1JwogI
        "clientToken": "12345678-1234-abcd-1234-12345abcd1",
        "workspaceId": "ws-12345678-1234-abcd-1234-1234567890"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
trace-id,x-amzn-errormessage,x-amz-apigw-id,date",
        "status": {
            "statusCode": "CREATING"
        }
    },
    "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
    "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
    "readOnly": false,
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```

Contoh: CreateRuleGroupsNamespace

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateRuleGroupsNamespace tindakan.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE123EXAMPLE123-1234567890616",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {

            },
            "attributes": {
                "creationDate": "2021-09-23T20:22:19Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2021-09-23T20:25:08Z",
    "eventSource": "aps.amazonaws.com",
    "eventName": "CreateRuleGroupsNamespace",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "34.212.33.165",
    "userAgent": "Boto3/1.17.63 Python/3.6.14 Linux/4.14.238-182.422.amzn2.x86_64 exec-
env/AWS_ECS_FARGATE Botocore/1.20.63",
```

```
    "requestParameters": {
        "data":
 "Z3JvdXBzOgogIC0gbmFtZTogdGVzdFJ1bGVHcm91cHNOYW1lc3BhY2UKICAgIHJ1bGVzOgogICAgLSBhbGVydDogdGVzd
        "clientToken": "12345678-1234-abcd-1234-12345abcd1",
        "name": "exampleRuleGroupsNamespace",
        "workspaceId": "ws-12345678-1234-abcd-1234-1234567890"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
trace-id,x-amzn-errormessage,x-amz-apigw-id,date",
        "name": "exampleRuleGroupsNamespace",
        "arn": "arn:aws:aps:us-west-2:492980759322:rulegroupsnamespace/ws-
ae46a85c-1609-4c22-90a3-2148642c3b6c/exampleRuleGroupsNamespace",
        "status": {
            "statusCode": "CREATING"
        },
        "tags": {}
    },
    "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
    "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```


Mengatur peran IAM untuk akun layanan

Dengan peran IAM untuk akun layanan, Anda dapat mengaitkan peran IAM dengan akun layanan Kubernetes. Akun layanan ini kemudian dapat menyediakan izin AWS ke kontainer-kontainer di setiap pod yang menggunakan akun layanan tersebut. Untuk informasi selengkapnya, lihat peran IAM untuk akun layanan.

Peran IAM untuk akun layanan juga dikenal sebagai peran layanan.

Di Amazon Managed Service for Prometheus, menggunakan peran layanan dapat membantu Anda mendapatkan peran yang Anda perlukan untuk mengotorisasi dan mengautentikasi antara Amazon Managed Service untuk Prometheus, server Prometheus, dan server Grafana.

Prasyarat

Prosedur pada halaman ini mengharuskan Anda menginstal antarmuka baris perintah AWS CLI dan EKSCTL.

Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS

Untuk menyiapkan peran layanan guna mengaktifkan Layanan Terkelola Amazon untuk Prometheus untuk mengambil metrik dari server Prometheus di kluster Amazon EKS, Anda harus masuk ke akun dengan izin berikut:

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

Untuk mengatur peran layanan untuk masuk ke Amazon Managed Service untuk Prometheus

1. Buat file dengan nama `createIRSA-AMPIngest.sh` dengan konten berikut ini. Ganti `<my_amazon_eks_clustername>` dengan nama cluster Anda, dan ganti `<my_prometheus_namespace>` dengan namespace Prometheus Anda.

```
#!/bin/bash -e
CLUSTER_NAME=<my_amazon_eks_clustername>
SERVICE_ACCOUNT_NAMESPACE=<my_prometheus_namespace>
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
OIDC_PROVIDER=$(aws eks describe-cluster --name $CLUSTER_NAME --query
 "cluster.identity.oidc.issuer" --output text | sed -e "s/^https:\/\///")
SERVICE_ACCOUNT_AMP_INGEST_NAME=amp-iamproxy-ingest-service-account
SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE=amp-iamproxy-ingest-role
SERVICE_ACCOUNT_IAM_AMP_INGEST_POLICY=AMPIngestPolicy
#
# Set up a trust policy designed for a specific combination of K8s service account
 and namespace to sign in from a Kubernetes cluster which hosts the OIDC Idp.
#
cat <<EOF > TrustPolicy.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-provider/
${OIDC_PROVIDER}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_PROVIDER}:sub": "system:serviceaccount:
${SERVICE_ACCOUNT_NAMESPACE}:${SERVICE_ACCOUNT_AMP_INGEST_NAME}"
        }
      }
    }
  ]
}
EOF
#
# Set up the permission policy that grants ingest (remote write) permissions for
 all AMP workspaces
#
cat <<EOF > PermissionPolicyIngest.json
{
  "Version": "2012-10-17",
   "Statement": [
       {"Effect": "Allow",
        "Action": [
           "aps:RemoteWrite",
           "aps:GetSeries",
           "aps:GetLabels",
           "aps:GetMetricMetadata"
        ],
        "Resource": "*"
      }
   ]
}
EOF

function getRoleArn() {
  OUTPUT=$(aws iam get-role --role-name $1 --query 'Role.Arn' --output text 2>&1)

  # Check for an expected exception
  if [[ $? -eq 0 ]]; then
    echo $OUTPUT
  elif [[ -n $(grep "NoSuchEntity" <<< $OUTPUT) ]]; then
    echo ""
```

```
  else
    >&2 echo $OUTPUT
    return 1
  fi
}

#
# Create the IAM Role for ingest with the above trust policy
#
SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN=$(getRoleArn
 $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE)
if [ "$SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN" = "" ];
then
  #
  # Create the IAM role for service account
  #
  SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN=$(aws iam create-role \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE \
  --assume-role-policy-document file://TrustPolicy.json \
  --query "Role.Arn" --output text)
  #
  # Create an IAM permission policy
  #
  SERVICE_ACCOUNT_IAM_AMP_INGEST_ARN=$(aws iam create-policy --policy-name
 $SERVICE_ACCOUNT_IAM_AMP_INGEST_POLICY \
  --policy-document file://PermissionPolicyIngest.json \
  --query 'Policy.Arn' --output text)
  #
  # Attach the required IAM policies to the IAM role created above
  #
  aws iam attach-role-policy \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE \
  --policy-arn $SERVICE_ACCOUNT_IAM_AMP_INGEST_ARN
else
    echo "$SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN IAM role for ingest already
 exists"
fi
echo $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN
#
# EKS cluster hosts an OIDC provider with a public discovery endpoint.
# Associate this IdP with AWS IAM so that the latter can validate and accept the
 OIDC tokens issued by Kubernetes to service accounts.
# Doing this with eksctl is the easier and best approach.
#
```

```
eksctl utils associate-iam-oidc-provider --cluster $CLUSTER_NAME --approve
```

2. Masukkan perintah berikut untuk memberikan skrip hak istimewa yang diperlukan.

```
chmod +x createIRSA-AMPIngest.sh
```

3. Jalankan penulisan.

Menyiapkan peran IAM untuk akun layanan untuk kueri metrik

Untuk menyiapkan peran IAM untuk akun layanan (peran layanan) guna mengaktifkan kueri metrik dari Amazon Managed Service untuk ruang kerja Prometheus, Anda harus masuk ke akun dengan izin berikut:

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

Untuk menyiapkan peran layanan untuk kueri Amazon Managed Service untuk metrik Prometheus;

1. Buat file dengan nama `createIRSA-AMPQuery.sh` dengan konten berikut ini. Ganti <my_amazon_eks_clustername> dengan nama cluster Anda, dan ganti <my_prometheus_namespace>dengan namespace Prometheus Anda.

```
#!/bin/bash -e
CLUSTER_NAME=<my_amazon_eks_clustername>
SERVICE_ACCOUNT_NAMESPACE=<my_prometheus_namespace>
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
OIDC_PROVIDER=$(aws eks describe-cluster --name $CLUSTER_NAME --query
 "cluster.identity.oidc.issuer" --output text | sed -e "s/^https:\/\///")
SERVICE_ACCOUNT_AMP_QUERY_NAME=amp-iamproxy-query-service-account
SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE=amp-iamproxy-query-role
SERVICE_ACCOUNT_IAM_AMP_QUERY_POLICY=AMPQueryPolicy
#
# Setup a trust policy designed for a specific combination of K8s service account
 and namespace to sign in from a Kubernetes cluster which hosts the OIDC Idp.
#
```

```
cat <<EOF > TrustPolicy.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-provider/
${OIDC_PROVIDER}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_PROVIDER}:sub": "system:serviceaccount:
${SERVICE_ACCOUNT_NAMESPACE}:${SERVICE_ACCOUNT_AMP_QUERY_NAME}"
        }
      }
    }
  ]
}
EOF
#
# Set up the permission policy that grants query permissions for all AMP workspaces
#
cat <<EOF > PermissionPolicyQuery.json
{
  "Version": "2012-10-17",
   "Statement": [
       {"Effect": "Allow",
        "Action": [
           "aps:QueryMetrics",
           "aps:GetSeries",
           "aps:GetLabels",
           "aps:GetMetricMetadata"
        ],
        "Resource": "*"
      }
   ]
}
EOF

function getRoleArn() {
  OUTPUT=$(aws iam get-role --role-name $1 --query 'Role.Arn' --output text 2>&1)
```

```
  # Check for an expected exception
  if [[ $? -eq 0 ]]; then
    echo $OUTPUT
  elif [[ -n $(grep "NoSuchEntity" <<< $OUTPUT) ]]; then
    echo ""
  else
    >&2 echo $OUTPUT
    return 1
  fi
}

#
# Create the IAM Role for query with the above trust policy
#
SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN=$(getRoleArn
 $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE)
if [ "$SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN" = "" ];
then
  #
  # Create the IAM role for service account
  #
  SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN=$(aws iam create-role \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE \
  --assume-role-policy-document file://TrustPolicy.json \
  --query "Role.Arn" --output text)
  #
  # Create an IAM permission policy
  #
  SERVICE_ACCOUNT_IAM_AMP_QUERY_ARN=$(aws iam create-policy --policy-name
 $SERVICE_ACCOUNT_IAM_AMP_QUERY_POLICY \
  --policy-document file://PermissionPolicyQuery.json \
  --query 'Policy.Arn' --output text)
  #
  # Attach the required IAM policies to the IAM role create above
  #
  aws iam attach-role-policy \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE \
  --policy-arn $SERVICE_ACCOUNT_IAM_AMP_QUERY_ARN
else
    echo "$SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN IAM role for query already
 exists"
fi
echo $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN
#
```

```
# EKS cluster hosts an OIDC provider with a public discovery endpoint.
# Associate this IdP with AWS IAM so that the latter can validate and accept the
 OIDC tokens issued by Kubernetes to service accounts.
# Doing this with eksctl is the easier and best approach.
#
eksctl utils associate-iam-oidc-provider --cluster $CLUSTER_NAME --approve
```

2. Masukkan perintah berikut untuk memberikan skrip hak istimewa yang diperlukan.

```
chmod +x createIRSA-AMPQuery.sh
```

3. Jalankan penulisan.

Menggunakan Amazon Managed Service untuk Prometheus dengan titik akhir VPC antarmuka

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC dan Amazon Managed Service untuk Prometheus. Anda dapat menggunakan koneksi ini untuk mengaktifkan Amazon Managed Service untuk Prometheus untuk berkomunikasi dengan sumber daya Anda di VPC Anda tanpa melalui internet publik.

