aws

Panduan Pengguna

Layanan Terkelola Amazon untuk Prometheus

Layanan Terkelola Amazon untuk Prometheus: Panduan Pengguna

Table of Contents

Apa itu Layanan Dikelola 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.

Layanan Terkelola Amazon 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 penyebaran 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 dimasukkan ke dalam ruang kerja disimpan selama 150 hari secara default, dan kemudian dihapus secara otomatis. Anda dapat menyesuaikan periode retensi dengan mengonfigurasi ruang kerja Anda hingga maksimum 1095 hari (tiga tahun). Untuk informasi selengkapnya, lihat Mengonfigurasi ruang kerja Anda.

Wilayah yang Didukung

Layanan Terkelola Amazon untuk Prometheus saat ini mendukung Wilayah berikut:

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Timur (Ohio)	us–east-2	aps.us-east-2.amazonaws.com aps-workspaces.us-east-2.amazonaws.com aps-workspaces-fips.us-east-2.amazonaws.com aps-workspaces-fips.us-east-2.api.aws aps-workspaces.us-east-2.api.aws aps-fips.us-east-2.amazonaws.com aps.us-east-2.api.aws aps-fips.us-east-2.api.aws	HTTPS HTTPS HTTPS HTTPS HTTPS HTTPS HTTPS HTTPS
AS Timur (Virginia Utara)	us-east-1	aps.us-east-1.amazonaws.com aps-workspaces.us-east-1.amazonaws.com aps-workspaces-fips.us-east-1.amazonaws.com aps-workspaces-fips.us-east-1.api.aws aps-workspaces.us-east-1.api.aws aps-fips.us-east-1.amazonaws.com aps.us-east-1.api.aws aps-fips.us-east-1.api.aws	HTTPS HTTPS HTTPS HTTPS HTTPS HTTPS HTTPS HTTPS
AS Barat (Oregon)	us-west-2	aps.us-west-2.amazonaws.com aps-workspaces.us-west-2.amazonaws.com	HTTPS HTTPS HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
		aps-workspaces-fips.us-west-2.amazonaws.com	HTTPS
		aps-workspaces-fips.us-west-2.api.aws	HTTPS
		aps-workspaces.us-west-2.api.aws	HTTPS
		aps-fips.us-west-2.amazonaws.com	HTTPS
		aps.us-west-2.api.aws	HTTPS
		aps-fips.us-west-2.api.aws	
Afrika (Cape Town)	af-south-1	aps.af-south-1.amazonaws.com	HTTPS
		aps-workspaces.af-south-1.amazonaws.com	HTTPS
		aps-workspaces.af-south-1.api.aws	HTTPS
		aps.af-south-1.api.aws	HTTPS
Asia Pasifik (Hong Kong)	ap-east-1	aps.ap-east-1.amazonaws.com	HTTPS
		aps-workspaces.ap-east-1.amazonaws.com	HTTPS
		aps-workspaces.ap-east-1.api.aws	HTTPS
		aps.ap-east-1.api.aws	HTTPS
Asia Pasifik (Malaysia)	ap-southeast-5	aps.ap-southeast-5.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-5.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-5.api.aws	HTTPS
		aps.ap-southeast-5.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Mumbai)	ap-south-1	aps.ap-south-1.amazonaws.com	HTTPS
		aps-workspaces.ap-south-1.amazonaws.com	HTTPS
		aps-workspaces.ap-south-1.api.aws	HTTPS
		aps.ap-south-1.api.aws	HTTPS
Asia Pasifik (Seoul)	ap-northeast-2	aps.ap-northeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-2.api.aws	HTTPS
		aps.ap-northeast-2.api.aws	HTTPS
Asia Pasifik (Singapura)	ap-southeast-1	aps.ap-southeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-1.api.aws	HTTPS
		aps.ap-southeast-1.api.aws	HTTPS
Asia Pasifik (Sydney)	ap-southeast-2	aps.ap-southeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-2.api.aws	HTTPS
		aps.ap-southeast-2.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pasifik (Thailand)	ap-tenggara-7	aps.ap-southeast-7.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-7.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-7.api.aws	HTTPS
		aps.ap-southeast-7.api.aws	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-1.api.aws	HTTPS
		aps.ap-northeast-1.api.aws	HTTPS
Kanada (Pusat)	ca-central-1	aps.ca-central-1.amazonaws.com	HTTPS
		aps-workspaces.ca-central-1.amazonaws.com	HTTPS
		aps-workspaces-fips.ca-central-1.amazonaws.com	HTTPS
		aps-workspaces-fips.ca-central-1.api.aws	HTTPS
		aps-workspaces.ca-central-1.api.aws	HTTPS
		aps-fips.ca-central-1.amazonaws.com	HTTPS
		aps.ca-central-1.api.aws	HTTPS
		aps-fips.ca-central-1.api.aws	

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Frankfurt)	eu-central-1	aps.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.api.aws	HTTPS
		aps.eu-central-1.api.aws	HTTPS
Eropa (Irlandia)	eu-west-1	aps.eu-west-1.amazonaws.com	HTTPS
		aps-workspaces.eu-west-1.amazonaws.com	HTTPS
		aps-workspaces.eu-west-1.api.aws	HTTPS
		aps.eu-west-1.api.aws	HTTPS
Eropa (London)	eu-west-2	aps.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.api.aws	HTTPS
		aps.eu-west-2.api.aws	HTTPS
Eropa (Milan)	eu-south-1	aps.eu-south-1.amazonaws.com	HTTPS
		aps-workspaces.eu-south-1.amazonaws.com	HTTPS
		aps-workspaces.eu-south-1.api.aws	HTTPS
		aps.eu-south-1.api.aws	HTTPS
Eropa (Paris)	eu-west-3	aps.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.api.aws	HTTPS
		aps.eu-west-3.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Stockholm)	eu-north-1	aps.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.api.aws	HTTPS
		aps.eu-north-1.api.aws	HTTPS
Eropa (Zürich)	eu-central-2	aps.eu-central-2.amazonaws.com	HTTPS
		aps-workspaces.eu-central-2.amazonaws.com	HTTPS
		aps-workspaces.eu-central-2.api.aws	HTTPS
		aps.eu-central-2.api.aws	HTTPS
Timur Tengah (UAE)	me-central-1	aps.me-central-1.amazonaws.com	HTTPS
		aps-workspaces.me-central-1.amazonaws.com	HTTPS
		aps-workspaces.me-central-1.api.aws	HTTPS
		aps.me-central-1.api.aws	HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	aps.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.api.aws	HTTPS
		aps.sa-east-1.api.aws	HTTPS

Amazon Managed Service untuk Prometheus menyertakan titik akhir bidang kontrol (untuk melakukan tugas manajemen ruang kerja) dan titik akhir bidang data (untuk bekerja dengan data yang kompatibel dengan Prometheus dalam instance ruang kerja). Titik akhir bidang kontrol dimulai dengan`aps.*`, dan titik akhir jalur data dimulai dengan. `aps-workspaces.*` Titik akhir yang berakhir dengan `.amazonaws.com` dukungan IPv4, dan titik akhir yang berakhir dengan `.api.aws` dukungan keduanya IPv4 dan. IPv6

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 AWS Cost Explorer dan Laporan AWS Biaya dan Penggunaan untuk memantau biaya 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 Layanan Terkelola Amazon untuk Prometheus

Amazon Managed Service for Prometheus adalah layanan tanpa server yang kompatibel dengan Prometheus untuk memantau metrik kontainer yang memudahkan pemantauan lingkungan kontainer dengan aman dalam skala besar. Bagian ini membawa Anda melalui tiga area utama dalam menggunakan Amazon Managed Service untuk Prometheus:

- Buat ruang kerja — Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus untuk menyimpan dan memantau metrik Anda.
- Mengkonsumsi metrik — Ruang kerja Anda kosong sampai Anda mendapatkan metrik ke ruang kerja Anda. Anda dapat mengirim metrik ke Layanan Terkelola Amazon untuk Prometheus, atau meminta metrik mengikis Layanan Terkelola Amazon untuk Prometheus secara otomatis.
- Metrik kueri — Setelah Anda memiliki metrik sebagai data di ruang kerja Anda, Anda siap untuk menanyakan data untuk menjelajahi atau memantau metrik tersebut.

Jika Anda baru mengenal AWS, bagian ini juga mencakup detail tentang pengaturan Akun AWS.

Topik

- Mengatur AWS
- Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus
- Menelan metrik Prometheus ke ruang kerja
- Kueri metrik Prometheus Anda

Mengatur AWS

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

Ketika Anda mendaftar AWS, AWS akun Anda secara otomatis memiliki akses ke semua layanan di AWS, termasuk Amazon Managed Service untuk Prometheus. Nmaun, Anda hanya dikenai biaya untuk layanan yang digunakan.

Topik

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif

Mendaftar untuk Akun AWS

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

Untuk mendaftar untuk Akun AWS

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

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

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

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

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

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

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

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

   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 dengan akses administratif

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

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

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus

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.

Untuk menyiapkan ruang kerja, ikuti langkah-langkah ini.

> **Note**
>
> Untuk informasi lebih rinci tentang membuat ruang kerja dan opsi yang tersedia, lihatBuat Layanan Terkelola Amazon untuk ruang kerja Prometheus.

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 ruang kerja yang unik IDs, 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 tanda tersebut. Anda dapat menambahkan sebuah nilai opsional untuk tanda di Nilai.

   Untuk menambahkan tanda lainnya, silakan pilih Tambahkan tanda baru lagi.
4. Pilih Buat ruang kerja.

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

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

   Buat catatan yang URLs ditampilkan untuk Endpoint - URL tulis jarak jauh dan Titik Akhir - 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 menyerap 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 menghasilkan metrik di Amazon EKS, seperti Kubernetes atau metrik tingkat simpul, Anda dapat menggunakan add-on komunitas Amazon EKS. Untuk informasi selengkapnya, lihat Add-on komunitas yang tersedia di Panduan Pengguna Amazon EKS.

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

> **Note**
>
> Metrik yang dimasukkan ke dalam ruang kerja disimpan selama 150 hari secara default, dan kemudian dihapus secara otomatis. Anda dapat menyesuaikan periode retensi dengan mengonfigurasi ruang kerja Anda hingga maksimum 1095 hari (tiga tahun). Untuk informasi selengkapnya, lihat Mengonfigurasi ruang kerja Anda.

Petunjuk di bagian ini membuat Anda siap dan menjalankan Layanan Terkelola Amazon untuk Prometheus dengan cepat. Ini mengasumsikan bahwa Anda telah membuat ruang kerja. Di bagian ini, 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 cluster Amazon EKS dari mana server Prometheus baru akan mengumpulkan metrik.
- Cluster Amazon EKS Anda harus memiliki driver Amazon EBS CSI yang diinstal (diperlukan oleh Helm).
- 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 memberikan AWS izin ke kontainer di pod mana pun 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 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 dari amp-iamproxy-ingest-roleyang Anda buat. Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS
   - Ganti *WORKSPACE_ID* dengan ID Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.
   - Ganti *REGION* 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 Prometheus, lihat. APIs Kueri metrik Prometheus Anda

Bagian ini mengasumsikan Anda sudah memiliki ruang kerja yang dibuat, dan memasukkan metrik ke dalamnya.

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 kredensional 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, lihatIntegrasikan 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. Lihat informasi yang lebih lengkap di Menghubungkan ke Grafana yang Dikelola Amazon dalam VPC pribadi.

Kelola Layanan Terkelola Amazon untuk ruang kerja Prometheus

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.

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

Topik

Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus

Ikuti langkah-langkah ini untuk membuat Layanan Terkelola Amazon untuk ruang kerja Prometheus. Anda dapat memilih untuk menggunakan AWS CLI atau Amazon Managed Service untuk konsol Prometheus.

> **Note**
>
> Jika Anda menjalankan klaster Amazon EKS, Anda juga dapat membuat ruang kerja baru menggunakan AWS Controller untuk Kubernetes.

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 ruang kerja yang unik IDs, 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 ruang kerja yang unik IDs, 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 tanda di Nilai.

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

6. Pilih Buat ruang kerja.

   Halaman detail ruang kerja muncul. Ini menampilkan informasi termasuk status, ARN, ID ruang kerja, dan titik akhir untuk ruang kerja ini URLs 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 URLs 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

Konfigurasikan ruang kerja Anda

Anda dapat mengonfigurasi ruang kerja Anda untuk hal-hal berikut:

- Tentukan set label dan tentukan batas pada deret waktu aktif yang cocok dengan set label yang Anda tentukan. Kumpulan label adalah satu set dari satu atau lebih label, yang merupakan name/value pasangan yang membantu memberikan konteks pada metrik deret waktu.

  Dengan menentukan set label dan menetapkan batas deret waktu aktif, Anda dapat membatasi lonjakan dalam satu penyewa atau sumber untuk hanya memengaruhi penyewa atau sumber tersebut. Misalnya, jika Anda menetapkan batas deret waktu aktif 1.000.000 pada set label`team=A env=prod`, maka jika jumlah deret waktu yang tertelan yang cocok dengan set label tersebut

melebihi batas, maka hanya deret waktu yang cocok dengan set label yang dibatasi. Dengan cara ini, penyewa lain atau sumber metrik tidak terpengaruh.

Untuk informasi selengkapnya tentang label di Prometheus, lihat Model Data.

- Tetapkan periode retensi untuk menentukan jumlah hari untuk data yang akan disimpan di ruang kerja.

Untuk mengonfigurasi ruang kerja Anda

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.
4. Pilih tab Workspace configurations.
5. Untuk mengatur periode retensi ruang kerja, pilih Edit di bagian Periode retensi. Kemudian tentukan periode retensi baru dalam beberapa hari. Maksimal 1095 hari (tiga tahun).
6. Untuk menambah atau memodifikasi set label dan batas seri aktifnya, pilih Edit di bagian Set label. Kemudian, lakukan hal berikut:

   a. (Opsional) Masukkan nilai di Batas bucket default untuk menetapkan batas jumlah maksimum deret waktu aktif yang dapat dicerna di ruang kerja, hanya menghitung deret waktu yang tidak cocok dengan kumpulan label yang ditentukan.

   b. Untuk menentukan set label, masukkan batas deret waktu aktif untuk label baru yang ditetapkan di bawah batas seri Aktif.

      Kemudian, masukkan label dan nilai untuk satu label yang akan digunakan dalam set label, dan pilih Tambahkan label.

   c. (Opsional) Untuk menentukan set label lain, pilih Tambahkan set label lain dan ulangi langkah sebelumnya.
7. Setelah selesai, pilih Simpan perubahan.

Mengedit alias ruang kerja

Anda dapat mengedit ruang kerja untuk mengubah aliasnya. Untuk mengubah alias ruang kerja menggunakan AWS 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 Layanan Terkelola Amazon Anda untuk detail ruang kerja Prometheus, termasuk ARN

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

Console

Untuk menemukan detail ruang kerja Anda menggunakan Amazon Managed Service for Prometheus console

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. Ini akan menampilkan detail tentang ruang kerja Anda, termasuk:
   - Status saat ini - Status ruang kerja Anda, misalnya Aktif, ditampilkan di bawah Status.
   - ARN - ARN ruang kerja ditampilkan di bawah ARN.
   - ID - ID ruang kerja ditampilkan di bawah ID Ruang Kerja.
   - URLs— Konsol menampilkan beberapa URLs untuk ruang kerja, termasuk URLs untuk menulis ke atau menanyakan data dari ruang kerja.

> **Note**
>
> Secara default, yang URLs diberikan adalah IPv4 URLs. Anda juga dapat menggunakan dualstack (IPv4 dan IPv6 didukung). URLs Ini sama, tetapi berada di domain `api.aws` daripada default`amazonaws.com`. Misalnya, jika Anda melihat yang berikut (IPv4 URL):
>
> ```
> https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-abcd1234-ef56-7890-ab12-example/api/v1/remote_write
> ```
>
> Anda dapat membuat dualstack (termasuk dukungan untuk IPv6), URL sebagai berikut:
>
> ```
> https://aps-workspaces.us-east-1.api.aws/workspaces/ws-abcd1234-ef56-7890-ab12-example/api/v1/remote_write
> ```

Di bawah bagian ini adalah tab dengan informasi tentang aturan, manajer peringatan, log, konfigurasi, dan tag.

AWS CLI

Untuk menemukan detail ruang kerja Anda menggunakan AWS CLI

Perintah berikut mengembalikan rincian ruang kerja. Anda harus mengganti *my-workspace-id* dengan ID ruang kerja ruang kerja yang Anda inginkan detailnya.

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

Ini mengembalikan detail tentang ruang kerja Anda, termasuk:

- Status saat ini — Status ruang kerja Anda, misalnya`ACTIVE`, dikembalikan di `statusCode` properti.
- ARN — ARN ruang kerja dikembalikan di properti. `arn`
- URLs— AWS CLI Mengembalikan URL dasar untuk ruang kerja di `prometheusEndpoint` properti.

> **Note**
>
> Secara default, URL yang dikembalikan adalah IPv4 URL. Anda juga dapat menggunakan URL dualstack (IPv4 dan IPv6 didukung) di domain `api.aws` daripada default. `amazonaws.com` Misalnya, jika Anda melihat yang berikut (IPv4 URL):
>
> ```
> https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-abcd1234-ef56-7890-ab12-example/
> ```
>
> Anda dapat membuat dualstack (termasuk dukungan untuk IPv6), URL sebagai berikut:
>
> ```
> https://aps-workspaces.us-east-1.api.aws/workspaces/ws-abcd1234-ef56-7890-ab12-example/
> ```
>
> Anda juga dapat membuat penulisan dan kueri jarak jauh URLs untuk ruang kerja, dengan menambahkan `/api/v1/remote_write` atau`/api/v1/query`, masing-masing.

Hapus Layanan Terkelola Amazon untuk ruang kerja Prometheus

Menghapus ruang kerja akan menghapus data yang telah dicerna ke dalamnya.

> **Note**
>
> Menghapus Layanan Terkelola Amazon untuk ruang kerja Prometheus tidak secara otomatis menghapus kolektor terkelola yang mengikis AWS metrik dan mengirimkannya ke ruang kerja. Untuk informasi selengkapnya, lihat Temukan dan hapus pencakar.

Untuk menghapus ruang kerja menggunakan AWS CLI

Gunakan perintah berikut ini.

```
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.

Menyerap metrik ke Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus

Metrik harus dimasukkan ke dalam Layanan Terkelola Amazon untuk ruang kerja Prometheus sebelum Anda dapat menanyakan atau memberi tahu metrik tersebut. Bagian ini menjelaskan cara mengatur konsumsi metrik ke dalam ruang kerja Anda.

> **Note**
>
> Metrik yang dimasukkan ke dalam ruang kerja disimpan selama 150 hari secara default, dan kemudian dihapus secara otomatis. Anda dapat menyesuaikan periode retensi dengan mengonfigurasi ruang kerja Anda hingga maksimum 1095 hari (tiga tahun). Untuk informasi selengkapnya, lihat Mengonfigurasi 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

- Kolektor yang dikelola pelanggan

Menelan metrik dengan 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**
>
> Amazon EKS mengekspos metrik, metrik, dan `kube-controller-manager` `kube-scheduler` metrik server API dalam sebuah cluster. Banyak teknologi dan aplikasi lain 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 dan Fargate. EC2

Layanan Terkelola Amazon untuk kolektor Prometheus membuat Antarmuka Jaringan Elastis (ENI) per subnet yang ditentukan saat membuat scraper. Kolektor mengikis metrik melalui ini ENIs, 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

- Apa itu metrik yang kompatibel dengan Prometheus?

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

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? Cluster Amazon EKS mengekspos metrik untuk server API. Cluster Amazon EKS yang merupakan versi Kubernetes `1.28` atau lebih tinggi juga mengekspos metrik untuk dan. `kube-scheduler` `kube-controller-manager` Untuk informasi selengkapnya, lihat Mengambil metrik mentah bidang kontrol dalam format Prometheus di Panduan Pengguna Amazon EKS.

> **Note**
>
> Metrik pengikisan dari cluster dapat dikenakan biaya untuk penggunaan jaringan. Salah satu cara untuk mengoptimalkan biaya ini adalah dengan mengonfigurasi `/metrics` titik akhir Anda untuk mengompres metrik yang disediakan (misalnya, dengan gzip), mengurangi data yang harus dipindahkan di seluruh jaringan. Cara melakukannya tergantung pada aplikasi atau perpustakaan yang menyediakan metrik. Beberapa perpustakaan gzip secara default.

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

Topik

- Buat scraper
- Mengonfigurasi klaster Amazon EKS Anda
- Temukan dan hapus pencakar

- Konfigurasi scraper
- Memecahkan masalah konfigurasi scraper
- Keterbatasan scraper

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.

Ada tiga cara untuk membuat scraper:

- Scraper dibuat secara otomatis untuk Anda saat Anda membuat cluster Amazon EKS melalui konsol Amazon EKS dan memilih untuk mengaktifkan metrik Prometheus.
- Anda dapat membuat scraper dari konsol Amazon EKS untuk cluster yang ada. Buka cluster di konsol Amazon EKS, lalu, pada tab Observability, pilih Add scraper.

  Untuk detail selengkapnya tentang pengaturan yang tersedia, lihat Mengaktifkan metrik Prometheus di Panduan Pengguna Amazon EKS.
- Anda dapat membuat scraper menggunakan AWS API atau file. AWS CLI

  Opsi-opsi ini dijelaskan dalam prosedur berikut.

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.
- VPC Amazon tempat klaster Amazon EKS berada harus mengaktifkan DNS.

> **Note**
>
> Cluster akan dikaitkan dengan scraper dengan nama sumber daya Amazon (ARN). Jika Anda menghapus cluster, dan kemudian membuat yang baru dengan nama yang sama, ARN akan digunakan kembali untuk cluster baru. Karena itu, scraper akan mencoba mengumpulkan

metrik untuk cluster baru. Anda menghapus pencakar secara terpisah dari menghapus cluster.

AWS API

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 milik Anda sendiri IDs, dan menyediakan konfigurasi yang akan digunakan untuk scraper Anda.

> **Note**
>
> Grup keamanan dan subnet harus diatur ke grup keamanan dan subnet untuk cluster yang Anda hubungkan.
> 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. Untuk informasi lebih lanjut tentang format`scrapeConfiguration`, lihatKonfigurasi scraper.

```
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>
    }
}
```

AWS CLI

Untuk membuat scraper menggunakan AWS CLI

Gunakan `create-scraper` perintah untuk membuat scraper dengan file. AWS CLI Contoh berikut membuat scraper di `us-west-2` Wilayah. Anda perlu mengganti informasi cluster Akun AWS, ruang kerja, keamanan, dan Amazon EKS dengan milik Anda sendiri IDs, dan menyediakan konfigurasi yang akan digunakan untuk scraper Anda.

> **Note**
>
> Grup keamanan dan subnet harus diatur ke grup keamanan dan subnet untuk cluster yang Anda hubungkan.
> Anda harus menyertakan setidaknya dua subnet, setidaknya dalam dua zona ketersediaan.

`scrape-configuration`Ini adalah file YAMAL konfigurasi Prometheus yang dikodekan base64. Anda dapat mengunduh konfigurasi tujuan umum dengan `get-default-scraper-configuration` perintah. Untuk informasi lebih lanjut tentang format`scrape-configuration`, lihatKonfigurasi scraper.

```
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.
- Perbarui alias, konfigurasi, atau tujuan scraper dengan operasi UpdateScraperAPI.
- 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.

Pengaturan lintas akun

Untuk membuat scraper dalam penyiapan lintas akun saat klaster Amazon EKS tempat Anda ingin mengumpulkan metrik berada di akun yang berbeda dari kolektor Layanan Terkelola Amazon untuk Prometheus, gunakan prosedur di bawah ini.

Misalnya, ketika Anda memiliki dua akun, akun sumber pertama `account_id_source` tempat Amazon EKS berada, dan akun target kedua `account_id_target` tempat Amazon Managed Service untuk ruang kerja Prometheus berada.

Untuk membuat scraper dalam pengaturan lintas akun

1. Di akun sumber, buat peran `arn:aws:iam::account_id_source:role/Source` dan tambahkan kebijakan kepercayaan berikut.

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

```
    "Principal": {
    "Service": [
        "scraper.aps.amazonaws.com"
     ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {
        "ArnEquals": {
            "aws:SourceArn": "scraper_ARN"
        },
        "StringEquals": {
            "AWS:SourceAccount": "account_id"
        }
    }
}
```

2. Di setiap kombinasi sumber (klaster Amazon EKS) dan target (Layanan Terkelola Amazon untuk ruang kerja Prometheus), Anda perlu membuat `arn:aws:iam::`*`account_id`*`_target:role/Target` peran dan menambahkan kebijakan kepercayaan berikut dengan izin untuk. AmazonPrometheusRemoteWriteAccess

```
{
  "Effect": "Allow",
  "Principal": {
     "AWS": "arn:aws:iam::account_id_source:role/Source"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
     "StringEquals": {
        "sts:ExternalId": "scraper_ARN"
      }
  }
}
```

3. Buat scraper dengan `--role-configuration` opsi.

```
aws amp create-scraper \
  --source eksConfiguration="{clusterArn='arn:aws:eks:us-west-2:account-
id_source:cluster/xarw,subnetIds=[subnet-subnet-id]}" \
```

```
  --scrape-configuration configurationBlob=<base64-encoded-blob> \
  --destination ampConfiguration="{workspaceArn='arn:aws:aps:us-west-2:account-
id_target:workspace/ws-workspace-id'}"\
  --role-configuration '{"sourceRoleArn":"arn:aws:iam::account-id_source:role/
Source", "targetRoleArn":"arn:aws:iam::account-id_target:role/Target"}'
```

4. Validasi pembuatan scraper.

```
aws amp list-scrapers
{
    "scrapers": [
        {
            "scraperId": "scraper-id",
            "arn": "arn:aws:aps:us-west-2:account_id_source:scraper/scraper-id",
            "roleArn": "arn:aws:iam::account_id_source:role/aws-service-role/
scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraperInternal_cc319052-41a3-4",
            "status": {
                "statusCode": "ACTIVE"
            },
            "createdAt": "2024-10-29T16:37:58.789000+00:00",
            "lastModifiedAt": "2024-10-29T16:55:17.085000+00:00",
            "tags": {},
            "source": {
                "eksConfiguration": {
                    "clusterArn": "arn:aws:eks:us-west-2:account_id_source:cluster/
xarw",
                    "securityGroupIds": [
                        "sg-security-group-id",
                        "sg-security-group-id"
                    ],
                    "subnetIds": [
                        "subnet-subnet_id"
                    ]
                }
            },
            "destination": {
                "ampConfiguration": {
                    "workspaceArn": "arn:aws:aps:us-
west-2:account_id_target:workspace/ws-workspace-id"
                }
            }
        }
    ]
```

```
}
```

Mengubah antara RoleConfiguration dan peran terkait layanan

Jika Anda ingin beralih kembali ke peran terkait layanan alih-alih menulis `RoleConfiguration` ke Layanan Terkelola Amazon untuk ruang kerja Prometheus, Anda harus memperbarui `UpdateScraper` dan menyediakan ruang kerja di akun yang sama dengan scraper tanpa. `RoleConfiguration` `RoleConfiguration`Akan dihapus dari scraper dan peran terkait layanan akan digunakan.