Amazon VPC adalah AWS layanan yang dapat Anda gunakan untuk meluncurkan AWS sumber daya dalam jaringan virtual yang Anda tetapkan. Dengan VPC, Anda memiliki kendali terhadap pengaturan jaringan, seperti rentang alamat IP, subnet, tabel rute, dan pintu masuk jaringan. Untuk menghubungkan VPC Anda ke Amazon Managed Service untuk Prometheus, Anda menentukan titik akhir VPC antarmuka untuk menghubungkan VPC Anda ke layanan. AWS Titik akhir menyediakan konektivitas yang andal dan dapat diskalakan ke Amazon Managed Service untuk Prometheus tanpa memerlukan gateway internet, instance terjemahan alamat jaringan (NAT), atau koneksi VPN. Untuk informasi selengkapnya, silakan lihat Apa itu Amazon VPC dalam Panduan Pengguna Amazon VPC.

Endpoint VPC antarmuka didukung olehAWS PrivateLink, sebuah AWS teknologi yang memungkinkan komunikasi pribadi antara AWS layanan menggunakan antarmuka jaringan elastis dengan alamat IP pribadi. Untuk informasi selengkapnya, silakan lihat kiriman blog Baru – AWS PrivateLink untuk Layanan AWS.

Informasi berikut adalah untuk pengguna Amazon VPC. Untuk informasi tentang cara memulai Amazon VPC, lihat Memulai di Panduan Pengguna Amazon VPC.

Buat titik akhir VPC antarmuka untuk Amazon Managed Service untuk Prometheus

Buat titik akhir VPC antarmuka untuk mulai menggunakan Amazon Managed Service untuk Prometheus. Pilih dari titik akhir nama layanan berikut:

- `com.amazonaws.region.aps-workspaces`

  Pilih nama layanan ini untuk bekerja dengan API yang kompatibel dengan Prometheus. Untuk informasi selengkapnya, lihat API yang kompatibel dengan Prometheus di Amazon Managed Service for Prometheus User Guide.

- `com.amazonaws.region.aps`

  Pilih nama layanan ini untuk melakukan tugas manajemen ruang kerja. Untuk informasi selengkapnya, lihat Layanan Terkelola Amazon untuk API Prometheus di Panduan Pengguna Layanan Terkelola Amazon untuk Prometheus.

> **Note**
>
> Jika Anda menggunakan remote_write dalam VPC tanpa akses internet langsung, Anda juga harus membuat antarmuka VPC endpoint untukAWS Security Token Service, untuk memungkinkan sigv4 bekerja melalui titik akhir. Untuk informasi tentang membuat titik akhir VPCAWS STS, lihat Menggunakan titik akhir AWS STSVPC antarmuka di Panduan Pengguna. AWS Identity and Access Management Anda harus mengatur AWS STS untuk menggunakan endpoint regional.

Untuk informasi selengkapnya, termasuk step-by-step petunjuk untuk membuat titik akhir VPC antarmuka, lihat Membuat titik akhir antarmuka di Panduan Pengguna Amazon VPC.

> **Note**
>
> Anda dapat menggunakan kebijakan titik akhir VPC untuk mengontrol akses ke Layanan Terkelola Amazon untuk titik akhir VPC antarmuka Prometheus. Lihat bagian selanjutnya untuk informasi lebih lanjut.

Jika Anda membuat titik akhir VPC antarmuka untuk Amazon Managed Service untuk Prometheus dan sudah memiliki data yang mengalir ke ruang kerja yang terletak di VPC Anda, metrik akan mengalir melalui titik akhir VPC antarmuka secara default. Layanan Terkelola Amazon untuk Prometheus menggunakan titik akhir publik atau titik akhir antarmuka pribadi (mana pun yang digunakan) untuk melakukan tugas ini.

Mengontrol akses ke Layanan Terkelola Amazon untuk titik akhir VPC Prometheus

Anda dapat menggunakan kebijakan titik akhir VPC untuk mengontrol akses ke Layanan Terkelola Amazon untuk titik akhir VPC antarmuka Prometheus. Kebijakan VPC endpoint adalah kebijakan sumber daya IAM yang Anda lampirkan ke titik akhir ketika membuat atau mengubah titik akhir. Jika Anda tidak melampirkan kebijakan ketika membuat titik akhir, Amazon VPC melampirkan kebijakan default untuk Anda sehingga memungkinkan akses penuh ke layanan. Kebijakan endpoint tidak mengganti atau mengganti kebijakan berbasis identitas IAM atau kebijakan khusus layanan. Ini adalah kebijakan terpisah untuk mengendalikan akses dari titik akhir ke layanan tertentu.

Untuk informasi selengkapnya, silakan lihat Mengendalikan Akses ke Layanan dengan titik akhir VPC dalam Panduan Pengguna Amazon VPC.

Berikut ini adalah contoh kebijakan endpoint untuk Amazon Managed Service untuk Prometheus. Kebijakan ini memungkinkan pengguna dengan peran yang `PromUser` terhubung ke Amazon Managed Service untuk Prometheus melalui VPC untuk melihat ruang kerja dan grup aturan, tetapi tidak, misalnya, untuk membuat atau menghapus ruang kerja.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonManagedPrometheusPermissions",
            "Effect": "Allow",
            "Action": [
                "aps:DescribeWorkspace",
                "aps:DescribeRuleGroupsNamespace",
                "aps:ListRuleGroupsNamespace",
                "aps:ListWorkspaces"
            ],
            "Resource": "arn:aws:aps:*:*:/workspaces*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/PromUser"
                ]
```

```
            }
        }
    ]
}
```

Contoh berikut menunjukkan kebijakan yang hanya mengizinkan permintaan yang berasal dari alamat IP tertentu di VPC yang ditentukan untuk berhasil. Permintaan dari alamat IP lain akan gagal.

```
{
    "Statement": [
        {
            "Action": "aps:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
}
```

Memecahkan masalah

Gunakan bagian berikut untuk membantu memecahkan masalah dengan Amazon Managed Service for Prometheus.

Topik

429 kesalahan

Jika Anda melihat kesalahan 429 yang mirip dengan contoh berikut, permintaan Anda telah melampaui kuota konsumsi Layanan Terkelola Amazon untuk Prometheus.

```
ts=2020-10-29T15:34:41.845Z caller=dedupe.go:112 component=remote level=error
 remote_name=e13b0c
url=http://iamproxy-external.prometheus.uswest2-prod.eks:9090/workspaces/workspace_id/
api/v1/remote_write
msg="non-recoverable error" count=500 err="server returned HTTP status 429
Too Many Requests: ingestion rate limit (6666.666666666667) exceeded while adding 499
 samples and 0 metadata
```

Jika Anda melihat kesalahan 429 yang mirip dengan contoh berikut, permintaan Anda telah melampaui kuota Layanan Terkelola Amazon untuk Prometheus untuk jumlah metrik aktif di ruang kerja.

```
ts=2020-11-05T12:40:33.375Z caller=dedupe.go:112 component=remote level=error
 remote_name=aps
url=http://iamproxy-external.prometheus.uswest2-prod.eks:9090/workspaces/workspace_id/
api/v1/remote_write
msg="non-recoverable error" count=500 err="server returned HTTP status 429 Too Many
 Requests: user=accountid_workspace_id:
```