Ketika Anda mengubah ruang kerja di akun yang sama dengan scraper dan Anda ingin terus menggunakan`RoleConfiguration`, Anda harus memberikan on lagi. `RoleConfiguration` `UpdateScraper`

Membuat scraper untuk ruang kerja diaktifkan dengan kunci yang dikelola pelanggan

Untuk membuat scraper untuk memasukkan metrik ke dalam Layanan Terkelola Amazon untuk ruang kerja Prometheus dengan kunci yang dikelola pelanggan, gunakan dengan sumber dan target yang disetel ke akun yang `--role-configuration` sama.

```
aws amp create-scraper \
  --source eksConfiguration="{clusterArn='arn:aws:eks:us-west-2:account-id:cluster/
xarw,subnetIds=[subnet-subnet_id]}" \
  --scrape-configuration configurationBlob=<base64-encoded-blob> \
  --destination ampConfiguration="{workspaceArn='arn:aws:aps:us-west-2:account-
id:workspace/ws-workspace-id'}"\
  --role-configuration '{"sourceRoleArn":"arn:aws:iam::account_id:role/Source",
 "targetRoleArn":"arn:aws:iam::account_id:role/Target"}'
```

Kesalahan umum saat membuat pencakar

Berikut ini adalah masalah paling umum saat mencoba membuat scraper baru.

- AWS Sumber daya yang dibutuhkan tidak ada. Grup keamanan, subnet, dan klaster Amazon EKS yang ditentukan harus ada.

- Ruang alamat IP tidak mencukupi. Anda harus memiliki setidaknya satu alamat IP yang tersedia di setiap subnet yang Anda lewatkan ke `CreateScraper` API.

Mengonfigurasi klaster Amazon EKS Anda

Cluster Amazon EKS Anda harus dikonfigurasi untuk memungkinkan scraper mengakses metrik. Ada dua opsi untuk konfigurasi ini:

- Gunakan entri akses Amazon EKS untuk secara otomatis menyediakan Layanan Terkelola Amazon untuk akses kolektor Prometheus ke klaster Anda.
- Konfigurasikan cluster Amazon EKS Anda secara manual untuk pengikisan metrik terkelola.

Topik berikut menjelaskan masing-masing secara lebih rinci.

Konfigurasikan Amazon EKS untuk akses scraper dengan entri akses

Menggunakan entri akses untuk Amazon EKS adalah cara termudah untuk memberi Amazon Managed Service for Prometheus akses untuk mengikis metrik dari cluster Anda.

Cluster Amazon EKS yang Anda gores harus dikonfigurasi untuk memungkinkan otentikasi API. Mode otentikasi cluster harus diatur ke salah satu `API` atau`API_AND_CONFIG_MAP`. Ini dapat dilihat di konsol Amazon EKS pada tab konfigurasi Access pada detail cluster. Untuk informasi selengkapnya, lihat Mengizinkan peran IAM atau pengguna mengakses objek Kubernetes di klaster Amazon EKS Anda di Panduan Pengguna Amazon EKS.

Anda dapat membuat scraper saat membuat cluster, atau setelah membuat cluster:

- Saat membuat klaster — Anda dapat mengonfigurasi akses ini saat membuat klaster Amazon EKS melalui konsol Amazon EKS (ikuti petunjuk untuk membuat scraper sebagai bagian dari cluster), dan kebijakan entri akses akan dibuat secara otomatis, memberikan Layanan Terkelola Amazon untuk Prometheus akses ke metrik klaster.
- Menambahkan setelah cluster dibuat — jika kluster Amazon EKS Anda sudah ada, maka setel mode otentikasi ke salah satu `API` atau`API_AND_CONFIG_MAP`, dan pencakar apa pun yang Anda buat melalui Layanan Terkelola Amazon untuk API Prometheus atau CLI atau melalui konsol Amazon EKS akan secara otomatis memiliki kebijakan entri akses yang benar dibuat untuk Anda, dan pencakar akan memiliki akses ke cluster Anda.

Kebijakan entri akses dibuat

Saat Anda membuat scraper dan membiarkan Amazon Managed Service untuk Prometheus membuat kebijakan entri akses untuk Anda, itu akan menghasilkan kebijakan berikut. Untuk informasi selengkapnya tentang entri akses, lihat Mengizinkan peran IAM atau pengguna mengakses Kubernetes di Panduan Pengguna Amazon EKS.

```
{
    "rules": [
        {
            "effect": "allow",
            "apiGroups": [
                ""
            ],
            "resources": [
                "nodes",
                "nodes/proxy",
                "nodes/metrics",
                "services",
                "endpoints",
                "pods",
                "ingresses",
                "configmaps"
            ],
            "verbs": [
                "get",
                "list",
                "watch"
            ]
        },
        {
            "effect": "allow",
            "apiGroups": [
                "extensions",
                "networking.k8s.io"
            ],
            "resources": [
                "ingresses/status",
                "ingresses"
            ],
            "verbs": [
                "get",
                "list",
                "watch"
            ]
```

```
        },
        {
            "effect": "allow",
            "apiGroups": [
                "metrics.eks.amazonaws.com"
            ],
            "resources": [
                "kcm/metrics",
                "ksh/metrics"
            ],
            "verbs": [
                "get"
            ]
        },
        {
            "effect": "allow",
            "nonResourceURLs": [
                "/metrics"
            ],
            "verbs": [
                "get"
            ]
        }
    ]
}
```

Mengkonfigurasi Amazon EKS secara manual untuk akses scraper

Jika Anda lebih suka menggunakan akses kontrol `aws-auth ConfigMap` to ke cluster kubernetes Anda, Anda masih dapat memberikan Amazon Managed Service untuk Prometheus scraper akses ke metrik Anda. Langkah-langkah berikut akan memberi Amazon Managed Service for Prometheus akses untuk mengikis metrik dari klaster Amazon EKS Anda.

> **Note**
>
> Untuk informasi selengkapnya tentang `ConfigMap` dan mengakses entri, lihat Mengizinkan peran IAM atau pengguna mengakses Kubernetes di Panduan Pengguna Amazon EKS.

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 secara manual 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"]
  - apiGroups: ["metrics.eks.amazonaws.com"]
    resources: ["kcm/metrics", "ksh/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.

> **Note**
>
> Pastikan Anda menggunakan versi terbaru AWS CLI atau SDK. Versi terbaru memberi Anda fitur dan fungsionalitas terbaru, serta pembaruan keamanan. Atau, gunakan AWS Cloudshell, yang selalu memberikan pengalaman baris up-to-date perintah, secara otomatis.

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

Atau, dengan AWS CLI, hubungi:

```
aws amp list-scrapers
```

`ListScrapers`mengembalikan semua scraper 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.

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, Anda dapat menggunakan `UpdateScraper` operasi. Jika Anda perlu memperbarui sumber metrik (misalnya, ke cluster Amazon EKS yang berbeda), Anda harus menghapus scraper dan membuatnya kembali dengan sumber baru.

Konfigurasi yang didukung

Untuk informasi 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.

Karena Amazon EKS adalah satu-satunya layanan yang didukung, satu-satunya service discovery config (`<*_sd_config>`) yang didukung adalah. `<kubernetes_sd_config>`

Daftar lengkap bagian konfigurasi diperbolehkan:

- `<global>`
- `<scrape_config>`
- `<static_config>`
- `<relabel_config>`
- `<metric_relabel_configs>`
- `<kubernetes_sd_config>`

Keterbatasan dalam bagian ini tercantum setelah file konfigurasi sampel.

Contoh file konfigurasi

Berikut ini adalah contoh file konfigurasi YAMAL dengan interval scrape 30 detik. Contoh ini mencakup dukungan untuk metrik server API kube, serta metrik kube-controller-manager dan kube-scheduler. Untuk informasi selengkapnya, lihat Mengambil metrik mentah bidang kontrol dalam format Prometheus di Panduan Pengguna Amazon EKS.

```
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:
      type: Bearer
      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:
      type: Bearer
      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
  # Scheduler metrics
  - job_name: 'ksh-metrics'
```

```
    kubernetes_sd_configs:
    - role: endpoints
    metrics_path: /apis/metrics.eks.amazonaws.com/v1/ksh/container/metrics
    scheme: https
    bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    relabel_configs:
    - source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_service_name
      - __meta_kubernetes_endpoint_port_name
      action: keep
      regex: default;kubernetes;https
  # Controller Manager metrics
  - job_name: 'kcm-metrics'
    kubernetes_sd_configs:
    - role: endpoints
    metrics_path: /apis/metrics.eks.amazonaws.com/v1/kcm/container/metrics
    scheme: https
    bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    relabel_configs:
    - source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_service_name
      - __meta_kubernetes_endpoint_port_name
      action: keep
      regex: default;kubernetes;https
```

Berikut ini adalah 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.
- Resolusi DNS — Terkait dengan nama target, satu-satunya nama server yang dikenali dalam konfigurasi ini adalah server api Kubernetes,. `kubernetes.default.svc` Semua nama mesin lainnya harus ditentukan oleh alamat IP.
- Otorisasi - Hilangkan jika tidak ada otorisasi yang diperlukan. Jika diperlukan, otorisasi harus`Bearer`, dan harus menunjuk ke file`/var/run/secrets/kubernetes.io/serviceaccount/token`. Dengan kata lain, jika digunakan, bagian otorisasi harus terlihat seperti berikut:

```
    authorization:
```

```
    type: Bearer
    credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
```

> **Note**
>
> `type: Bearer`adalah default, sehingga dapat dihilangkan.

Memecahkan masalah konfigurasi scraper

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

> **Important**
>
> Verifikasi bahwa akses pribadi untuk kluster Amazon EKS Anda diaktifkan. Untuk informasi selengkapnya, lihat Titik akhir pribadi cluster di Panduan Pengguna Amazon EKS.

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, ada beberapa alasan mengapa kolektor mungkin tidak dapat menemukannya.

  - Anda mungkin perlu menyesuaikan konfigurasi scrape. Penemuan ini `relabel_config` mungkin perlu disesuaikan.
  - Mungkin ada masalah dengan yang `role` digunakan untuk penemuan.
  - VPC Amazon yang digunakan oleh cluster Amazon EKS mungkin tidak mengaktifkan DNS, yang akan mencegah kolektor menemukan titik akhir.
- 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.)

- `up`hadir 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. Anda dapat memeriksa jumlah titik akhir metrik yang dikikis dengan memeriksa jumlah metrik, menggunakan. `up` `count(up)`

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
- 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:

- AWS Distro 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 AWS Distro 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 Anda VPCs 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

AWS Identity 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 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 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 Version 4 adalah proses untuk menambahkan informasi otentikasi ke AWS permintaan yang dikirim oleh HTTP. Untuk keamanan, sebagian besar permintaan 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

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 AWS Distro untuk. OpenTelemetry

Topik berikut menjelaskan tiga cara berbeda untuk mengatur ADOT sebagai kolektor untuk metrik Anda, berdasarkan apakah metrik Anda berasal dari Amazon EKS, Amazon ECS, atau instans Amazon. EC2

Topik

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

Siapkan konsumsi metrik menggunakan AWS Distro untuk klaster Amazon Elastic OpenTelemetry Kubernetes Service

Anda dapat menggunakan kolektor AWS Distor for OpenTelemetry (ADOT) untuk mengikis metrik dari aplikasi yang diinstrumentasi Prometheus, dan mengirim metrik ke Amazon Managed Service untuk Prometheus.

> **Note**
>
> Untuk informasi selengkapnya tentang kolektor ADOT, lihat AWS Distro untuk. OpenTelemetry
> Untuk informasi selengkapnya tentang aplikasi yang diinstrumentasi Prometheus, lihat. Apa itu metrik yang kompatibel dengan Prometheus?

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:

   JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.us-
east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:adot-
col:amp-iamproxy-ingest-service-account",
          "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
        }
      }
    }
  ]
}
```

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

   JSON

```
{
    "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 Anda AWS .

   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, ganti`AMP_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 untuk 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 selengkapnya, lihat Mengonfigurasi AWS SDK untuk 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 untuk 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 terbiasa dengan beban kerja kontainer, Layanan Terkelola Amazon untuk Prometheus, dan Grafana yang Dikelola Amazon.

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.

Langkah 1: 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 `region` nilai-nilai `endpoint` dan Anda. 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]
```

Langkah 2: 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 *`my-region`* dengan `region` nilai Anda.

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

5. Dorong gambar kontainer Anda.

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

Langkah 3: 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**
>
> Layanan Terkelola Amazon untuk Prometheus berjalan sebagai layanan, mengumpulkan metrik dari kontainer. 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"
```

```
}
```

Langkah 4: Berikan izin tugas Anda untuk mengakses Amazon Managed Service untuk Prometheus

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 Anda dan melampirkan `AmazonPrometheusRemoteWriteAccess` kebijakan padanya. 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.

Langkah 5: Visualisasikan metrik Anda di Amazon Managed Grafana

> **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 Anda, 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 EC2 instans Amazon 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 terbiasa dengan Prometheus, Amazon, dan Layanan Terkelola EC2 Amazon 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, kami sarankan Anda memulai 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 AWS layanan. Untuk kasus penggunaan, pilih EC2. Pilih Berikutnya: Izin.
4. Di bilah pencarian, masukkan AmazonPrometheusRemoteWriteAccess. Untuk nama Kebijakan, pilih AmazonPrometheusRemoteWriteAccess, lalu pilih Lampirkan kebijakan. Pilih Selanjutnya: Tag.
5. (Opsional) Buat tag IAM untuk peran IAM Anda. Pilih Berikutnya: Tinjauan.
6. Masukkan nama untuk peran Anda. Pilih Buat kebijakan.

Luncurkan EC2 instans Amazon

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

Jalankan aplikasi demo

Setelah membuat peran IAM Anda, dan meluncurkan EC2 instance dengan peran tersebut, Anda dapat menjalankan aplikasi demo untuk melihatnya berfungsi.

Untuk menjalankan aplikasi demo dan menguji metrik

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 *`my-region`* dengan nilai Wilayah Anda dan *`my-workspace-id`* dengan ID ruang kerja yang dihasilkan Amazon Managed Service untuk Prometheus untuk Anda. Untuk`region`, ganti *`my-region`* 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

Anda dapat menggunakan instance Prometheus, berjalan dalam mode agen (dikenal sebagai agen Prometheus), untuk mengikis metrik dan mengirimkannya ke Amazon Managed Service untuk ruang kerja Prometheus.

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

> **Warning**
>
> Saat Anda membuat agen Prometheus, Anda bertanggung jawab atas konfigurasi dan pemeliharaannya. Hindari mengekspos titik akhir scrape Prometheus 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 pada 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 dari mana server Prometheus baru akan mengumpulkan metrik.
- Cluster Amazon EKS Anda harus memiliki driver Amazon EBS CSI yang diinstal (diperlukan oleh Helm).
- 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 dari amp-iamproxy-ingest-roleyang Anda buat. Menyiapkan peran layanan untuk menelan metrik dari kluster Amazon EKS
   - Ganti *`WORKSPACE_ID`* dengan ID Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus.

- Ganti *REGION* 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 pada 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

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 memberikan AWS izin ke kontainer di pod mana pun 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
```

Siapkan 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. Gunakan ruang kerja lintas Wilayah untuk menambahkan ketersediaan tinggi di Amazon Managed Service untuk Prometheus

Topik

- Mendeduplikasi metrik ketersediaan tinggi yang dikirim ke Amazon Managed Service untuk Prometheus
- Kirim data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus dengan Prometheus
- Siapkan data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus menggunakan bagan Helm Operator Prometheus
- Kirim data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus dengan Distro untuk AWS OpenTelemetry
- Kirim data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus dengan bagan Helm komunitas Prometheus
- Jawaban atas pertanyaan umum tentang konfigurasi ketersediaan tinggi di Amazon Managed Service untuk Prometheus
- Gunakan ruang kerja lintas Wilayah untuk menambahkan ketersediaan tinggi di Amazon Managed Service untuk Prometheus

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 tersimpan terakhir menyertakan label cluster, tetapi bukan replika, sehingga metrik tampaknya berasal dari satu sumber.

> **Note**
>
> Versi Kubernetes tertentu (1,28 dan 1,29) dapat memancarkan metriknya sendiri dengan label. `cluster` Hal ini dapat menyebabkan masalah dengan Amazon Managed Service untuk deduplikasi Prometheus. Lihat FAQ ketersediaan tinggi untuk informasi lebih lanjut.

Topik berikut menunjukkan cara mengirim data dan menyertakan `cluster` dan `__replica__` label, 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.

> **Note**
>
> Versi Kubernetes tertentu (1,28 dan 1,29) dapat memancarkan metriknya sendiri dengan label. `cluster` Hal ini dapat menyebabkan masalah dengan Amazon Managed Service untuk deduplikasi Prometheus. Lihat FAQ ketersediaan tinggi untuk informasi lebih lanjut.

Siapkan data ketersediaan tinggi ke Amazon Managed Service untuk Prometheus menggunakan bagan Helm Operator Prometheus

Untuk menyiapkan konfigurasi ketersediaan tinggi dengan Operator Prometheus di Helm, Anda harus menerapkan label eksternal pada semua instance grup ketersediaan tinggi, sehingga Layanan Terkelola Amazon untuk 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
```

> **Note**
>
> Versi Kubernetes tertentu (1,28 dan 1,29) dapat memancarkan metriknya sendiri dengan label. `cluster` Hal ini dapat menyebabkan masalah dengan Amazon Managed Service untuk deduplikasi Prometheus. Lihat FAQ ketersediaan tinggi untuk informasi lebih lanjut.

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

AWS Distro for OpenTelemetry (ADOT) adalah distribusi proyek yang aman dan siap produksi. OpenTelemetry ADOT memberi Anda sumber APIs, 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 mengkonfigurasi gambar kontainer kolektor ADOT dan menerapkan label eksternal `cluster` dan `__replica__` ke eksportir tulis jarak jauh Prometheus AWS . 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 penulisan 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.

> **Note**
>
> Versi Kubernetes tertentu (1,28 dan 1,29) dapat memancarkan metriknya sendiri dengan label. `cluster` Hal ini dapat menyebabkan masalah dengan Amazon Managed Service untuk deduplikasi Prometheus. Lihat FAQ ketersediaan tinggi untuk informasi lebih lanjut.

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.

> **Note**
>
> Versi Kubernetes tertentu (1,28 dan 1,29) dapat memancarkan metriknya sendiri dengan label. `cluster` Hal ini dapat menyebabkan masalah dengan Amazon Managed Service untuk deduplikasi Prometheus. Lihat FAQ ketersediaan tinggi untuk informasi lebih lanjut.

Jawaban atas pertanyaan umum tentang konfigurasi ketersediaan tinggi di Amazon Managed Service untuk Prometheus

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.

Kubernetes tampaknya memiliki label klaster sendiri, dan tidak men-deduplikasi metrik saya. Bagaimana saya bisa memperbaikinya?

Sebuah metrik baru, `apiserver_storage_size_bytes` diperkenalkan di Kubernetes 1.28, dengan label. `cluster` Hal ini dapat menyebabkan masalah dengan deduplikasi di Amazon Managed Service untuk Prometheus, yang bergantung pada label. `cluster` Di Kubernetes 1.3, label diubah namanya menjadi `storage-cluster_id` (juga diganti namanya di tambalan selanjutnya dari 1,28 dan 1,29). Jika klaster Anda memancarkan metrik ini dengan `cluster` label, Amazon Managed Service untuk Prometheus tidak dapat men-dedupe deret waktu terkait. Kami menyarankan Anda meningkatkan klaster Kubernetes Anda ke versi patch terbaru untuk menghindari masalah ini. Sebagai alternatif, Anda dapat memberi label ulang `cluster` label pada `apiserver_storage_size_bytes` metrik Anda sebelum memasukkannya ke Amazon Managed Service for Prometheus.

> **Note**
>
> Untuk detail selengkapnya tentang perubahan ke Kubernetes, lihat Mengganti nama klaster Label menjadi storage_cluster_id untuk metrik apiserver_storage_size_bytes dalam proyek Kubernetes. GitHub

Gunakan ruang kerja lintas Wilayah untuk menambahkan ketersediaan tinggi di Amazon Managed Service untuk Prometheus

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]
```

Kueri metrik Prometheus Anda

Sekarang metrik sedang dicerna ke ruang kerja, Anda dapat menanyakannya.

Untuk membuat dasbor dengan representasi visual metrik, Anda dapat menggunakan layanan seperti Grafana Terkelola Amazon. Grafana yang Dikelola Amazon (atau instance Grafana mandiri) dapat membuat antarmuka grafis yang menampilkan metrik Anda dalam berbagai gaya presentasi tampilan. Untuk informasi selengkapnya tentang Grafana Terkelola Amazon, lihat Panduan Pengguna Grafana Terkelola Amazon.

Anda juga dapat membuat kueri satu kali, menjelajahi data Anda, atau menulis aplikasi Anda sendiri yang menggunakan metrik Anda dengan menggunakan kueri langsung. Kueri langsung menggunakan Layanan Terkelola Amazon untuk Prometheus API dan bahasa kueri Prometheus standar, PromQL, untuk mendapatkan data dari ruang kerja Prometheus Anda. Untuk informasi selengkapnya tentang PromQL dan sintaksnya, lihat Meminta Prometheus dalam dokumentasi Prometheus.

Topik

Amankan 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 menggunakan AWS PrivateLink, lalu lintas jaringan dari Anda VPCs 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 Management adalah 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 Version 4 adalah proses untuk menambahkan informasi otentikasi ke AWS permintaan yang dikirim oleh HTTP. Untuk keamanan, sebagian besar permintaan 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, lihatIntegrasikan 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

Anda dapat menggunakan instance Grafana untuk menanyakan metrik Anda di Amazon Managed Service for Prometheus. Topik ini akan membawa Anda melalui cara menanyakan metrik dari Amazon Managed Service untuk Prometheus menggunakan instance Grafana mandiri.

Prasyarat

Instans Grafana — Anda harus memiliki instance Grafana yang mampu mengautentikasi dengan Amazon Managed Service untuk Prometheus.

Amazon Managed Service for 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 memeriksa versi Grafana Anda, masukkan perintah berikut, ganti *grafana_install_directory* dengan jalur ke instalasi Grafana Anda:

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

Jika Anda belum memiliki Grafana mandiri, atau memerlukan versi yang lebih baru, Anda dapat menginstal instance baru. Untuk petunjuk cara menyiapkan Grafana mandiri, lihat Menginstal Grafana di dokumentasi Grafana. Untuk informasi tentang memulai Grafana, lihat Memulai Grafana di dokumentasi Grafana.

Akun AWS— Anda harus memiliki izin Akun AWS yang benar untuk mengakses Layanan Terkelola Amazon Anda untuk metrik Prometheus.

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.

Bagian selanjutnya menjelaskan pengaturan otentikasi dari Grafana secara lebih rinci.

Langkah 1: Siapkan AWS SiGv4

Amazon Managed Service for Prometheus bekerja AWS Identity and Access Management dengan (IAM) untuk mengamankan semua panggilan ke Prometheus dengan kredensyal IAM. APIs 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, SIGv4 autentikasi 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 mengesampingkan 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
```

Langkah 2: 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.

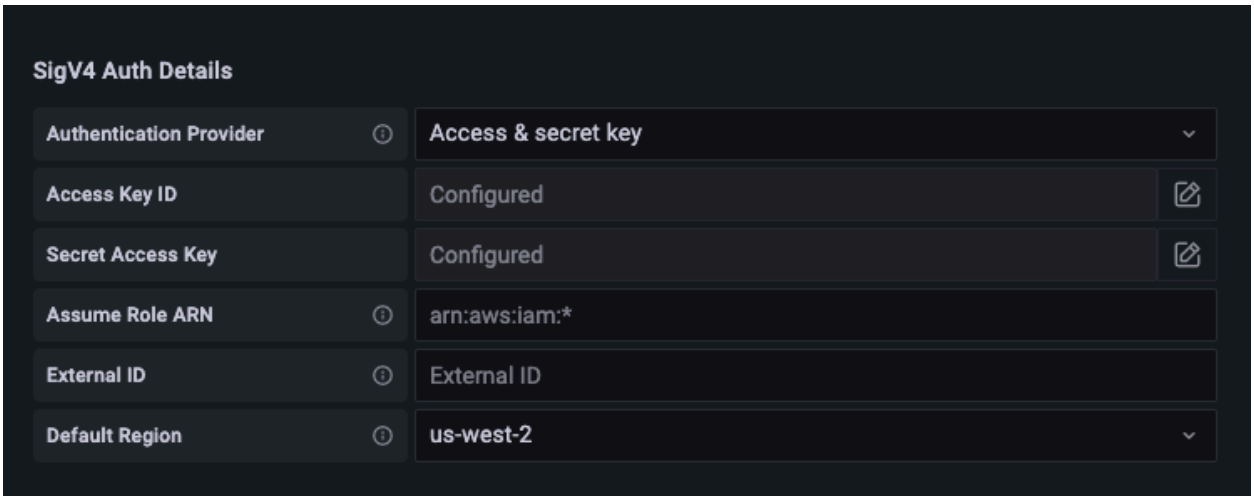
Untuk menambahkan sumber data Prometheus di server Grafana Anda

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 Kredensial.
   - 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.