```
per-user series limit (local limit: 0 global limit: 3000000 actual local limit: 500000)
 exceeded
```

Untuk informasi selengkapnya tentang kuota layanan Amazon Managed Service untuk Prometheus dan tentang cara meminta peningkatan, lihat Layanan Terkelola Amazon untuk kuota layanan Prometheus

Saya melihat sampel duplikat

Jika Anda menggunakan grup Prometheus dengan ketersediaan tinggi, Anda perlu menggunakan label eksternal pada instance Prometheus Anda untuk mengatur deduplikasi. Untuk informasi selengkapnya, lihat Mendeduplikasi metrik ketersediaan tinggi yang dikirim ke Amazon Managed Service untuk Prometheus.

Masalah lain seputar data duplikat dibahas di bagian selanjutnya.

Saya melihat kesalahan tentang stempel waktu sampel

Layanan Terkelola Amazon untuk Prometheus menyerap data secara berurutan, dan mengharapkan setiap sampel memiliki stempel waktu lebih lambat dari sampel sebelumnya.

Jika data Anda tidak tiba secara berurutan, Anda dapat melihat kesalahan tentang`out-of-order samples`,`duplicate sample for timestamp`, atau`samples with different value but same timestamp`. Masalah ini biasanya disebabkan oleh penyiapan klien yang salah yang mengirim data ke Amazon Managed Service untuk Prometheus. Jika Anda menggunakan klien Prometheus yang berjalan dalam mode agen, periksa konfigurasi untuk aturan dengan nama seri duplikat, atau target duplikat. Jika metrik Anda memberikan stempel waktu secara langsung, periksa apakah metrik tersebut tidak rusak.

Untuk detail selengkapnya tentang cara kerjanya, atau cara untuk memeriksa pengaturan Anda, lihat posting blog Memahami Sampel Duplikat dan O ut-of-order Timestamp Errors di Prometheus dari Prom Labs.

Saya melihat pesan kesalahan yang terkait dengan batas

> **Note**
>
> Layanan Terkelola Amazon untuk Prometheus menyediakan metrik penggunaan untuk memantau CloudWatch penggunaan sumber daya Prometheus. Menggunakan fitur alarm

metrik CloudWatch penggunaan, Anda dapat memantau sumber daya dan penggunaan Prometheus untuk mencegah kesalahan batas.

Jika Anda melihat salah satu pesan galat berikut, Anda dapat meminta peningkatan salah satu kuota Layanan Terkelola Amazon untuk Prometheus untuk menyelesaikan masalah. Untuk informasi selengkapnya, lihat Layanan Terkelola Amazon untuk kuota layanan Prometheus.

- batas seri per pengguna terlampaui, silakan hubungi administrator untuk menaikkannya <value>
- batas seri per metrik terlampaui, silakan hubungi administrator untuk menaikkannya <value>
- batas tingkat konsumsi (...) terlampaui
- seri memiliki terlalu banyak label (...) seri: '%s'
- rentang waktu kueri melebihi batas (panjang kueri: xxx, batas: yyy)
- kueri mencapai batas jumlah maksimum potongan saat mengambil potongan dari ingester
- Batas terlampaui. Ruang kerja maksimum per akun.

Output server Prometheus lokal Anda melebihi batas.

Amazon Managed Service untuk Prometheus memiliki kuota layanan untuk jumlah data yang dapat diterima ruang kerja dari server Prometheus. Untuk menemukan jumlah data yang dikirim server Prometheus Anda ke Amazon Managed Service for Prometheus, Anda dapat menjalankan kueri berikut di server Prometheus Anda. Jika Anda menemukan bahwa output Prometheus Anda melebihi batas Layanan Terkelola Amazon untuk Prometheus, Anda dapat meminta peningkatan kuota layanan terkait. Untuk informasi selengkapnya, lihat Layanan Terkelola Amazon untuk kuota layanan Prometheus.

Kueri terhadap server Prometheus mandiri lokal Anda untuk menemukan batas output.

Jenis data	Kueri untuk digunakan	
Seri aktif saat ini	`prometheus_tsdb_head_series`	
Tingkat konsumsi saat ini	`rate(prometheus_ts`	

Jenis data	Kueri untuk digunakan	
	`db_head_samples_appended_total[5m])`	
M ost-to-least daftar seri aktif per nama metrik	`sort_desc(count by(__name__)({__name__!=""}))`	
Jumlah label per seri metrik	`group by(mylabelname)({__name__!=""})`	

Beberapa data saya tidak muncul

Data yang dikirim ke Amazon Managed Service untuk Prometheus dapat dibuang karena berbagai alasan. Tabel berikut menunjukkan alasan bahwa data mungkin dibuang daripada dicerna.

Anda dapat melacak jumlah dan alasan bahwa data dibuang menggunakan Amazon. CloudWatch Untuk informasi selengkapnya, lihat CloudWatch metrik.

Alasan	Arti
greater_than_max_sample_age	Membuang baris log yang lebih tua dari waktu saat ini
new-value-for-timestamp	Sampel duplikat dikirim dengan stempel waktu yang berbeda dari yang direkam sebelumnya
per_metric_series_limit	Pengguna telah mencapai seri aktif per batas metrik

Alasan	Arti
per_user_series_limit	Pengguna telah mencapai jumlah total batas seri aktif
rate_limited	Tingkat konsumsi terbatas
sample-out-of-order	Sampel dikirim keluar dari pesanan dan tidak dapat diproses
label_value_too_long	Nilai label lebih panjang dari batas karakter yang diizinkan
max_label_names_per_series	Pengguna telah menekan nama label per metrik
hilang_metric_name	Nama metrik tidak disediakan
metric_name_invalid	Nama metrik yang diberikan tidak valid
label_invalid	Label tidak valid yang diberikan
duplikate_label_names	Nama label duplikat yang disediakan

Penandaan

Tanda adalah label atribut khusus yang Anda atau AWS tetapkan ke sumber daya AWS. Setiap tanda AWS memiliki dua bagian:

- Sebuah kunci tag (misalnya, `CostCenter`, `Environment`, `Project`, atau `Secret`). Kunci tanda peka terhadap huruf besar dan kecil.
- Bidang opsional yang dikenal sebagai nilai tag (misalnya, `111122223333`, `Production`, atau nama tim). Mengabaikan nilai tag sama dengan menggunakan rangkaian kosong. Seperti kunci tanda, nilai tanda peka huruf besar dan kecil.

Bersama-sama ini dikenal sebagai pasangan nilai-kunci. Anda dapat memiliki tanda yang ditetapkan untuk setiap ruang kerja.

Tag membantu Anda mengidentifikasi dan mengatur sumber daya AWS. Banyak layanan AWS yang mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya tersebut terkait. Misalnya, Anda dapat menetapkan tanda yang sama ke Amazon Managed Service untuk ruang kerja Prometheus yang Anda tetapkan ke bucket Amazon S3. Untuk informasi selengkapnya tentang strategi penandaan, lihat Menandai Sumber Daya AWS.

Di Amazon Managed Service untuk Prometheus, ruang kerja dan ruang nama grup aturan dapat diberi tag. Anda dapat menggunakan konsol,AWS CLI, API, atau SDK untuk menambahkan, mengelola, dan menghapus tanda untuk resource ini. Selain mengidentifikasi, mengatur, dan melacak ruang kerja dan ruang nama grup aturan dengan tanda, Anda dapat menggunakan tanda dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan sumber daya Amazon untuk sumber daya Prometheus.

Pembatasan tag

Batasan dasar berikut berlaku untuk tag:

- Setiap sumber daya dapat memiliki maksimum tanda 50.
- Untuk setiap sumber daya, setiap tanda kunci harus unik, dan setiap tanda kunci hanya dapat memiliki satu nilai.
- Panjang tanda kunci maksimum adalah 128 karakter Unicode dalam UTF-8.

- Panjang tanda nilai maksimum adalah 256 karakter Unicode dalam UTF-8.
- Jika skema penandaan Anda digunakan di beberapa layanan dan sumber daya AWS, ingatlah bahwa layanan lain mungkin memiliki pembatasan pada karakter yang diizinkan. Pada umumnya karakter yang diizinkan adalah huruf, angka, spasi yang dapat direpresentasikan dalam UTF-8, dan karakter berikut:.: + = @ _/-(tanda hubung).
- Tanda Kunci dan nilai peka terhadap huruf besar dan kecil. Sebagai praktik terbaik, tentukan strategi untuk memanfaatkan tag dan secara konsisten menerapkan strategi tersebut di semua jenis sumber daya. Misalnya, putuskan apakah akan menggunakan `Costcenter`, `costcenter`, atau `CostCenter` dan menggunakan kesepakatan yang sama untuk semua tag. Hindari penggunaan tag yang serupa dengan perlakuan kasus yang tidak konsisten.
- Jangan gunakan `aws:`, `AWS:`, atau kombinasi huruf besar atau kecil sebagai prefiks, baik untuk kunci ataupun nilai. Ini disimpan hanya untuk penggunaan AWS. Anda tidak dapat mengedit atau menghapus kunci atau nilai tanda dengan prefiks ini. Tanda dengan awalan ini tidak dihitung terhadap tanda Anda tags-per-resource membatasi.

Topik

Menandai ruang kerja

Gunakan prosedur di bagian ini untuk bekerja dengan tanda untuk Amazon Managed Service untuk ruang kerja Prometheus.

Topik

Tambahkan tag ke ruang kerja

Menambahkan tanda ke Amazon Managed Service untuk ruang kerja Prometheus dapat membantu Anda mengidentifikasi dan mengaturAWSsumber daya dan kelola akses ke sana. Pertama,

Anda menambahkan satu atau beberapa tanda (pasangan kunci-value) ke ruang kerja. Setelah Anda memiliki tanda, Anda dapat membuat kebijakan IAM untuk mengelola akses ke ruang kerja berdasarkan tanda ini. Anda dapat menggunakan konsol atau Konsol.AWS CLIuntuk menambahkan tanda ke Amazon Managed Service untuk ruang kerja Prometheus.

> **Important**
>
> Menambahkan tanda ke ruang kerja dapat memengaruhi akses ke ruang kerja tersebut. Sebelum Anda menambahkan tanda ke ruang kerja, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tanda untuk mengontrol akses ke resource.

Untuk informasi selengkapnya tentang menambahkan tanda ke Amazon Managed Service untuk ruang kerja Prometheus saat Anda membuatnya, lihatBuat ruang kerja.

Topik

- Tambahkan tag ke ruang kerja (konsol)
- Tambahkan tag ke ruang kerja (AWS CLI)

Tambahkan tag ke ruang kerja (konsol)

Anda dapat menggunakan konsol untuk menambahkan satu atau beberapa tanda ke Amazon Managed Service untuk ruang kerja Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Di panel navigasi, pilih ikon Menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. Pilih tab Tag (Tanda).
6. Jika tidak ada tanda yang ditambahkan ke Amazon Managed Service untuk ruang kerja Prometheus, pilihBuat tag. Jika tidak, pilihKelola tag.
7. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
8. (Opsional) Untuk menambahkan tag lain, pilih Tambahkan tag lagi.

9. Setelah selesai menambahkan tag, pilih tag.Simpan perubahan.

Tambahkan tag ke ruang kerja (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk menambahkan tag ke Amazon Managed Service untuk ruang kerja Prometheus. Untuk menambahkan tanda ke ruang kerja saat Anda membuatnya, lihatBuat ruang kerja.

Dalam langkah-langkah ini, kami menganggap bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi terkini. Untuk informasi lebih lanjut, lihat Menginstal AWS Command Line Interface.

Di terminal atau baris perintah, jalankan tomboltag-resourceperintah, dengan menentukan Amazon Resource Name (ARN) dari ruang kerja tempat Anda ingin menambahkan tanda dan nilai dari tanda yang ingin Anda tambahkan. Anda dapat menambahkan lebih dari satu tanda ke ruang kerja. Misalnya, untuk menandai Layanan Terkelola Amazon untuk ruang kerja Prometheus bernamaRuang Kerja Sayadengan dua tag, kunci tag bernama*Status*dengan nilai tag*Rahasia*, dan kunci tag bernama*Tim*dengan nilai tag*Tim Saya*:

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
--tags Status=Secret,Team=My-Team
```

Jika berhasil, perintah ini tidak mengembalikan apa pun.