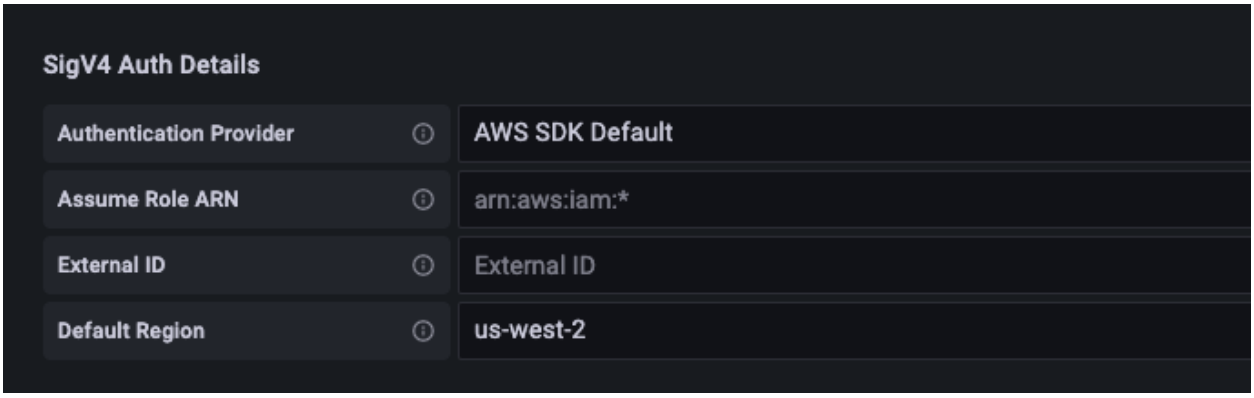
  c. Untuk Wilayah Default, pilih Wilayah Layanan Terkelola Amazon untuk ruang kerja Prometheus. Wilayah ini harus cocok dengan Wilayah yang terdapat dalam URL yang Anda cantumkan di langkah 5.

  d. Pilih Simpan & Uji.

     Anda akan melihat pesan berikut: Sumber data berfungsi

     Jika Anda tidak melihat pesan itu, bagian selanjutnya memberikan tips pemecahan masalah untuk menghubungkan.

     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
```

Langkah 3: (opsional) 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 kredensi default tidak dapat menemukan AWS kredensi yang valid untuk digunakan. Pastikan Anda telah menyiapkan kredensil Anda seperti yang didokumentasikan dalam Menentukan Kredensil. Jika Anda ingin menggunakan konfigurasi bersama, pastikan bahwa `AWS_SDK_LOAD_CONFIG` lingkungan disetel `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 menerapkan 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,Mengatur 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.

Mengatur 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 adalah contohnya.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": [
            "system:serviceaccount:aws-amp:amp-iamproxy-query-service-account",
            "system:serviceaccount:grafana-namespace:grafana-service-account-name"
          ],
          "oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
        }
      }
    }
  ]
}
```

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 Prometheus-kompatibel APIs

Meskipun menggunakan alat seperti Amazon Managed Grafana adalah cara termudah untuk melihat dan menanyakan metrik Anda, Amazon Managed Service for Prometheus juga mendukung

beberapa Prometheus yang kompatibel dengan Prometheus yang dapat Anda gunakan untuk menanyakan metrik Anda. APIs Untuk informasi selengkapnya tentang semua yang kompatibel dengan Prometheus APIs yang tersedia, lihat. Kompatibel dengan Prometheus APIs

Prometheus kompatibel menggunakan APIs bahasa query Prometheus, PromQL, untuk menentukan data yang ingin Anda kembalikan. Untuk detail tentang PromQL dan sintaksnya, lihat Meminta Prometheus di dokumentasi Prometheus.

Saat Anda menggunakan ini APIs untuk menanyakan metrik Anda, permintaan harus ditandatangani dengan proses penandatanganan Versi AWS Tanda Tangan 4. Anda dapat mengatur AWS Signature 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 membuat kueri yang kompatibel dengan Prometheus APIs memandu Anda menggunakan untuk menyiapkan SigV4. `awscurl` AWS

Topik

- Gunakan awscurl untuk melakukan kueri dengan kompatibel dengan Prometheus APIs

Gunakan awscurl untuk melakukan kueri dengan kompatibel dengan Prometheus APIs

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*`Region`*, *`Workspace-id`* dan *`QUERY`* input 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 dalam https://console.aws.amazon.com/iam/.

  Misalnya:

```
$ 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` dan. Anda juga dapat memilih untuk menentukan nama profil yang akan digunakan. Jika tidak ditentukan, `default` file akan digunakan. Misalnya:

```
$ 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 instance yang terkait dengan EC2 instance.

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"
```

Dapatkan statistik tentang penggunaan kueri Anda untuk setiap kueri

Harga kueri didasarkan pada jumlah sampel kueri yang diproses dalam sebulan dari kueri yang dieksekusi. Anda bisa mendapatkan statistik tentang setiap kueri yang Anda buat untuk melacak sampel Anda diproses. Respons kueri untuk `queryRange` API `query` atau dapat menyertakan data statistik tentang sampel kueri yang diproses dengan menyertakan parameter kueri `stats=all` dalam permintaan. Sebuah `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.
`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
                    ]
                ]
            }
        }
```

```
    }
}
```

Menggunakan aturan untuk memodifikasi atau memantau metrik saat diterima

Anda dapat mengatur aturan untuk menindaklanjuti metrik saat diterima oleh Amazon Managed Service for Prometheus. Aturan ini dapat memantau metrik atau bahkan membuat metrik baru yang dihitung berdasarkan metrik yang diterima.

Amazon Managed Service untuk Prometheus mendukung dua jenis aturan yang dievaluasi secara berkala:

- Aturan perekaman memungkinkan Anda untuk menghitung 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 peringatan memungkinkan Anda menentukan kondisi peringatan berdasarkan PromQL dan ambang batas. Ketika aturan memicu ambang batas, pemberitahuan dikirim ke manajer peringatan, yang dapat dikonfigurasi untuk mengelola aturan, atau meneruskannya ke notifikasi 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 Layanan Terkelola Amazon untuk Prometheus memiliki format yang sama dengan file aturan di Prometheus mandiri. Untuk informasi selengkapnya, lihat Mendefinisikan aturan Perekaman dan Aturan Peringatan di dokumentasi Prometheus.

Anda dapat memiliki beberapa file aturan di ruang kerja. Setiap file aturan terpisah terkandung dalam namespace terpisah. 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 dalam kelompok aturan. 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

Memahami izin IAM yang diperlukan untuk menggunakan aturan

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.

> **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.

JSON

```
{
    "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 memberikan akses hanya ke yang `RuleGroupNamespace` ditentukan. Untuk menggunakan kebijakan ini, ganti*<account>*,*<region>*,*<workspace-id>*, dan *<namespace-name>* dengan nilai yang sesuai untuk akun Anda.

JSON

```
{
    "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:*:111122223333:workspace/*",
                "arn:aws:aps:us-east-1:111122223333:rulegroupnamespace/workspace-
id/namespace-name"
            ]
        }
    ]
```

```
}
```

Buat file aturan

Untuk menggunakan aturan di Amazon Managed Service untuk Prometheus, Anda membuat file aturan yang menentukan aturan. Layanan Terkelola Amazon untuk file aturan Prometheus adalah file teks YAMM yang memiliki format yang sama dengan file aturan di Prometheus mandiri. Untuk informasi selengkapnya, lihat Mendefinisikan aturan Perekaman dan Aturan Peringatan di dokumentasi Prometheus.

Berikut ini adalah contoh dasar dari file aturan:

```
groups:
  - name: cpu_metrics
    interval: 60s
    rules:
     - record: avg_cpu_usage
       expr: avg(rate(node_cpu_seconds_total[5m])) by (instance)
     - alert: HighAverageCPU
       expr: avg_cpu_usage > 0.8
       for: 10m
       keep_firing_for: 20m
       labels:
         severity: critical
       annotations:
         summary: "Average CPU usage across cluster is too high"
```

Contoh ini membuat grup aturan `cpu_metrics` yang dievaluasi setiap 60 detik. Grup aturan ini membuat metrik baru menggunakan aturan perekaman, dipanggil `avg_cpu_usage` dan kemudian menggunakannya dalam peringatan. Berikut ini menjelaskan beberapa properti yang digunakan. Untuk informasi selengkapnya tentang aturan peringatan dan properti lain yang dapat Anda sertakan, lihat Aturan peringatan di dokumentasi Prometheus.

- `record: avg_cpu_usage`— Aturan perekaman ini menciptakan metrik baru yang disebut`avg_cpu_usage`.
- Interval evaluasi default grup aturan adalah 60 detik jika `interval` properti tidak ditentukan.
- `expr: avg(rate(node_cpu_seconds_total[5m])) by (instance)`— Ekspresi untuk aturan perekaman ini menghitung tingkat rata-rata penggunaan CPU selama 5 menit terakhir untuk setiap node, dikelompokkan berdasarkan label. `instance`

- `alert: HighAverageCPU`— Aturan peringatan ini membuat peringatan baru yang disebut `HighAverageCPU`
- `expr: avg_cpu_usage > 0.8` — Ekspresi ini memberi tahu peringatan untuk mencari sampel di mana penggunaan CPU rata-rata lebih dari 80%.
- `for: 10m`— Peringatan akan menyala ketika ekspresi terpenuhi selama 10 menit. Dalam hal ini, sampel rata-rata lebih dari 5 menit, sehingga peringatan akan menyala ketika menerima setidaknya 2 sampel yang melebihi ambang batas.
- `keep_firing_for: 20m`— Peringatan ini akan terus menyala sampai sampel berada di bawah ambang batas setidaknya selama 20 menit. Ini dapat berguna untuk menghindari peringatan naik dan turun berulang kali berturut-turut.

Untuk contoh aturan peringatan lainnya, lihat Contoh aturan peringatan.

> **Note**
>
> Anda dapat membuat file definisi aturan secara lokal lalu mengunggahnya ke Amazon Managed Service for Prometheus, atau Anda dapat membuat, mengedit, dan mengunggah definisi secara langsung di dalam konsol Amazon Managed Service for Prometheus. Either way, aturan pemformatan yang sama berlaku. Untuk mempelajari lebih lanjut tentang mengunggah dan mengedit file Anda, lihatUnggah file konfigurasi aturan ke Amazon Managed Service untuk Prometheus.

Unggah file konfigurasi aturan ke Amazon Managed Service untuk Prometheus

Setelah Anda mengetahui aturan apa yang Anda inginkan dalam file konfigurasi aturan, Anda dapat membuat dan mengeditnya di dalam konsol, atau Anda dapat mengunggah file dengan konsol atau AWS CLI.

> **Note**
>
> Jika Anda menjalankan klaster Amazon EKS, Anda juga dapat mengunggah file konfigurasi aturan menggunakan AWS Controllers for Kubernetes.

Untuk menggunakan Amazon Managed Service for Prometheus console untuk mengedit atau mengganti konfigurasi aturan dan membuat namespace

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 Manajemen aturan.
4. Pilih Tambahkan namespace.
5. Pilih Pilih file, dan pilih file definisi aturan.

   Sebagai alternatif, Anda dapat membuat dan mengedit file definisi aturan secara langsung di konsol Amazon Managed Service for Prometheus dengan memilih Tentukan konfigurasi. Ini akan membuat contoh file definisi default yang Anda edit sebelum mengunggah.
6. (Opsional) Untuk menambahkan tag ke namespace, pilih Tambahkan tag baru.

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

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

Untuk menggunakan AWS CLI untuk meng-upload 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 mengunggah file.

   Pada AWS CLI versi 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
```

Pada AWS CLI versi 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, masukkan perintah berikut:

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

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

Mengedit atau mengganti file konfigurasi aturan

Jika ingin mengubah aturan dalam file aturan yang telah diunggah ke Amazon Managed Service untuk Prometheus, Anda dapat mengunggah file aturan baru untuk mengganti konfigurasi yang ada, atau Anda dapat mengedit konfigurasi saat ini secara langsung di konsol. Secara opsional, Anda dapat mengunduh file saat ini, mengeditnya di editor teks, lalu mengunggah versi baru.

Untuk menggunakan Amazon Managed Service untuk konsol Prometheus untuk mengedit konfigurasi aturan

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 Manajemen aturan.
4. Pilih nama file konfigurasi aturan yang ingin Anda edit.
5. (Opsional) Jika Anda ingin mengunduh file konfigurasi aturan saat ini, pilih Unduh atau Salin.
6. Pilih Ubah untuk mengedit konfigurasi langsung di dalam konsol. Pilih Simpan setelah selesai.

   Sebagai alternatif, Anda dapat memilih Ganti konfigurasi untuk mengunggah file konfigurasi baru. Jika demikian, pilih file definisi aturan baru, dan pilih Lanjutkan untuk mengunggahnya.

Untuk menggunakan AWS CLI untuk 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 mengunggah file baru.

   Pada AWS CLI versi 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
   ```

   Pada AWS CLI versi 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, masukkan perintah berikut:

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

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

Memecahkan masalah evaluasi aturan

Panduan ini menyediakan prosedur step-by-step pemecahan masalah untuk masalah umum dengan evaluasi aturan di Amazon Managed Service for Prometheus (AMP). Ikuti prosedur ini untuk mendiagnosis dan menyelesaikan masalah dengan aturan peringatan dan pencatatan Anda.

Topik

Validasi status penembakan peringatan

Saat memecahkan masalah evaluasi aturan, verifikasi terlebih dahulu apakah peringatan Anda telah diaktifkan dengan menanyakan deret waktu sintetis. `ALERTS` Deret `ALERTS` waktu mencakup label berikut:

- alertname — Nama peringatan.
- alertstate - Baik tertunda atau menembak.
  - pending - Peringatan sedang menunggu durasi yang ditentukan dalam `for` klausa.
  - menembak — Peringatan telah memenuhi persyaratan untuk durasi yang ditentukan. Label tambahan ditentukan dalam aturan peringatan Anda.

> **Note**
>
> Saat peringatan menyala atau tertunda, nilai sampelnya adalah 1. Saat peringatan Anda menganggur, tidak ada sampel yang diproduksi.

Selesaikan pemberitahuan peringatan yang hilang

Jika peringatan diaktifkan tetapi pemberitahuan tidak tiba, verifikasi pengaturan Alertmanager berikut:

1. Verifikasi konfigurasi Alertmanager Anda — Periksa apakah penerima rute, dan pengaturan dikonfigurasi dengan benar. Tinjau pengaturan blok rute, termasuk waktu tunggu, interval waktu, dan label yang diperlukan, yang dapat memengaruhi penembakan peringatan. Bandingkan aturan peringatan dengan rute dan penerima yang sesuai untuk mengonfirmasi pencocokan yang tepat. Untuk rute dengan`time_interval`, verifikasi bahwa stempel waktu termasuk dalam interval yang ditentukan.

2. Periksa izin penerima peringatan — Saat menggunakan topik Amazon SNS, verifikasi AMP memiliki izin yang diperlukan untuk memublikasikan notifikasi. Untuk informasi selengkapnya, lihat Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan peringatan ke topik Amazon SNS Anda.
3. Validasi kompatibilitas payload penerima — Konfirmasikan penerima peringatan Anda menerima format payload Alertmanager. Untuk persyaratan Amazon SNS, lihat. Memahami aturan validasi pesan Amazon SNS
4. Tinjau log Alertmanager — AMP menyediakan log vended dari Alertmanager untuk membantu men-debug masalah notifikasi. Untuk informasi selengkapnya, lihat Pantau Layanan Terkelola Amazon untuk acara Prometheus dengan Log CloudWatch .

Untuk informasi selengkapnya tentang Alertmanager, lihat. Mengelola dan meneruskan peringatan di Amazon Managed Service untuk Prometheus dengan manajer peringatan

Periksa status kesehatan aturan

Aturan yang salah dapat menyebabkan kegagalan evaluasi. Gunakan metode berikut untuk mengidentifikasi mengapa aturan gagal dievaluasi:

Example

Gunakan ListRules API

ListRulesAPI menyediakan informasi tentang kesehatan aturan. Periksa `health` dan `lastError` bidang untuk mendiagnosis masalah.

Contoh respon:

```
{
  "status": "success",
  "data": {
    "groups": [
      {
        "name": "my_rule_group",
        "file": "my_namespace",
        "rules": [
          {
            "state": "firing",
            "name": "broken_alerting_rule",
```

```
            "query": "...",
            "duration": 0,
            "keepFiringFor": 0,
            "labels": {},
            "annotations": {},
            "alerts": [],
            "health": "err",
            "lastError": "vector contains metrics with the same labelset after applying
 alert labels",
            "type": "alerting",
            "lastEvaluation": "1970-01-01T00:00:00.00000000Z",
            "evaluationTime": 0.08
          }
        ]
      }
    ]
  }
}
```

Example

Gunakan log vended

ListRules API hanya menampilkan informasi terbaru. Untuk riwayat yang lebih detail, aktifkan log vended di ruang kerja Anda untuk mengakses:

- Cap waktu kegagalan evaluasi
- Pesan kesalahan terperinci
- Data evaluasi historis

Contoh pesan log yang dijual:

```
{
  "workspaceId": "ws-a2c55905-e0b4-4065-a310-d83ce597a391",
  "message": {
    "log": "Evaluating rule failed, name=broken_alerting_rule, group=my_rule_group,
 namespace=my_namespace, err=vector contains metrics with the same labelset after
 applying alert labels",
    "level": "ERROR",
    "name": "broken_alerting_rule",
    "group": "my_rule_group",
    "namespace": "my_namespace"
```

```
  },
  "component": "ruler"
}
```

Untuk lebih banyak contoh log dari Ruler atau Alertmanager, lihat dan. Pemecahan Masalah Penggaris Mengelola dan meneruskan peringatan di Amazon Managed Service untuk Prometheus dengan manajer peringatan

Gunakan offset dalam kueri untuk menangani penundaan konsumsi

Secara default, ekspresi dievaluasi tanpa offset (permintaan instan), menggunakan nilai pada waktu evaluasi. Jika konsumsi metrik tertunda, aturan perekaman mungkin tidak mewakili nilai yang sama seperti saat Anda mengevaluasi ekspresi secara manual setelah semua metrik dicerna.

> **Tip**
>
> Menggunakan pengubah offset dapat mengurangi masalah yang disebabkan oleh penundaan konsumsi. Untuk informasi selengkapnya, lihat Pengubah offset dalam dokumentasi Prometheus.

Contoh: Menangani metrik yang tertunda

Jika aturan Anda dievaluasi pada 12:00, tetapi sampel terbaru untuk metrik adalah dari 11:45 karena penundaan konsumsi, aturan tidak akan menemukan sampel pada stempel waktu 12:00. Untuk mengurangi ini, tambahkan offset, seperti:. **`my_metric_name offset 15m`**

Contoh: Menangani metrik dari berbagai sumber

Ketika metrik berasal dari sumber yang berbeda, seperti dua server, mereka mungkin tertelan pada waktu yang berbeda. Untuk mengurangi ini, bentuk ekspresi, seperti: **`metric_from_server_A / metric_from_server_B`**

Jika aturan mengevaluasi antara waktu konsumsi server A dan server B, Anda akan mendapatkan hasil yang tidak terduga. Menggunakan offset dapat membantu menyelaraskan waktu evaluasi.

Masalah dan solusi umum

Kesenjangan dalam merekam data aturan

Jika Anda melihat celah dalam data aturan perekaman dibandingkan dengan evaluasi manual (saat Anda langsung menjalankan ekspresi promQL asli aturan perekaman melalui API kueri atau UI), ini mungkin disebabkan oleh salah satu hal berikut:

1. Waktu evaluasi yang panjang — Kelompok aturan tidak dapat memiliki beberapa evaluasi simultan. Jika waktu evaluasi melebihi interval yang dikonfigurasi, evaluasi selanjutnya mungkin terlewatkan. Beberapa evaluasi yang terlewat berturut-turut melebihi interval yang dikonfigurasi dapat menyebabkan aturan perekaman menjadi basi. Untuk informasi lebih lanjut, lihat Kebuntuan dalam dokumentasi Prometheus. Anda dapat memantau durasi evaluasi menggunakan CloudWatch metrik `RuleGroupLastEvaluationDuration` untuk mengidentifikasi kelompok aturan yang terlalu lama untuk dievaluasi.
2. Memantau evaluasi yang tidak terjawab — AMP menyediakan `RuleGroupIterationsMissed` CloudWatch metrik untuk dilacak saat evaluasi dilewati. ListRules API menampilkan waktu evaluasi dan waktu evaluasi terakhir untuk setiap aturan/grup, yang dapat membantu mengidentifikasi pola evaluasi yang terlewat. Untuk informasi selengkapnya, lihat ListRules.

Rekomendasi: Pisahkan aturan menjadi grup terpisah

Untuk mengurangi durasi evaluasi, pisahkan aturan menjadi kelompok aturan terpisah. Aturan dalam grup dijalankan secara berurutan, sedangkan kelompok aturan dapat mengeksekusi secara paralel. Simpan aturan terkait yang saling bergantung satu sama lain dalam kelompok yang sama. Umumnya, kelompok aturan yang lebih kecil memastikan evaluasi yang lebih konsisten dan lebih sedikit kesenjangan.

Praktik terbaik untuk evaluasi aturan

1. Optimalkan ukuran grup aturan - Jaga agar kelompok aturan tetap kecil untuk memastikan evaluasi yang konsisten. Kelompokkan aturan terkait bersama, tetapi hindari kelompok aturan besar.
2. Tetapkan interval evaluasi yang sesuai - Seimbangkan antara peringatan tepat waktu dan beban sistem. Tinjau pola stabilitas metrik yang dipantau untuk memahami rentang fluktuasi normalnya.
3. Gunakan pengubah offset untuk metrik tertunda — Tambahkan offset untuk mengkompensasi penundaan konsumsi. Sesuaikan durasi offset berdasarkan pola konsumsi yang diamati.
4. Pantau kinerja evaluasi — Lacak `RuleGroupIterationsMissed` metrik. Tinjau waktu evaluasi di ListRules API.

5. Validasi ekspresi aturan - Pastikan ekspresi cocok persis antara definisi aturan dan kueri manual. Uji ekspresi dengan rentang waktu yang berbeda untuk memahami perilaku.
6. Tinjau aturan kesehatan secara teratur — Periksa kesalahan dalam evaluasi aturan. Pantau log yang dijual untuk masalah berulang.

Dengan mengikuti langkah-langkah pemecahan masalah dan praktik terbaik ini, Anda dapat mengidentifikasi dan menyelesaikan masalah umum dengan evaluasi aturan di Amazon Managed Service for Prometheus.

Pemecahan Masalah Penggaris

Dengan menggunakanPantau Layanan Terkelola Amazon untuk acara Prometheus dengan Log CloudWatch , Anda dapat memecahkan masalah terkait Pengelola 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 diambil

Gunakan pesan kesalahan untuk memecahkan masalah eksekusi aturan.

Mengelola dan meneruskan peringatan di Amazon Managed Service untuk Prometheus dengan manajer peringatan

Saat aturan peringatan yang dijalankan Amazon Managed Service untuk Prometheus diaktifkan, manajer peringatan menangani peringatan yang dikirim. Ini menghapus duplikat, 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.
- Silences - 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 file template terpisah. 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

- Memahami izin IAM yang diperlukan untuk bekerja dengan manajer peringatan
- Buat konfigurasi pengelola peringatan di Amazon Managed Service untuk Prometheus untuk mengelola dan merutekan peringatan
- Teruskan peringatan ke penerima peringatan dengan pengelola peringatan di Amazon Managed Service untuk Prometheus
- Unggah file konfigurasi pengelola peringatan Anda ke Amazon Managed Service untuk Prometheus
- Integrasikan peringatan dengan Grafana Terkelola Amazon atau Grafana open source
- Memecahkan masalah manajer peringatan dengan Log CloudWatch

Memahami izin IAM yang diperlukan untuk bekerja dengan manajer peringatan

Anda harus memberi pengguna izin untuk menggunakan pengelola peringatan 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.

JSON

```
{
    "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": "*"
        }
    ]
}
```

Buat konfigurasi pengelola peringatan di Amazon Managed Service untuk Prometheus untuk mengelola dan merutekan 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.

> **Note**
>
> Anda juga dapat mengedit file konfigurasi langsung di Amazon Managed Service untuk konsol Prometheus, tetapi harus tetap mengikuti format yang ditentukan di sini. Untuk informasi selengkapnya tentang mengunggah atau mengedit file konfigurasi, lihatUnggah file konfigurasi pengelola peringatan Anda ke Amazon Managed Service untuk Prometheus.

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 }}
```

Teruskan peringatan ke penerima peringatan dengan pengelola peringatan di Amazon Managed Service untuk Prometheus

Ketika peringatan dinaikkan oleh aturan peringatan, itu dikirim ke manajer peringatan. Manajer peringatan melakukan fungsi seperti menghilangkan duplikasi peringatan, menghambat peringatan selama pemeliharaan, atau mengelompokkannya sesuai kebutuhan. Kemudian meneruskan peringatan sebagai pesan ke penerima peringatan. Anda dapat mengatur penerima peringatan yang dapat memberi tahu operator, memiliki respons otomatis, atau menanggapi peringatan dengan cara lain.

Satu-satunya penerima peringatan yang didukung di Amazon Managed Service untuk Prometheus adalah Amazon Simple Notification Service (Amazon SNS). Untuk informasi lebih lanjut, lihat Apa itu Amazon SNS? . Amazon SNS dapat digunakan untuk menanggapi peringatan dengan berbagai cara, termasuk meneruskan ke sistem lain, seperti email, SMS, atau titik akhir HTTP.

Topik berikut menjelaskan tugas yang terkait dengan membuat dan mengonfigurasi penerima peringatan Amazon SNS Anda.

Topik

- Membuat topik Amazon SNS baru untuk digunakan sebagai penerima peringatan di Amazon Managed Service untuk Prometheus
- Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan peringatan ke topik Amazon SNS Anda
- Konfigurasikan pengelola peringatan untuk mengirim pesan ke topik Amazon SNS Anda
- Konfigurasikan pengelola peringatan untuk mengirim pesan ke Amazon SNS sebagai JSON
- Konfigurasikan Amazon SNS untuk mengirim pesan peringatan ke tujuan lain
- Memahami aturan validasi pesan Amazon SNS

Membuat topik Amazon SNS baru untuk digunakan sebagai penerima peringatan di Amazon Managed Service untuk Prometheus

Anda dapat menggunakan topik Amazon SNS yang ada sebagai penerima peringatan untuk Amazon Managed Service for Prometheus, atau Anda dapat 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.

Baik Anda menggunakan topik Amazon SNS baru atau yang sudah ada, Anda memerlukan Nama Sumber Daya Amazon (ARN) dari topik Amazon SNS Anda untuk menyelesaikan tugas-tugas berikut.

Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan peringatan 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 akan memberikan izin itu. Ini termasuk `Condition` pernyataan untuk membantu mencegah masalah keamanan yang dikenal sebagai masalah wakil yang bingung. `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 Amazon SNS Anda diaktifkan enkripsi sisi layanan (SSE), Anda harus mengizinkan Layanan Terkelola Amazon untuk Prometheus mengirim pesan ke topik terenkripsi ini dengan menambahkan `kms:GenerateDataKey*` dan `kms:Decrypt` izin ke kebijakan kunci kunci yang digunakan untuk mengenkripsi topik. AWS KMS

Misalnya, Anda dapat menambahkan yang berikut ini ke kebijakan:

```
{
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "aps.amazonaws.com"
    },
    "Action": [
      "kms:GenerateDataKey*",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }]
}
```

Untuk informasi selengkapnya, lihat Izin AWS KMS untuk Topik SNS.

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 memilik hak akses lebih tinggi untuk melakukan tindakan tersebut. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila 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 "confused deputy" adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap 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. Misalnya, `arn:aws:servicename::123456789012:*`.

Kebijakan yang ditampilkan di Memberikan izin Layanan Terkelola Amazon untuk Prometheus untuk mengirim pesan peringatan 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.

Konfigurasikan pengelola peringatan untuk mengirim pesan ke topik Amazon SNS Anda

Setelah memiliki topik Amazon SNS tipe Standar (baru atau yang sudah ada), Anda dapat menambahkannya ke konfigurasi manajer peringatan sebagai penerima peringatan. Manajer peringatan dapat meneruskan peringatan Anda ke penerima peringatan yang dikonfigurasi. Untuk menyelesaikan 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: Wilayah AWS
      topic_arn: ARN_of_SNS_topic
      subject: yoursubject
      attributes:
        key: yourkey
        value: yourvalue
```

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 Konfigurasikan pengelola peringatan untuk mengirim pesan ke Amazon SNS sebagai JSON di panduan ini.

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

Konfigurasikan pengelola peringatan untuk mengirim pesan ke Amazon SNS sebagai JSON

Secara default, Amazon Managed Service for Prometheus alert manager mengeluarkan pesan dalam format daftar teks biasa. Ini bisa lebih sulit bagi layanan lain untuk diuraikan. Anda dapat mengonfigurasi manajer peringatan untuk mengirim peringatan dalam format JSON sebagai gantinya. JSON dapat mempermudah proses pesan di hilir dari Amazon SNS AWS Lambda di atau di titik akhir penerima webhook. 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 Anda`template_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, lihatBuat konfigurasi pengelola peringatan di Amazon Managed Service untuk Prometheus untuk mengelola dan merutekan peringatan.
- Mengunggah file konfigurasi Anda, lihatUnggah file konfigurasi pengelola peringatan Anda ke Amazon Managed Service untuk Prometheus.

Konfigurasikan Amazon SNS untuk mengirim pesan peringatan ke tujuan lain

Amazon Managed Service untuk Prometheus hanya dapat mengirim pesan peringatan ke Amazon Simple Notification Service (Amazon SNS). Untuk mengirim pesan tersebut ke tujuan lain, seperti email, webhook, Slack, atau OpsGenie, Anda harus mengonfigurasi Amazon SNS untuk meneruskan pesan ke titik akhir tersebut.

Bagian berikut yang menjelaskan konfigurasi Amazon SNS untuk meneruskan peringatan ke tujuan lain.

Topik

- Email
- Webhook
- Kendur
- 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 penggunaan 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.

Memahami aturan validasi pesan Amazon SNS

Amazon Simple Notification Service (Amazon SNS) memerlukan pesan untuk memenuhi standar tertentu. Pesan yang tidak memenuhi standar ini akan dimodifikasi saat diterima. Pesan peringatan

akan divalidasi, dipotong, atau dimodifikasi, jika perlu, oleh penerima Amazon 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 true.
  - 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 tidak boleh 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 true.
  - 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 berisi karakter kontrol atau non-ASCII.
  - Jika subjek berisi karakter kontrol atau karakter non-ASCII, SNS menggantikan subjek dengan Kesalahan - berisi karakter kontrol - atau non-ASCII.
  - Untuk subjek email SNS, hapus karakter kontrol, seperti baris baru:. \n
- Subjek bukan ASCII.
  - Subjek akan diganti dengan Error - berisi karakter ASCII yang tidak dapat dicetak.
  - Satu atribut pesan akan ditambahkan dengan kunci yang dimodifikasi dan nilai Subject: Error - berisi karakter ASCII yang tidak dapat dicetak.
- Subjek telah terpotong.
  - Subjek akan memiliki konten terpotong.
  - Satu atribut pesan akan ditambahkan dengan kunci yang dimodifikasi dan nilai Subject: Error - Subject telah dipotong dari ***X*** karakter, karena melebihi batas ukuran karakter 100.
- Atribut pesan memiliki kunci/nilai yang tidak valid.
  - Atribut pesan tidak valid akan dihapus.

  - Satu atribut 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.

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

Setelah Anda mengetahui apa yang Anda inginkan di file konfigurasi manajer Peringatan, Anda dapat membuat dan mengeditnya di dalam konsol, atau Anda dapat mengunggah file yang ada dengan Amazon Managed Service untuk konsol Prometheus atau. AWS CLI

> **Note**
>
> Jika Anda menjalankan kluster Amazon EKS, Anda juga dapat mengunggah file konfigurasi manajer Peringatan menggunakan AWS Pengontrol untuk Kubernetes.

Untuk menggunakan Amazon Managed Service untuk konsol Prometheus untuk mengedit atau mengganti 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.

   > **Note**
   >
   > Jika ruang kerja memiliki definisi manajer peringatan yang ingin Anda ganti, pilih Ubah sebagai gantinya.

5. Pilih Pilih file, pilih file definisi manajer peringatan, dan pilih Lanjutkan.

> **Note**
>
> Sebagai alternatif, Anda dapat membuat file baru dan mengeditnya langsung di konsol, dengan memilih opsi Buat definisi. Ini akan membuat contoh konfigurasi default yang Anda edit sebelum mengunggah.

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 file://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.

Integrasikan 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 kredensyal yang ada Akun AWS dan IAM untuk membuat Layanan Terkelola Amazon untuk peran Prometheus dan IAM secara terprogram.

  Untuk informasi selengkapnya tentang membuat kredenal IAM Akun AWS dan IAM, lihat. Mengatur AWS
- Anda harus memiliki Layanan Terkelola Amazon untuk ruang kerja Prometheus, dan memasukkan data ke dalamnya. Untuk menyiapkan ruang kerja baru, lihatBuat Layanan Terkelola Amazon untuk ruang kerja Prometheus. 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. Mengelola dan meneruskan peringatan di Amazon Managed Service untuk Prometheus dengan manajer peringatan Untuk informasi selengkapnya tentang aturan, lihatMenggunakan aturan untuk memodifikasi atau memantau metrik saat diterima.
- 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. Langkah 2: 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 AWS otentikasi 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.

Memecahkan masalah manajer peringatan dengan Log CloudWatch

Dengan menggunakanPantau Layanan Terkelola Amazon untuk acara Prometheus dengan Log CloudWatch , Anda dapat memecahkan masalah terkait Pengelola Peringatan dan Penggaris. Bagian ini berisi topik pemecahan masalah terkait Alert Manager.

Topik

- Peringatan peringatan aktif
- Peringatan ukuran grup agregasi peringatan
- Ukuran peringatan peringatan terlalu besar
- Peringatan konten kosong
- Peringatan tidak valid key/value
- Peringatan batas pesan
- Tidak ada kesalahan kebijakan berbasis sumber daya
- Peringatan non ASCII
- Tidak berwenang untuk menelepon KMS
- Kesalahan template

Peringatan peringatan aktif

Ketika log berisi peringatan berikut

```
{
    "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
    "message": {
        "log": "too many alerts, limit: 1000",
        "level": "WARN"
    },
    "component": "alertmanager"
}
```

Ini berarti bahwa kuota peringatan Aktif manajer Peringatan terlampaui.

Tindakan yang harus diambil

Minta peningkatan kuota. Masuk ke AWS Management Console dan buka konsol Service Quotas di. https://console.aws.amazon.com/servicequotas/

Peringatan ukuran grup agregasi peringatan

Ketika log berisi peringatan berikut

```
{
    "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
    "message": {
        "log": "Too many aggregation groups, cannot create new group for alert,
 groups=1000, limit=1000, alert=sample-alert",
        "level": "WARN"
    },
    "component": "alertmanager"
}
```

Ini berarti bahwa kuota ukuran grup agregasi Alert manager Alert telah terlampaui.

Tindakan yang harus diambil

Kurangi ukuran grup agregasi Alert dengan menggunakan `group_by` parameter. Untuk informasi selengkapnya, lihat Pengaturan terkait rute di dokumentasi Prometheus.

Anda juga dapat meminta peningkatan kuota. Masuk ke AWS Management Console dan buka konsol Service Quotas di. https://console.aws.amazon.com/servicequotas/

Ukuran peringatan peringatan terlalu besar

Ketika log berisi peringatan berikut

```
{
    "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
    "message": {
        "log": "alerts too big, total size limit: 20000000 bytes",
        "level": "WARN"
    },
    "component": "alertmanager"
}
```

Ini berarti bahwa Alert manager Alerts per ruang kerja, dalam ukuran kuota telah terlampaui.

Tindakan yang harus diambil

Hapus anotasi dan label yang tidak perlu untuk mengurangi ukuran peringatan.

Peringatan konten kosong

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "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 tidak valid **key/value**

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "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 keys/values karena 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-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "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-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "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. Buat Kebijakan Akses SNS yang memberikan layanan `aps.amazonaws.com` (Amazon Managed Service for Prometheus) akses ke topik Amazon SNS Anda. Untuk informasi selengkapnya tentang Kebijakan Akses SNS, lihat Menggunakan Bahasa Kebijakan Akses dan Contoh kasus untuk kontrol akses Amazon SNS di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Peringatan non ASCII

Ketika log berisi peringatan berikut

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "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.

Tidak berwenang untuk menelepon KMS

Ketika log berisi AWS KMS kesalahan berikut

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "Notify for alerts failed, AMP is not authorized to call KMS",
       "level": "ERROR"
   },
   "component": "alertmanager"
}
```

Tindakan yang harus diambil

Validasi bahwa kebijakan kunci kunci yang digunakan untuk mengenkripsi topik Amazon SNS memungkinkan layanan Amazon Managed Service for Prometheus service principal untuk melakukan tindakan berikut:, dan. `aps.amazonaws.com` `kms:GenerateDataKey*` `kms:Decrypt` Untuk informasi selengkapnya, lihat Izin AWS KMS untuk Topik SNS.

Kesalahan template

Ketika log berisi kesalahan berikut

```
             {
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
   "message": {
       "log": "Notify for alerts failed. There is an error in a receiver that is using
 templates in the AlertManager definition. Make sure that the syntax is correct and
 only template functions and variables that exist are used in the receiver 'default',
 sns_configs position #2, section 'attributes'"
       "level": "ERROR"
   },
   "component": "alertmanager"
}
```

Ini berarti bahwa ada kesalahan dalam template yang digunakan dalam AlertManager definisi. Entri kesalahan berisi petunjuk tentang penerima apa, posisi di sns_configs dan properti yang berisi kesalahan.

Tindakan yang harus diambil

Validasi definisi Alert Manager Anda. Pastikan sintaksnya benar dan Anda mereferensikan variabel template dan fungsi yang ada. Untuk informasi selengkapnya, lihat Referensi Template Pemberitahuan di dokumentasi sumber terbuka Prometheus.

Pencatatan dan pemantauan Amazon Managed Service untuk ruang kerja Prometheus

Amazon Managed Service untuk Prometheus menggunakan CloudWatch Amazon untuk menyediakan data tentang operasinya. Anda dapat menggunakan CloudWatch metrik untuk mempelajari tentang penggunaan sumber daya dan permintaan ke Layanan Terkelola Amazon untuk ruang kerja Prometheus. Anda dapat mengaktifkan dukungan CloudWatch Log untuk mendapatkan log untuk peristiwa yang terjadi di ruang kerja Anda.

Topik-topik berikut menjelaskan penggunaan CloudWatch secara lebih rinci.