Lihat tag untuk ruang kerja

Tag dapat membantu Anda mengidentifikasi dan mengatur sumber daya AWS Anda serta mengelola akses ke mereka. Untuk informasi selengkapnya tentang strategi penandaan, lihat Menandai Sumber Daya AWS.

Lihat tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus (konsol)

Anda dapat menggunakan konsol untuk melihat tanda yang terkait dengan Amazon Managed Service untuk ruang kerja Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.

2. Di panel navigasi, pilih ikon Menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. Pilih tab Tag (Tanda).

Lihat tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk melihatAWStag untuk ruang kerja. Jika tidak ada tanda yang telah ditambahkan, daftar yang ditampilkan kosong.

Pada terminal atau baris perintah, jalankan perintah list-tags-for-resource. Misalnya, untuk melihat daftar kunci tanda dan nilai tanda dan tanda untuk ruang kerja:

```
aws amp list-tags-for-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
```

Jika berhasil, perintah ini menampilkan informasi yang serupa dengan yang berikut:

```
{
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

Mengedit tag untuk ruang kerja

Anda dapat mengubah nilai untuk tanda yang terkait dengan ruang kerja. Anda juga dapat mengubah nama kunci, yang setara dengan menghapus tag saat ini dan menambahkan tag yang berbeda dengan nama baru dan nilai yang sama dengan kunci lainnya.

> **Important**
>
> Mengedit tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus dapat memengaruhi akses ke ruang kerja tersebut. Sebelum Anda mengedit nama (kunci) atau nilai tanda untuk ruang kerja, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tanda atau tanda untuk mengontrol akses ke sumber daya seperti repositori.

Mengedit tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus (konsol)

Anda dapat menggunakan konsol untuk mengedit tanda yang terkait dengan Amazon Managed Service untuk ruang kerja Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Pada panel navigasi, pilih ikon menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. Pilih tab Tag (Tanda).
6. Jika tidak ada tanda yang ditambahkan ke ruang kerja, pilihBuat tag. Jika tidak, pilihKelola tag.
7. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
8. (Opsional) Untuk menambahkan tag lain, pilih Tambahkan tag lagi.
9. Setelah selesai menambahkan tag, pilih tag.Simpan perubahan.

Mengedit tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk memperbarui tag untuk ruang kerja. Anda dapat mengubah nilai untuk kunci yang ada, atau menambahkan kunci lain.

Di terminal atau baris perintah, jalankan tomboltag-resourceperintah, menentukan nama sumber daya Amazon (ARN) dari layanan terkelola Amazon untuk ruang kerja Prometheus di mana Anda ingin memperbarui tanda dan menentukan tanda dan tanda tanda:

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tags Team=New-Team
```

Menghapus tag dari ruang kerja

Anda dapat menghapus satu atau beberapa tanda yang terkait dengan ruang kerja. Menghapus tag tidak menghapus tag dari AWS sumber daya yang terkait dengan tag tersebut.

> **Important**
>
> Menghapus tanda untuk Amazon Managed Service untuk ruang kerja Prometheus dapat memengaruhi akses ke ruang kerja tersebut. Sebelum Anda menghapus tanda dari ruang kerja, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tanda atau tanda untuk mengontrol akses ke sumber daya seperti repositori.

Menghapus tag dari Amazon Managed Service untuk ruang kerja Prometheus (konsol)

Anda dapat menggunakan konsol untuk menghapus hubungan antara tanda dan ruang kerja.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Pada panel navigasi, pilih ikon menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. Pilih tab Tag (Tanda).
6. Pilih Kelola tanda.
7. Temukan tanda yang ingin Anda hapus, dan pilihHapus.

Menghapus tag dari Layanan Terkelola Amazon untuk ruang kerja Prometheus (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk menghapus tag dari ruang kerja. Menghapus tanda tidak menghapusnya, tetapi hanya menghapus hubungan antara tanda dan ruang kerja.

> **Note**
>
> Jika Anda menghapus Amazon Managed Service untuk ruang kerja Prometheus, yang menghapus semua tanda dari ruang kerja yang dihapus. Anda tidak perlu menghapus tanda sebelum menghapus tanda sebelum menghapus ruang kerja.

Di terminal atau baris perintah, jalankan tomboluntag-resourceperintah, dengan menentukan Amazon Resource Name (ARN) dari ruang kerja tempat Anda ingin menghapus tanda dan kunci tanda dari tanda yang ingin Anda hapus. Misalnya, menghapus tanda pada ruang kerja yang diberi namaRuang Kerja Sayadengan tombol tag*Status*:

```
aws amp untag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tag-keys Status
```

Jika berhasil, perintah ini tidak mengembalikan apa pun. Untuk memverifikasi tag yang terkait dengan ruang kerja, jalankanlist-tags-for-resourceperintah.

Menandai ruang nama grup aturan

Gunakan prosedur di bagian ini untuk bekerja dengan tanda untuk Amazon Managed Service untuk ruang nama grup aturan Prometheus.

Topik

- Menambahkan tag ke namespace Kalender
- Melihat tag untuk namespace grup aturan
- Mengedit tag untuk namespace Kalender
- Menghapus tag dari namespace Kalender

Menambahkan tag ke namespace Kalender

Menambahkan tanda ke Amazon Managed Service untuk ruang nama grup aturan Prometheus dapat membantu Anda mengidentifikasi dan mengaturAWSsumber daya dan kelola akses ke sana. Pertama, Anda menambahkan satu atau beberapa tanda (pasangan kunci-value) ke ruang nama grup. Setelah Anda memiliki tanda, Anda dapat membuat kebijakan IAM untuk mengelola akses ke namespace berdasarkan tanda ini. Anda dapat menggunakan konsol atau Konsol.AWS CLIuntuk menambahkan tag ke namespace grup aturan Amazon Managed Service untuk Prometheus.

> **Important**
>
> Menambahkan tanda ke ruang nama grup aturan dapat memengaruhi akses ke ruang nama grup aturan. Sebelum Anda menambahkan tanda, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tanda untuk mengontrol akses ke resource.

Untuk informasi selengkapnya tentang menambahkan tanda ke ruang nama grup aturan saat Anda membuatnya, lihatMembuat file aturan.

Topik

- Menambahkan tanda ke aturan grup namespace (konsol)
- Tambahkan tag ke namespace grup aturan (AWS CLI)

Menambahkan tanda ke aturan grup namespace (konsol)

Anda dapat menggunakan konsol untuk menambahkan satu atau beberapa tanda ke Amazon Managed Service untuk ruang nama grup aturan Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Pada panel navigasi, pilih ikon menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. PilihManajemen aturantab.
6. Pilih tombol di sebelah nama namespace dan pilihSunting.
7. PilihBuat tag,Tambahkan tag baru.
8. Di Kunci, masukkan sebuah nama untuk tag tersebut. Anda dapat menambahkan nilai opsional untuk tag di Nilai.
9. (Opsional) Untuk menambahkan tanda lain, pilih Tambahkan tanda baru lagi.
10. Setelah selesai menambahkan tag, pilih tag.Simpan perubahan.

Tambahkan tag ke namespace grup aturan (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk menambahkan tag ke namespace grup aturan Amazon Managed Service untuk Prometheus. Untuk menambahkan tanda ke aturan mengelompokkan namespace saat Anda membuatnya, lihatMengunggah file konfigurasi aturan ke Amazon Managed Service untuk Prometheus.

Dalam langkah-langkah ini, kami menganggap bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi terkini. Untuk informasi lebih lanjut, lihat Menginstal AWS Command Line Interface.

Di terminal atau baris perintah, jalankan tomboltag-resourceperintah, menentukan nama sumber daya Amazon (ARN) dari namespace grup aturan tempat Anda ingin menambahkan tanda dan kunci dan nilai tanda yang ingin Anda tambahkan. Anda dapat menambahkan lebih dari satu tanda ke ruang nama grup aturan. Misalnya, untuk menandai Layanan Terkelola Amazon untuk namespace Prometheus bernamaRuang Kerja Sayadengan dua tag, kunci tag bernama*Status*dengan nilai tag*Rahasia*, dan kunci tag bernama*Tim*dengan nilai tag*Tim Saya*:

```
aws amp tag-resource \
    --resource-arn arn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name \
    --tags Status=Secret,Team=My-Team
```

Jika berhasil, perintah ini tidak mengembalikan apa pun.

Melihat tag untuk namespace grup aturan

Tag dapat membantu Anda mengidentifikasi dan mengatur sumber daya AWS Anda serta mengelola akses ke mereka. Untuk informasi selengkapnya tentang strategi penandaan, lihat Menandai Sumber Daya AWS.

Melihat tag untuk namespace grup aturan Amazon Managed Service untuk Prometheus (konsol)

Anda dapat menggunakan konsol untuk melihat tanda yang terkait dengan Amazon Managed Service untuk ruang nama grup aturan Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Pada panel navigasi, pilih ikon menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. PilihManajemen aturantab.
6. pilih nama Namespace.

Lihat tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk melihatAWStag untuk namespace grup aturan. Jika tidak ada tanda yang telah ditambahkan, daftar yang ditampilkan kosong.

Pada terminal atau baris perintah, jalankan perintah list-tags-for-resource. Misalnya, untuk melihat daftar kunci tanda dan nilai tanda untuk aturan grup namespace:

```
aws amp list-tags-for-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name
```

Jika berhasil, perintah ini menampilkan informasi yang serupa dengan yang berikut:

```
{
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

Mengedit tag untuk namespace Kalender

Anda dapat mengubah nilai untuk tanda yang terkait dengan aturan mengelompokkan namespace. Anda juga dapat mengubah nama kunci, yang setara dengan menghapus tag saat ini dan menambahkan tag yang berbeda dengan nama baru dan nilai yang sama dengan kunci lainnya.

> **Important**
>
> Mengedit tag untuk namespace grup aturan dapat memengaruhi akses ke sana. Sebelum Anda mengedit nama (kunci) atau nilai tanda untuk sumber daya, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tanda atau tanda untuk mengontrol akses ke sumber daya.

Mengedit tag untuk namespace grup aturan Amazon Managed Service untuk Prometheus (konsol)

Anda dapat menggunakan konsol untuk mengedit tanda yang terkait dengan Amazon Managed Service untuk ruang nama grup aturan Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Pada panel navigasi, pilih ikon menu.

3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. PilihManajemen aturantab.
6. pilih nama Namespace.
7. PilihKelola tag,Tambahkan tag baru.
8. Untuk mengubah nilai tanda yang ada, masukkan nilai baru untuk tanda yang sudah ada, masukkan nilai tanda yang sudah ada, masukkan nilai baru untuk tanda yang ada,Nilai.
9. o tambahkan tag tambahan, pilihTambahkan tag baru.
10. Setelah Anda selesai menambahkan dan mengedit tanda, memilihSimpan perubahan.

Mengedit tag untuk ruang nama grup aturan Amazon Managed Service untuk Prometheus (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLIuntuk memperbarui tag untuk namespace grup aturan. Anda dapat mengubah nilai untuk kunci yang ada, atau menambahkan kunci lain.

Di terminal atau baris perintah, jalankan tomboltag-resourceperintah, dengan menentukan Amazon Resource Name (ARN) dari resource tempat Anda ingin memperbarui tanda dan menentukan kunci tanda dan nilai tanda:

```
aws amp tag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tags Team=New-Team
```

Menghapus tag dari namespace Kalender

Anda dapat menghapus satu atau beberapa tanda yang terkait dengan aturan grup namespace. Menghapus tag tidak menghapus tag dari AWS sumber daya yang terkait dengan tag tersebut.

> **Important**
>
> Menghapus tanda untuk resource dapat memengaruhi akses ke resource tersebut. Sebelum Anda menghapus tanda dari sumber daya, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan tanda atau tanda untuk mengontrol akses ke sumber daya seperti repositori.

Menghapus tag dari Layanan Terkelola Amazon untuk namespace grup aturan Prometheus (konsol)

Anda dapat menggunakan konsol untuk menghapus hubungan antara tanda dan namespace grup aturan.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus dihttps://console.aws.amazon.com/prometheus/.
2. Di panel navigasi, pilih ikon Menu.
3. PilihSemua ruang kerja.
4. Pilih ID ruang kerja dari ruang kerja yang ingin Anda kelola.
5. PilihManajemen aturantab.
6. pilih nama Namespace.
7. Pilih Kelola tanda.
8. Di samping tag yang ingin Anda hapus, pilih tag.Hapus.
9. Setelah selesai, pilihSimpan perubahan.

Menghapus tag dari namespace grup aturan Amazon Managed Service untuk Prometheus (AWS CLI)

Ikuti langkah-langkah ini untuk menggunakanAWS CLImenghapus tanda dari aturan grup namespace. Menghapus tanda tidak menghapusnya, tetapi menghapus hubungan antara tanda dan menghapus tanda dan menghapus tanda.

> **Note**
>
> Jika Anda menghapus Amazon Managed Service untuk ruang nama grup aturan Prometheus, yang menghapus semua asosiasi tanda dari nnamespace yang dihapus. Anda tidak perlu menghapus tanda sebelum menghapus tanda sebelum menghapus namespace.

Di terminal atau baris perintah, jalankan tomboluntag-resourceperintah, menentukan nama sumber daya Amazon (ARN) dari namespace grup aturan tempat Anda ingin menghapus tanda dan tanda tanda tanda yang ingin Anda hapus. Misalnya, menghapus tanda pada ruang kerja yang diberi namaRuang Kerja Sayadengan tombol tag`Status`:

```
aws amp untag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tag-keys Status
```

Jika berhasil, perintah ini tidak mengembalikan apa pun. Untuk memverifikasi tanda yang terkait dengan sumber daya, jalankan perintah list-tags-for-resource.

Layanan Terkelola Amazon untuk kuota layanan Prometheus

Dua bagian berikut menjelaskan kuota dan batas yang terkait dengan Amazon Managed Service untuk Prometheus.

Kuota layanan

Amazon Managed Service untuk Prometheus memiliki kuota berikut. Layanan Terkelola Amazon untuk Prometheus menjual metrik penggunaan untuk memantau penggunaan sumber daya PrometheusCloudWatch . Menggunakan fitur alarm metrik CloudWatch penggunaan, Anda dapat memantau sumber daya dan penggunaan Prometheus untuk mencegah kesalahan batas.

Seiring pertumbuhan proyek dan ruang kerja Anda, kuota paling umum yang mungkin perlu Anda pantau atau minta peningkatan adalah: Seri aktif per ruang kerja, Tingkat konsumsi per ruang kerja, dan Ukuran ledakan konsumsi per ruang kerja.

Untuk semua kuota yang dapat disesuaikan, Anda dapat meminta peningkatan kuota dengan memilih tautan di kolom Adjustable, atau dengan meminta peningkatan kuota.

Seri Aktif per batas ruang kerja diterapkan secara dinamis. Untuk informasi selengkapnya, lihat Seri aktif default. Tingkat konsumsi per ruang kerja dan ukuran ledakan konsumsi per ruang kerja bersama-sama mengontrol seberapa cepat Anda dapat menyerap data ke dalam ruang kerja Anda. Untuk informasi selengkapnya, lihat Pelambatan konsumsi.

> **Note**
>
> Kecuali dinyatakan lain, kuota ini per ruang kerja.

Nama	Default	Dapat disesuaikan	Deskripsi
Metrik aktif dengan metadata per ruang kerja	Setiap Wilayah yang didukung: 20.000	Tidak	Jumlah metrik aktif unik dengan metadata per ruang kerja.

Nama	Default	Dapat disesuaikan	Deskripsi
Seri aktif per ruang kerja	Setiap Wilayah yang didukung: 10.000.000 per 2 jam	Ya	Jumlah seri aktif unik per ruang kerja. Serangkaian aktif jika sampel telah dilaporkan dalam 2 jam terakhir. Kapasitas dari 2M hingga 10M secara otomatis disesuaikan berdasarkan 30 menit terakhir penggunaan.
Ukuran grup agregasi peringatan dalam file definisi manajer peringatan	Setiap Wilayah yang didukung: 1.000	Ya	Ukuran maksimum grup agregasi peringatan dalam file definisi manajer peringatan. Setiap kombinasi nilai label group_by akan membuat grup agregasi.
Ukuran file definisi manajer peringatan	Setiap Wilayah yang didukung: 1 Megabyte	Tidak	Ukuran maksimum file definisi manajer peringatan.
Ukuran payload peringatan di Alert Manager	Setiap Wilayah yang didukung: 20 Megabyte	Tidak	Ukuran payload peringatan maksimum dari semua peringatan Alert Manager per ruang kerja. Ukuran peringatan tergantung pada label dan anotasi.
Peringatan di Manajer Peringatan	Setiap Wilayah yang didukung: 1.000	Ya	Jumlah maksimum peringatan Manajer Peringatan bersamaan per ruang kerja.

Nama	Default	Dapat disesuaikan	Deskripsi
Cluster pelacak HA	Setiap Wilayah yang didukung: 500	Tidak	Jumlah maksimum cluster yang akan dilacak oleh pelacak HA untuk sampel yang dicerna per ruang kerja.
Ukuran burst konsumsi per ruang kerja	Setiap Wilayah yang didukung: 1.000.000	Ya	Jumlah maksimum sampel yang dapat dicerna per ruang kerja dalam satu ledakan per detik.
Tingkat konsumsi per ruang kerja	Setiap Wilayah yang didukung: 170.000	Ya	Tingkat konsumsi sampel metrik per ruang kerja per detik.
Aturan penghambatan dalam file definisi manajer peringatan	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum aturan penghambatan dalam file definisi manajer peringatan.
Ukuran label	Setiap Wilayah yang didukung: 7 Kilobyte	Tidak	Ukuran gabungan maksimum dari semua label dan nilai label diterima untuk seri.
Label per seri metrik	Setiap Wilayah yang didukung: 70	Ya	Jumlah label per seri metrik.
Panjang metadata	Setiap Wilayah yang didukung: 1 Kilobyte	Tidak	Panjang maksimum yang diterima untuk metadata metrik. Metadata mengacu pada Nama Metrik, HELP dan UNIT.

Nama	Default	Dapat disesuaikan	Deskripsi
Metadata per metrik	Setiap Wilayah yang didukung: 10	Tidak	Jumlah metadata per metrik.
Node di pohon perutean manajer peringatan	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum node di pohon routing manajer peringatan.
Jumlah operasi API dalam transaksi per detik	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum operasi API per detik per wilayah. Ini termasuk API CRUD ruang kerja, API penandaan, API CRUD namespace grup aturan, dan API CRUD definisi manajer peringatan.
Byte kueri untuk kueri instan	Setiap Wilayah yang didukung: 5 Gigabytes	Tidak	Byte maksimum yang dapat dipindai oleh satu kueri instan.
Byte kueri untuk kueri rentang	Setiap Wilayah yang didukung: 5 Gigabytes	Tidak	Byte maksimum yang dapat dipindai per interval 24 jam dalam satu kueri rentang.
Potongan kueri diambil	Setiap Wilayah yang didukung: 20.000.000	Tidak	Jumlah maksimum potongan yang dapat dipindai selama satu kueri.
Sampel kueri	Setiap Wilayah yang didukung: 50.000.000	Tidak	Jumlah maksimum sampel yang dapat dipindai selama satu kueri.

Nama	Default	Dapat disesuaikan	Deskripsi
Seri kueri diambil	Setiap Wilayah yang didukung: 12.000.000	Tidak	Jumlah maksimum seri yang dapat dipindai selama satu kueri.
Rentang waktu kueri dalam beberapa hari	Setiap Wilayah yang didukung: 32	Tidak	Rentang waktu maksimum dari setiap kueri PromQL.
Ukuran permintaan	Setiap Wilayah yang didukung: 1 Megabyte	Tidak	Ukuran permintaan maksimum untuk konsumsi atau kueri.
Waktu retensi untuk data yang tertelan dalam beberapa hari	Setiap Wilayah yang didukung: 150	Ya	Jumlah hari data di ruang kerja dipertahankan. Data yang lebih tua dari ini dihapus. Anda dapat meminta perubahan kuota untuk menambah atau mengurangi nilai ini.
Interval evaluasi aturan	Setiap Wilayah yang didukung: 30 Detik	Ya	Interval evaluasi aturan minimum dari kelompok aturan per ruang kerja.
Ukuran file definisi namespace grup aturan	Setiap Wilayah yang didukung: 1 Megabyte	Tidak	Ukuran maksimum file definisi namespace grup aturan.
Aturan per ruang kerja	Setiap Wilayah yang didukung: 2.000	Ya	Jumlah maksimum aturan per ruang kerja.

Nama	Default	Dapat disesuaikan	Deskripsi
Template dalam file definisi manajer peringatan	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum template dalam file definisi manajer peringatan.
Ruang kerja per wilayah per akun	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum ruang kerja per wilayah.

Seri aktif default

Amazon Managed Service untuk Prometheus memungkinkan Anda menggunakan hingga kuota rangkaian waktu aktif secara default.

Layanan Terkelola Amazon untuk ruang kerja Prometheus secara otomatis beradaptasi dengan volume konsumsi Anda. Saat penggunaan Anda meningkat, Layanan Terkelola Amazon untuk Prometheus akan secara otomatis meningkatkan kapasitas rangkaian waktu Anda untuk menggandakan penggunaan baseline Anda, hingga kuota default. Misalnya, jika deret waktu aktif rata-rata Anda selama 30 menit terakhir adalah 3,5 juta, Anda dapat menggunakan hingga 7 juta deret waktu tanpa pembatasan.

Jika Anda membutuhkan lebih dari dua kali lipat baseline sebelumnya, Amazon Managed Service for Prometheus secara otomatis mengalokasikan lebih banyak kapasitas saat volume konsumsi Anda meningkat, untuk membantu memastikan beban kerja Anda tidak mengalami pembatasan berkelanjutan, hingga kuota Anda. Namun, pelambatan dapat terjadi jika Anda melebihi dua kali lipat baseline sebelumnya yang dihitung selama 30 menit terakhir. Untuk menghindari pembatasan, Amazon Managed Service for Prometheus merekomendasikan peningkatan konsumsi secara bertahap saat meningkat menjadi lebih dari dua kali lipat deret waktu aktif Anda sebelumnya.

> **Note**
>
> Kapasitas minimum untuk deret waktu aktif adalah 2 juta, tidak ada pelambatan ketika Anda memiliki kurang dari 2 juta seri.
> Untuk melampaui kuota default Anda, Anda dapat meminta peningkatan kuota.

Pelambatan konsumsi

Layanan Terkelola Amazon untuk Prometheus membatasi konsumsi untuk setiap ruang kerja, berdasarkan batas Anda saat ini. Ini membantu menjaga kinerja ruang kerja. Jika Anda melebihi batas, Anda akan melihat `DiscardedSamples` dalam CloudWatch metrik (dengan `rate_limited` alasannya). Anda dapat menggunakan Amazon CloudWatch untuk memantau konsumsi Anda, dan untuk membuat alarm untuk memperingatkan Anda ketika Anda hampir mencapai batas pelambatan. Untuk informasi selengkapnya, lihat CloudWatch metrik.

Amazon Managed Service untuk Prometheus menggunakan algoritma token bucket untuk menerapkan pelambatan konsumsi. Dengan algoritme ini, akun Anda memiliki bucket yang memegang sejumlah tertentu token. Jumlah token dalam bucket mewakili batas konsumsi Anda pada detik tertentu.

Setiap sampel data yang dicerna menghapus satu token dari bucket. Jika ukuran bucket Anda (Ukuran burst konsumsi per ruang kerja) adalah 1.000.000, ruang kerja Anda dapat menyerap satu juta sampel data dalam satu detik. Jika melebihi satu juta sampel untuk dicerna, itu akan dibatasi, dan tidak akan menelan catatan lagi. Sampel data tambahan akan dibuang.

Bucket secara otomatis mengisi ulang pada tingkat yang ditetapkan. Jika bucket berada di bawah kapasitas maksimumnya, sejumlah token ditambahkan kembali setiap detik hingga mencapai kapasitas maksimumnya. Jika ember penuh saat token isi ulang tiba, mereka dibuang. Bucket tidak dapat menampung lebih dari jumlah token maksimumnya. Tingkat isi ulang untuk konsumsi sampel ditetapkan oleh tingkat konsumsi per batas ruang kerja. Jika tingkat konsumsi per ruang kerja Anda diatur ke 170.000, maka tingkat isi ulang untuk bucket adalah 170.000 token per detik.

Jika ruang kerja Anda menyerap 1.000.000 sampel data dalam satu detik, bucket Anda segera dikurangi menjadi nol token. Bucket kemudian diisi ulang oleh 170.000 token setiap detik, hingga mencapai kapasitas maksimum 1.000.000 token. Jika tidak ada lagi konsumsi, ember yang sebelumnya kosong akan kembali ke kapasitas maksimumnya dalam 6 detik.

> **Note**
>
> Tertelan terjadi dalam permintaan batch. Jika Anda memiliki 100 token yang tersedia, dan mengirim permintaan dengan 101 sampel, seluruh permintaan ditolak. Amazon Managed Service untuk Prometheus tidak menerima sebagian permintaan. Jika Anda menulis kolektor, Anda dapat mengelola percobaan ulang (dengan batch yang lebih kecil atau setelah beberapa waktu berlalu).

Anda tidak perlu menunggu ember penuh sebelum ruang kerja Anda dapat menelan lebih banyak sampel data. Anda dapat menggunakan token karena mereka ditambahkan ke bucket. Jika Anda segera menggunakan token isi ulang, ember tidak mencapai kapasitas maksimumnya. Misalnya, jika Anda menghabiskan ember, Anda dapat terus menelan 170.000 sampel data per detik. Bucket dapat diisi ulang hingga kapasitas maksimum hanya jika Anda menelan kurang dari 170.000 sampel data per detik.

Batas tambahan pada data yang dicerna

Layanan Terkelola Amazon untuk Prometheus juga memiliki persyaratan tambahan berikut untuk data yang tertelan ke dalam ruang kerja. Ini tidak dapat disesuaikan.

- Sampel metrik yang lebih tua dari 1 jam ditolak untuk dicerna.
- Setiap sampel dan metadata harus memiliki nama metrik.

Referensi API

Bagian ini mencantumkan operasi API dan struktur data yang didukung oleh Amazon Managed Service untuk Prometheus.

Untuk informasi tentang operasi API ini dan kuota untuk seri, label, dan permintaan API, lihat Amazon Managed Service untuk kuota layanan Prometheus di Amazon Managed Service for Prometheus User Guide.

Topik

- Layanan Dikelola Amazon untuk Prometheus API
- API yang kompatibel dengan Prometheus

Layanan Dikelola Amazon untuk Prometheus API

Layanan Terkelola Amazon untuk Prometheus menyediakan operasi API yang membuat dan memelihara Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus. Ini termasuk API untuk ruang kerja, pencakar, definisi manajer peringatan, ruang nama grup aturan, dan pencatatan.

Untuk informasi selengkapnya tentang Amazon Managed Service for Prometheus API, lihat Referensi API Amazon Managed Service for Prometheus.

Menggunakan Amazon Managed Service untuk Prometheus dengan SDK AWS

AWS kit pengembangan perangkat lunak (SDK) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan pengembang untuk membangun AWS aplikasi dalam bahasa pilihan mereka. Untuk daftar SDK dan alat menurut bahasa, lihat Alat untuk Dibangun AWS di Pusat AWS Pengembang.

> **Versi SDK**
>
> Kami menyarankan Anda menggunakan versi terbaru AWS SDK, dan SDK lainnya, yang Anda gunakan dalam proyek Anda, dan untuk menjaga SDK tetap up to date. AWS SDK memberi Anda fitur dan fungsionalitas terbaru, dan juga pembaruan keamanan.

API yang kompatibel dengan Prometheus

Layanan Terkelola Amazon untuk Prometheus mendukung API yang kompatibel dengan Prometheus berikut.

Untuk informasi selengkapnya tentang penggunaan API yang kompatibel dengan Prometheus, lihat. Kueri menggunakan API yang kompatibel dengan Prometheus

Topik

CreateAlertManagerAlerts

`CreateAlertManagerAlerts`Operasi membuat peringatan di ruang kerja.

Kata kerja HTTP yang valid:

```
POST
```

URI yang valid:

`/workspaces/workspaceId/alertmanager/api/v2/alerts`

Parameter kueri URL:

`alerts`Sebuah array objek, di mana setiap objek mewakili satu peringatan. Berikut ini adalah contoh objek peringatan:

```
[
  {
    "startsAt": "2021-09-24T17:14:04.995Z",
    "endsAt": "2021-09-24T17:14:04.995Z",
    "annotations": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "labels": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "generatorURL": "string"
  }
]
```

Permintaan sampel

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/alerts
 HTTP/1.1
Content-Length: 203,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0

[
  {
    "labels": {
      "alertname": "test-alert"
    },
    "annotations": {
      "summary": "this is a test alert used for demo purposes"
```

```
    },
    "generatorURL": "https://www.amazon.com/"
  }
]
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

DeleteAlertManagerSilence

`DeleteSilence`Menghapus satu keheningan peringatan.

Kata kerja HTTP yang valid:

`DELETE`

URI yang valid:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/silence/`*`silenceID`*

Parameter kueri URL: tidak ada

Permintaan sampel

```
DELETE /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/silence/
d29d9df3-9125-4441-912c-70b05f86f973 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

GetAlertManagerStatus

`GetAlertManagerStatus`Mengambil informasi tentang status manajer peringatan.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/status`

Parameter kueri URL: tidak ada

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/status
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 941
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

```
{
    "cluster": null,
    "config": {
        "original": "global:\n  resolve_timeout: 5m\n  http_config:\n
 follow_redirects: true\n  smtp_hello: localhost\n  smtp_require_tls: true\nroute:
\n  receiver: sns-0\n  group_by:\n  - label\n  continue: false\nreceivers:\n-
 name: sns-0\n  sns_configs:\n  - send_resolved: false\n    http_config:\n
      follow_redirects: true\n    sigv4: {}\n    topic_arn: arn:aws:sns:us-
west-2:123456789012:test\n    subject: '{{ template \"sns.default.subject\" . }}'\n
    message: '{{ template \"sns.default.message\" . }}'\n    workspace_arn:
 arn:aws:aps:us-west-2:123456789012:workspace/ws-58a6a446-5ec4-415b-9052-a449073bbd0a
\ntemplates: []\n"
    },
    "uptime": null,
    "versionInfo": null
}
```

GetAlertManagerSilence

Itu `GetAlertManagerSilence` mengambil informasi tentang satu keheningan peringatan.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/silence/`*`silenceID`*

Parameter kueri URL: tidak ada

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/silence/
d29d9df3-9125-4441-912c-70b05f86f973 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 310
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "id": "d29d9df3-9125-4441-912c-70b05f86f973",
    "status": {
        "state": "active"
    },
    "updatedAt": "2021-10-22T19:32:11.763Z",
    "comment": "hello-world",
    "createdBy": "test-person",
    "endsAt": "2023-07-24T01:05:36.000Z",
    "matchers": [
        {
            "isEqual": true,
            "isRegex": true,
            "name": "job",
            "value": "hello"
        }
    ],
    "startsAt": "2021-10-22T19:32:11.763Z"
}
```

GetLabels

`GetLabels`Operasi mengambil label yang terkait dengan deret waktu.

Kata kerja HTTP yang valid:

- `GET`, `POST`

URI yang valid:

- `/workspaces/`*`workspaceId`*`/api/v1/labels`
- `/workspaces/`*`workspaceId`*`/api/v1/label/`*`label-name`*`/values`URI ini hanya mendukung permintaan GET.

Parameter kueri URL:

`match[]=<series_selector>`Argumen pemilih seri berulang yang memilih seri untuk membaca nama label. Opsional.

`start=<rfc3339 | unix_timestamp>`Mulai stempel waktu. Opsional.

`end=<rfc3339 | unix_timestamp>`Akhiri stempel waktu. Opsional.

Permintaan sampel untuk **`/workspaces/workspaceId/api/v1/labels`**

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/labels HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon untuk **`/workspaces/workspaceId/api/v1/labels`**

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 1435
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "status": "success",
    "data": [
        "__name__",
        "access_mode",
        "address",
        "alertname",
        "alertstate",
        "apiservice",
        "app",
        "app_kubernetes_io_instance",
        "app_kubernetes_io_managed_by",
        "app_kubernetes_io_name",
        "area",
```

```
        "beta_kubernetes_io_arch",
        "beta_kubernetes_io_instance_type",
        "beta_kubernetes_io_os",
        "boot_id",
        "branch",
        "broadcast",
        "buildDate",
        ...
    ]
}
```

Permintaan sampel untuk **/workspaces/workspaceId/api/v1/label/label-name/values**

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/label/access_mode/values
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon untuk **/workspaces/workspaceId/api/v1/label/label-name/values**

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 74
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "status": "success",
    "data": [
        "ReadWriteOnce"
    ]
}
```

GetMetricMetadata

`GetMetricMetadata`Operasi mengambil metadata tentang metrik yang saat ini sedang dikikis dari target. Itu tidak memberikan informasi target apa pun.

Bagian data dari hasil kueri terdiri dari objek di mana setiap kunci adalah nama metrik dan setiap nilai adalah daftar objek metadata unik, seperti yang diekspos untuk nama metrik itu di semua target.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/workspaceId/api/v1/metadata`

Parameter kueri URL:

`limit=<number>`Jumlah maksimum metrik yang akan dikembalikan.

`metric=<string>`Nama metrik untuk memfilter metadata. Jika Anda membiarkan ini kosong, semua metadata metrik diambil.

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/metadata HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
Transfer-Encoding: chunked

{
    "status": "success",
    "data": {
        "aggregator_openapi_v2_regeneration_count": [
            {
                "type": "counter",
                "help": "[ALPHA] Counter of OpenAPI v2 spec regeneration count broken
 down by causing APIService name and reason.",
```

```
                "unit": ""
            }
        ],
        ...
    }
}
```

GetSeries

`GetSeries`Operasi mengambil daftar deret waktu yang cocok dengan set label tertentu.

Kata kerja HTTP yang valid:

- `GET`, `POST`

URI yang valid:

- `/workspaces/`*`workspaceId`*`/api/v1/series`

Parameter kueri URL:

- `match[]=<series_selector>`Argumen pemilih seri berulang yang memilih seri untuk dikembalikan. Setidaknya satu `match[]` argumen harus diberikan.
- `start=<rfc3339 | unix_timestamp>`Mulai stempel waktu. Opsional
- `end=<rfc3339 | unix_timestamp>`Akhiri stempel waktu. Opsional

Permintaan sampel

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/series --data-urlencode
 'match[]=node_cpu_seconds_total{app="prometheus"}' --data-urlencode 'start=1634936400'
 --data-urlencode 'end=1634939100' HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
```

```
Content-Type: application/json
Server: amazon
content-encoding: gzip

{
    "status": "success",
    "data": [
        {
            "__name__": "node_cpu_seconds_total",
            "app": "prometheus",
            "app_kubernetes_io_managed_by": "Helm",
            "chart": "prometheus-11.12.1",
            "cluster": "cluster-1",
            "component": "node-exporter",
            "cpu": "0",
            "heritage": "Helm",
            "instance": "10.0.100.36:9100",
            "job": "kubernetes-service-endpoints",
            "kubernetes_name": "servicesstackprometheuscf14a6d7-node-exporter",
            "kubernetes_namespace": "default",
            "kubernetes_node": "ip-10-0-100-36.us-west-2.compute.internal",
            "mode": "idle",
            "release": "servicesstackprometheuscf14a6d7"
        },
        {
            "__name__": "node_cpu_seconds_total",
            "app": "prometheus",
            "app_kubernetes_io_managed_by": "Helm",
            "chart": "prometheus-11.12.1",
            "cluster": "cluster-1",
            "component": "node-exporter",
            "cpu": "0",
            "heritage": "Helm",
            "instance": "10.0.100.36:9100",
            "job": "kubernetes-service-endpoints",
            "kubernetes_name": "servicesstackprometheuscf14a6d7-node-exporter",
            "kubernetes_namespace": "default",
            "kubernetes_node": "ip-10-0-100-36.us-west-2.compute.internal",
            "mode": "iowait",
            "release": "servicesstackprometheuscf14a6d7"
        },
        ...
    ]
```

```
}
```

ListAlerts

`ListAlerts`Operasi mengambil peringatan yang saat ini aktif di ruang kerja.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/workspaceId/api/v1/alerts`

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/alerts HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 386
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
  "status": "success",
  "data": {
    "alerts": [
      {
        "labels": {
          "alertname": "test-1.alert",
          "severity": "none"
        },
        "annotations": {
```

```
          "message": "message"
        },
        "state": "firing",
        "activeAt": "2020-12-01T19:37:25.429565909Z",
        "value": "1e+00"
      }
    ]
  },
  "errorType": "",
  "error": ""
}
```

ListAlertManagerAlerts

Ini `ListAlertManagerAlerts` mengambil informasi tentang peringatan yang saat ini ditembakkan di manajer peringatan di ruang kerja.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/workspaceId/alertmanager/api/v2/alerts`

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/alerts
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 354
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
```

```
vary: Origin

[
    {
        "annotations": {
            "summary": "this is a test alert used for demo purposes"
        },
        "endsAt": "2021-10-21T22:07:31.501Z",
        "fingerprint": "375eab7b59892505",
        "receivers": [
            {
                "name": "sns-0"
            }
        ],
        "startsAt": "2021-10-21T22:02:31.501Z",
        "status": {
            "inhibitedBy": [],
            "silencedBy": [],
            "state": "active"
        },
        "updatedAt": "2021-10-21T22:02:31.501Z",
        "labels": {
            "alertname": "test-alert"
        }
    }
]
```

ListAlertManagerAlertGroups

`ListAlertManagerAlertGroups`Operasi mengambil daftar grup peringatan yang dikonfigurasi di manajer peringatan di ruang kerja.

Kata kerja HTTP yang valid:

- `GET`

URI yang valid:

- `/workspaces/workspaceId/alertmanager/api/v2/alerts/groups`

Parameter kueri URL:

- `active`Boolean. Jika benar, daftar yang dikembalikan menyertakan peringatan aktif. Bawaannya adalah benar. Opsional

`silenced`Boolean. Jika benar, daftar yang dikembalikan menyertakan peringatan yang dibungkam. Bawaannya adalah benar. Opsional

`inhibited`Boolean. Jika benar, daftar yang dikembalikan menyertakan peringatan yang dihambat. Bawaannya adalah benar. Opsional

`filter`Sebuah array string. Daftar pencocokan untuk memfilter peringatan berdasarkan. Opsional

`receiver`Tali. Penerima pencocokan ekspresi reguler untuk memfilter peringatan berdasarkan. Opsional

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/alerts/
groups HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 443
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "alerts": [
            {
                "annotations": {
                    "summary": "this is a test alert used for demo purposes"
                },
                "endsAt": "2021-10-21T22:07:31.501Z",
                "fingerprint": "375eab7b59892505",
                "receivers": [
```

```
                    {
                        "name": "sns-0"
                    }
                ],
                "startsAt": "2021-10-21T22:02:31.501Z",
                "status": {
                    "inhibitedBy": [],
                    "silencedBy": [],
                    "state": "unprocessed"
                },
                "updatedAt": "2021-10-21T22:02:31.501Z",
                "generatorURL": "https://www.amazon.com/",
                "labels": {
                    "alertname": "test-alert"
                }
            }
        ],
        "labels": {},
        "receiver": {
            "name": "sns-0"
        }
    }
]
```

ListAlertManagerReceivers

`ListAlertManagerReceivers`Operasi mengambil informasi tentang penerima yang dikonfigurasi di manajer peringatan.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/receivers`

Parameter kueri URL: tidak ada

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/receivers
 HTTP/1.1
```

```
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 19
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "name": "sns-0"
    }
]
```

ListAlertManagerSilences

`ListAlertManagerSilences`Operasi mengambil informasi tentang keheningan peringatan yang dikonfigurasi di ruang kerja.

Kata kerja HTTP yang valid:

`GET`

URI yang valid:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/silences`

Permintaan sampel

```
GET /workspaces/ws-58a6a446-5ec4-415b-9052-a449073bbd0a/alertmanager/api/v2/silences
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
```

```
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 312
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "id": "d29d9df3-9125-4441-912c-70b05f86f973",
        "status": {
            "state": "active"
        },
        "updatedAt": "2021-10-22T19:32:11.763Z",
        "comment": "hello-world",
        "createdBy": "test-person",
        "endsAt": "2023-07-24T01:05:36.000Z",
        "matchers": [
            {
                "isEqual": true,
                "isRegex": true,
                "name": "job",
                "value": "hello"
            }
        ],
        "startsAt": "2021-10-22T19:32:11.763Z"
    }
]
```

ListRules

`ListRules`Mengambil informasi tentang aturan yang dikonfigurasi di ruang kerja.

Kata kerja HTTP yang valid:

- `GET`

URI yang valid:

```
/workspaces/workspaceId/api/v1/rules
```

Permintaan sampel

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/rules HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 423
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "status": "success",
    "data": {
        "groups": [
            {
                "name": "test-1.rules",
                "file": "test-rules",
                "rules": [
                    {
                        "name": "record:1",
                        "query": "sum(rate(node_cpu_seconds_total[10m:1m]))",
                        "labels": {},
                        "health": "ok",
                        "lastError": "",
                        "type": "recording",
                        "lastEvaluation": "2021-10-21T21:22:34.429565909Z",
                        "evaluationTime": 0.001005399
                    }
                ],
```

```
                "interval": 60,
                "lastEvaluation": "2021-10-21T21:22:34.429563992Z",
                "evaluationTime": 0.001010504
            }
        ]
    },
    "errorType": "",
    "error": ""
}
```

PutAlertManagerSilences

`PutAlertManagerSilences`Operasi menciptakan keheningan peringatan baru atau memperbarui yang sudah ada.

Kata kerja HTTP yang valid:

`POST`

URI yang valid:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/silences`

Parameter kueri URL:

`silence`Objek yang mewakili keheningan. Berikut ini adalah formatnya:

```
{
  "id": "string",
  "matchers": [
    {
      "name": "string",
      "value": "string",
      "isRegex": Boolean,
      "isEqual": Boolean
    }
  ],
  "startsAt": "timestamp",
  "endsAt": "timestamp",
  "createdBy": "string",
  "comment": "string"
}
```

Permintaan sampel

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/silences
 HTTP/1.1
Content-Length: 281,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0

{
   "matchers":[
      {
         "name":"job",
         "value":"up",
         "isRegex":false,
         "isEqual":true
      }
   ],
   "startsAt":"2020-07-23T01:05:36+00:00",
   "endsAt":"2023-07-24T01:05:36+00:00",
   "createdBy":"test-person",
   "comment":"test silence"
}
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 53
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "silenceID": "512860da-74f3-43c9-8833-cec026542b32"
}
```

QueryMetrics

`QueryMetrics`Operasi mengevaluasi kueri instan pada satu titik waktu atau selama rentang waktu.

Kata kerja HTTP yang valid:

`GET`, `POST`

URI yang valid:

`/workspaces/workspaceId/api/v1/query`URI ini mengevaluasi kueri instan pada satu titik waktu.

`/workspaces/workspaceId/api/v1/query_range`URI ini mengevaluasi kueri instan selama rentang waktu.

Parameter kueri URL:

`query=<string>`Sebuah string kueri ekspresi Prometheus. Digunakan di keduanya `query` dan`query_range`.

`time=<rfc3339 | unix_timestamp>`(Opsional) Timestamp evaluasi jika Anda menggunakan `query` untuk kueri instan pada satu titik waktu.

`timeout=<duration>`(Opsional) Batas waktu evaluasi. Default ke dan dibatasi oleh nilai bendera. `-query.timeout` Digunakan di keduanya `query` dan`query_range`.

`start=<rfc3339 | unix_timestamp>`Mulai stempel waktu jika Anda menggunakan kueri `query_range` untuk rentang waktu tertentu.

`end=<rfc3339 | unix_timestamp>`Akhiri stempel waktu jika Anda menggunakan kueri `query_range` untuk rentang waktu tertentu.

`step=<duration | float>`Lebar langkah resolusi kueri dalam `duration` format atau sebagai `float` beberapa detik. Gunakan hanya jika Anda menggunakan kueri `query_range` untuk rentang waktu tertentu, dan diperlukan untuk kueri tersebut.

Durasi

A `duration` dalam API yang kompatibel dengan Prometheus adalah angka, segera diikuti oleh salah satu unit berikut:

- `ms`milidetik
- `s` detik
- `m`menit

- h jam
- dhari, dengan asumsi sehari selalu memiliki 24 jam
- wminggu, dengan asumsi seminggu selalu memiliki 7d
- ytahun, dengan asumsi satu tahun selalu memiliki 365d

Permintaan sampel

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/query?
query=sum(node_cpu_seconds_total) HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 132
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
content-encoding: gzip

{
    "status": "success",
    "data": {
        "resultType": "vector",
        "result": [
            {
                "metric": {},
                "value": [
                    1634937046.322,
                    "252590622.81000024"
                ]
            }
        ]
    }
}
```

RemoteWrite

`RemoteWrite`Operasi menulis metrik dari server Prometheus ke URL jarak jauh dalam format standar. Biasanya, Anda akan menggunakan klien yang ada seperti server Prometheus untuk memanggil operasi ini.

Kata kerja HTTP yang valid:

`POST`

URI yang valid:

`/workspaces/workspaceId/api/v1/remote_write`

Parameter kueri URL:

Tidak ada

`RemoteWrite`memiliki tingkat konsumsi 70.000 sampel per detik dan ukuran ledakan konsumsi 1.000.000 sampel.

Permintaan sampel

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/remote_write --data-
binary "@real-dataset.sz" HTTP/1.1
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Prometheus/2.20.1
Content-Type: application/x-protobuf
Content-Encoding: snappy
X-Prometheus-Remote-Write-Version: 0.1.0

body
```

> **Note**
>
> Untuk sintaks isi permintaan, lihat definisi buffer protokol di https://github.com/prometheus/prometheus/blob/1c624c58ca934f618be737b4995e22051f5724c1/prompb/remote.pb.go#L64.

Sampel respon

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length:0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

Riwayat Dokumen untuk Layanan Terkelola Amazon untuk Panduan Pengguna Prometheus

Tabel berikut menjelaskan pembaruan dokumentasi penting di Amazon Managed Service for Prometheus User Guide. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Pindahkan AWS API ke panduan referensi API terpisah	Layanan Terkelola Amazon untuk API AWS Prometheus sekarang tersedia dalam referensi mereka sendiri, Layanan Terkelola Amazon untuk Referensi API Prometheus. API yang kompatibel dengan Prometheus terus didokumentasikan di Amazon Managed Service for Prometheus User Guide.	Februari 7, 2024
Menambahkan kunci terkelola pelanggan untuk enkripsi ruang kerja	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk kunci terkelola pelanggan untuk enkripsi ruang kerja. Untuk informasi selengkapnya, lihat Enkripsi data diam.	21 Desember 2023
Menambahkan izin baru ke AmazonPrometheusFullAccess	Menambahkan izin baru ke kebijakan AmazonPrometheusFullAccessterkelola untuk mendukung pembuatan kolektor AWS terkelola untuk kluster Amazon EKS.	26 November 2023

Menambahkan kebijakan terkelola baru, AmazonPrometheusScraperServiceLinkedRolePolicy	Menambahkan kebijakan terkelola baru, AmazonPrometheusScraperServiceLinkedRolePolicybagi kolektor AWS terkelola untuk mengumpulkan metrik dari kluster Amazon EKS.	26 November 2023
Menambahkan kolektor AWS terkelola sebagai metode konsumsi	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk kolektor terkelola. AWS	26 November 2023
Menambahkan dukungan untuk mengintegrasikan dengan Grafana Terkelola Amazon	Layanan Terkelola Amazon untuk Prometheus menambahkan dukungan untuk mengintegrasikan dengan peringatan Grafana Terkelola Amazon.	23 November 2022
Menambahkan izin baru ke AmazonPrometheusConsoleFullAccess	Menambahkan izin baru ke kebijakan AmazonPrometheusConsoleFullAccessterkelola untuk mendukung pengelola peringatan pencatatan dan peristiwa penggaris di CloudWatch Log.	24 Oktober 2022
Menambahkan solusi observabilitas Amazon EKS.	Amazon Managed Service untuk Prometheus menambahkan solusi baru menggunakan Observability Accelerator. AWS Untuk informasi selengkapnya, lihat Menggunakan Akselerator AWS Observabilitas.	14 Oktober 2022

Menambahkan dukungan untuk mengintegrasikan ke pemantauan biaya Amazon EKS.	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk mengintegrasikan ke dalam pemantauan biaya Amazon EKS. Untuk informasi selengkapnya, lihat Menginteg rasikan dengan pemantauan biaya Amazon EKS.	September 22, 2022
Meluncurkan dukungan untuk Alert Manager dan Ruler log di Amazon CloudWatch Logs.	Amazon Managed Service for Prometheus meluncurkan dukungan untuk Alert Manager dan Ruler error log di Amazon Logs. CloudWatch Untuk informasi selengkapnya, lihat CloudWatch Log Amazon.	September 1, 2022
Menambahkan dukungan retensi penyimpanan kustom.	Amazon Managed Service untuk Prometheus menambahkan dukungan penyimpanan kustom, per ruang kerja, dengan memodifik asi kuota untuk ruang kerja tersebut. Untuk informasi selengkapnya tentang kuota di Amazon Managed Service for Prometheus, lihat Kuota layanan.	12 Agustus 2022

Menambahkan metrik penggunaan ke Amazon CloudWatch.	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk mengirim metrik penggunaan ke Amazon. CloudWatch Untuk informasi selengkapnya, lihat CloudWatchmetrik Amazon.	6 Mei 2022
Menambahkan dukungan untuk Wilayah Eropa (London).	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk Wilayah Eropa (London).	4 Mei, 2022
Amazon Managed Service untuk Prometheus umumnya tersedia, dan menambahkan dukungan untuk aturan dan manajer peringatan.	Amazon Managed Service untuk Prometheus umumnya tersedia. Ini juga mendukung aturan dan manajer peringatan. Untuk informasi selengkapnya, lihat Merekam aturan dan aturan peringatan serta Pengelola peringatan dan templat.	29 September 2021
Dukungan penandaan ditambahkan.	Amazon Managed Service untuk Prometheus mendukung penandaan Amazon Managed Service untuk ruang kerja Prometheus.	7 September 2021
Seri aktif dan kuota tingkat konsumsi meningkat.	Kuota seri aktif meningkat menjadi 1.000.000 dan kuota tingkat konsumsi meningkat menjadi 70.000 sampel per detik.	22 Februari 2021

Layanan Terkelola Amazon untuk rilis pratinjau Prometheus.	Pratinjau Amazon Managed Service untuk Prometheus dirilis.	15 Desember 2020

AWSGlosarium

Untuk AWS terminologi terbaru, lihat AWSglosarium di Referensi. Glosarium AWS

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