Menggunakan CloudWatch metrik untuk memantau Layanan Terkelola Amazon untuk sumber daya Prometheus

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*	RemoteWriteTPS	`AWS/Usage`	Operasi tulis jarak jauh per detik
ResourceCount	HAReplicaGroupCount	`AWS/Usage`	Jumlah grup replika ketersediaan tinggi
ResourceCount*	QueryMetricsTPS	`AWS/Usage`	Operasi kueri per detik
ResourceCount	IngestionRate	`AWS/Usage`	Tingkat konsumsi sampel Satuan: hitung per detik

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
			Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
ResourceCount	ActiveSeries	AWS/Usage	Jumlah seri aktif per ruang kerja Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
ResourceCount	ActiveAlerts	AWS/Usage	Jumlah peringatan aktif per ruang kerja Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
ResourceCount	SizeOfAlerts	AWS/Usage	Ukuran total semua peringatan di ruang kerja, dalam byte Unit: byte Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
ResourceCount	SuppressedAlerts	AWS/Usage	Jumlah peringatan dalam keadaan ditekan per ruang kerja. Peringatan dapat ditekan oleh keheningan atau penghambatan. Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
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. Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
ResourceCount	AllAlerts	`AWS/Usage`	Jumlah peringatan di negara bagian mana pun per ruang kerja. Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
ResourceCount	AllRules	`AWS/Usage`	Jumlah aturan di setiap negara bagian per ruang kerja. Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
ActiveSeriesPerLabelSet	-	`AWS/Prometheus`	Penggunaan seri aktif saat ini untuk setiap set label yang ditentukan pengguna Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
ActiveSeriesLimitPerLabelSet	-	`AWS/Prometheus`	Nilai batas seri aktif saat ini untuk setiap set label yang ditentukan pengguna Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
AlertManagerAlertsReceived	-	`AWS/Prometheus`	Total lansiran yang berhasil diterima oleh manajer peringatan Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
AlertManagerNotificationsFailed	-	`AWS/Prometheus`	Jumlah pengiriman peringatan yang gagal Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
AlertManagerNotificationsThrottled	-	`AWS/Prometheus`	Jumlah peringatan yang dibatasi Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
Discarded Samples**	-	AWS/Prometheus	Jumlah sampel yang dibuang dengan alasan Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
Discarded SamplesPerLabelSet	-	AWS/Prometheus	Jumlah sampel yang dibuang untuk setiap set label yang ditentukan pengguna Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
Ingestion RatePerLabelSet	-	AWS/Prometheus	Tingkat konsumsi untuk setiap set label yang ditentukan pengguna Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
QuerySamplesProcessed	-	`AWS/Prometheus`	Jumlah sampel kueri yang diproses Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
RuleEvaluations	-	`AWS/Prometheus`	Jumlah total evaluasi aturan Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
RuleEvaluationFailures	-	`AWS/Prometheus`	Jumlah kegagalan evaluasi aturan dalam interval Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah
RuleGroupIterationsMissed	-	`AWS/Prometheus`	Jumlah iterasi Grup Aturan yang terlewatkan dalam interval. Satuan: hitung Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

CloudWatch nama metrik	Nama sumber daya	CloudWatch namespace	Deskripsi
RuleGroup LastEvaluationDuration	-	`AWS/Prometheus`	Durasi evaluasi terakhir kelompok aturan. Unit: detik Statistik yang Valid: Rata-rata, Minimum, Maksimum, Jumlah

* Metrik TPS dihasilkan setiap menit dan merupakan rata-rata per detik selama menit itu. Periode burst pendek tidak akan ditangkap dalam metrik TPS.

** Beberapa alasan yang menyebabkan sampel dibuang adalah sebagai berikut.

Alasan	Arti
greater_than_max_sample_age	Membuang sampel yang lebih tua dari satu jam.
new-value-for-timestamp	Sampel duplikat dikirim dengan stempel waktu yang berbeda dari yang direkam sebelumnya.
per_labelset_series_limit	Pengguna telah mencapai jumlah total seri aktif per batas yang ditetapkan label.
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 rusak 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.

Alasan	Arti
hilang_metric_name	Nama metrik tidak disediakan.
metric_name_invalid	Nama metrik yang diberikan tidak valid.
label_invalid	Label tidak valid disediakan.
duplikate_label_names	Nama label duplikat disediakan.

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

> **Note**
>
> `RuleGroupIterationsMissed`,`RuleEvaluationFailures`, dan `RuleGroupLastEvaluationDuration` 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.

Pantau Layanan Terkelola Amazon untuk acara Prometheus dengan Log CloudWatch

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 API AWS CLI . `create-logging-configuration`

Prasyarat

Sebelum meneleponcreate-logging-configuration, lampirkan kebijakan berikut atau izin yang setara ke ID atau peran yang akan Anda gunakan untuk mengonfigurasi CloudWatch Log.

JSON

```
{
    "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 file AWS CLI.

Untuk mengkonfigurasi logging menggunakan AWS CLI

- Menggunakan AWS 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.

Mengelola biaya kueri di Amazon Managed Service untuk Prometheus

Amazon Managed Service for Prometheus menawarkan kemampuan untuk membatasi biaya kueri dengan memberikan batasan berapa banyak Sampel Kueri yang Diproses (QSP) dapat digunakan oleh satu kueri. Anda dapat mengonfigurasi dua jenis ambang batas untuk QSP, peringatan dan kesalahan untuk membantu mengelola dan mengontrol biaya kueri secara efektif.

Saat kueri mencapai ambang peringatan, pesan peringatan muncul di respons kueri API. Untuk kueri yang dilihat melalui Grafana Terkelola Amazon, peringatan akan terlihat di UI Grafana Terkelola Amazon, membantu pengguna mengidentifikasi kueri mahal. Kueri yang mencapai ambang kesalahan tidak dikenakan biaya dan akan ditolak dengan kesalahan.

Selain pembatasan kueri, Amazon Managed Service untuk Prometheus menawarkan kemampuan untuk mencatat data kinerja kueri ke Log. CloudWatch Fitur ini memungkinkan Anda menganalisis kueri secara detail, membantu Anda mengoptimalkan Layanan Terkelola Amazon untuk kueri Prometheus dan mengelola biaya dengan lebih efektif. Pencatatan kueri menangkap informasi tentang kueri yang melebihi ambang batas Sampel Kueri yang Diproses (QSP) tertentu. Data ini kemudian dipublikasikan ke CloudWatch Log, memungkinkan Anda untuk menyelidiki dan menganalisis kinerja kueri. Kueri yang dicatat mencakup kueri API dan kueri Aturan. Secara default, pencatatan kueri dinonaktifkan untuk meminimalkan penggunaan CloudWatch Log yang tidak perlu. Anda dapat mengaktifkan fitur ini bila diperlukan untuk analisis kueri.

Topik

Mengkonfigurasi pencatatan kueri

Anda dapat mengonfigurasi pencatatan kueri di Amazon Managed Service untuk konsol Prometheus atau di AWS CLI dengan memanggil permintaan API. `create-query-logging-configuration` Badan API ini berisi daftar tujuan, tetapi untuk saat ini, kami hanya mendukung CloudWatch Log sebagai tujuan dan tujuan harus berisi tepat satu elemen dengan CloudWatch konfigurasi.

Prasyarat

Pastikan `logGroup` sudah dibuat. ID atau peran yang digunakan untuk mengonfigurasi harus memiliki kebijakan berikut atau izin yang setara.

JSON

```
{
"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:CreateQueryLoggingConfiguration",
                "aps:UpdateQueryLoggingConfiguration",
                "aps:DescribeQueryLoggingConfiguration",
                "aps:DeleteQueryLoggingConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

Konfigurasikan CloudWatch Log

Anda dapat mengonfigurasi CloudWatch Log dengan masuk ke Amazon Managed Service untuk Prometheus menggunakan file atau file. AWS Management Console AWS CLI

Untuk mengonfigurasi pencatatan kueri menggunakan Amazon Managed Service untuk konsol Prometheus

1. Arahkan ke tab Log di panel detail ruang kerja Anda.

2. Di bawah Wawasan Kueri, pilih Buat.
3. Pilih drop-down Grup Log dan pilih grup log untuk mempublikasikan log Anda.

   Anda juga dapat membuat grup log baru di CloudWatch konsol.
4. Masukkan Ambang Batas (QSP).
5. Pilih Simpan.

Untuk mengkonfigurasi pencatatan kueri AWS CLI menggunakan menggunakan perintah

```
aws amp create-query-logging-configuration \
--workspace-id my_workspace_ID \
--destinations '[{"cloudWatchLogs":{"logGroupArn":"$my-log-group-arn"},"filters":
{"qspThreshold":$qspThreshold}}]'
```

Untuk informasi tentang cara memperbarui, menghapus, dan menjelaskan operasi, lihat Layanan Terkelola Amazon untuk Referensi API Prometheus.

Mengkonfigurasi ambang batas pelambatan kueri

Untuk mengonfigurasi ambang QSP, Anda harus memberikan parameter kueri di API. QueryMetrics

- max_samples_processed_warning_threshold — Menetapkan ambang peringatan untuk sampel kueri yang diproses
- max_samples_processed_error_threshold — Menetapkan ambang kesalahan untuk sampel kueri yang diproses

Untuk pengguna Grafana Terkelola Amazon, Anda dapat menggunakan konfigurasi sumber data grafana untuk menerapkan batasan ke semua kueri dari sumber data:

1. Jelajahi Layanan Terkelola Amazon untuk konfigurasi sumber data Prometheus di Grafana Terkelola Amazon.
2. Di bawah Parameter kueri khusus, tambahkan header ambang batas.
3. Pilih Simpan.

Konten log

Untuk kueri yang berasal dari aturan, Anda akan melihat informasi berikut tentang kueri di Log: CloudWatch

```
{
    workspaceId: "workspace_id",
    message: {
        query: "avg(rate(go_goroutines[1m])) > 1",
        name: "alert_rule",
        kind: "alerting",
        group: "test-alert",
        namespace: "test",
        samples: "59321",
    },
    component: "ruler"
}
```

Untuk kueri yang berasal dari panggilan API, Anda akan melihat informasi berikut tentang kueri di Log: CloudWatch

```
{
    workspaceId: "ws-5e7658c2-7ccf-4c30-9de9-2ab26fa30639",
    message: {
        query: "sum by (instance) (go_memstats_alloc_bytes{job=\"node\"})",
        queryType: "range",
        start: "1683308700000",
        end: "1683913500000",
        step: "300000",
        samples: "11496",
        userAgent: "AWSPrometheusDPJavaClient/2.0.436.0 ",
        dashboardUid: "11234",
        panelId: "12"
    },
    component: "query-frontend"
}
```

Batasan

Batas ukuran kebijakan — Kebijakan sumber daya CloudWatch log dibatasi hingga 5120 karakter. Ketika CloudWatch Log mendeteksi bahwa kebijakan mendekati batas ukuran, secara otomatis

mengaktifkan grup log yang memulai`/aws/vendedlogs/`. Saat Anda mengaktifkan pencatatan kueri, 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/`.

Memahami dan mengoptimalkan biaya di Amazon Managed Service untuk Prometheus

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 dalam 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 substansif mengurangi biaya Anda.

Untuk informasi tentang cara mengonfigurasi periode retensi ruang kerja, lihatKonfigurasikan ruang kerja Anda.

Bagaimana saya bisa menjaga biaya kueri peringatan saya tetap rendah?

Peringatan membuat kueri terhadap data Anda, yang menambah biaya kueri Anda. Berikut adalah beberapa strategi yang dapat Anda gunakan untuk mengoptimalkan kueri peringatan Anda, dan menjaga biaya Anda lebih rendah.

- Gunakan Layanan Terkelola Amazon untuk peringatan Prometheus — Sistem peringatan di luar Layanan Terkelola Amazon untuk Prometheus mungkin memerlukan kueri tambahan untuk menambahkan ketahanan atau ketersediaan tinggi, karena layanan eksternal menanyakan metrik dari beberapa zona ketersediaan atau wilayah. Ini termasuk peringatan di Grafana untuk ketersediaan tinggi. Ini dapat melipatgandakan biaya Anda dengan tiga kali atau lebih. Peringatan di Amazon Managed Service untuk Prometheus dioptimalkan dan akan memberi Anda ketersediaan dan ketahanan yang tinggi dengan jumlah kueri paling sedikit.

  Sebaiknya gunakan peringatan asli di Amazon Managed Service untuk Prometheus daripada sistem peringatan eksternal.
- Optimalkan interval peringatan Anda — Salah satu cara cepat untuk mengoptimalkan kueri peringatan Anda adalah dengan meningkatkan interval penyegaran otomatis. Jika Anda memiliki peringatan yang menanyakan setiap menit, tetapi hanya diperlukan setiap lima menit, meningkatkan interval penyegaran otomatis dapat menghemat lima kali biaya kueri untuk peringatan itu.

- Gunakan lookback yang optimal — Jendela lookback yang lebih besar dalam kueri Anda meningkatkan biaya kueri, karena menarik lebih banyak data. Pastikan bahwa jendela lookback dalam kueri PromQL Anda berukuran cukup untuk data yang perlu Anda waspadai. Misalnya, dalam aturan berikut, ekspresi menyertakan jendela lookback sepuluh menit:

```
  - alert: metric:alerting_rule
    expr: avg(rate(container_cpu_usage_seconds_total[10m])) > 0
    for: 2m
```

  Mengubah `expr` to `avg(rate(container_cpu_usage_seconds_total[5m])) > 0` dapat membantu mengurangi biaya kueri Anda.

Secara umum, lihat aturan peringatan Anda dan pastikan Anda memberi tahu metrik terbaik untuk layanan Anda. Sangat mudah untuk membuat peringatan yang tumpang tindih pada metrik yang sama atau beberapa peringatan yang memberi Anda informasi yang sama, terutama saat Anda menambahkan peringatan dari waktu ke waktu. Jika Anda menemukan bahwa Anda sering melihat grup peringatan terjadi pada saat yang sama, ada kemungkinan bahwa Anda dapat mengoptimalkan peringatan Anda dan tidak menyertakan semuanya.

Saran ini dapat membantu Anda mengurangi biaya. Pada akhirnya, Anda harus menyeimbangkan biaya dengan membuat rangkaian peringatan yang tepat untuk memahami keadaan sistem Anda.

Untuk informasi selengkapnya tentang peringatan di Amazon Managed Service for Prometheus, lihat. Mengelola dan meneruskan peringatan di Amazon Managed Service untuk Prometheus dengan manajer peringatan

Metrik apa yang dapat saya gunakan untuk memantau biaya saya?

Pantau `IngestionRate` di Amazon CloudWatch untuk melacak biaya konsumsi Anda.

> **Note**
>
> `IngestionRate`memberikan nilai estimasi dan mungkin tidak sama persis dengan biaya penagihan akhir Anda.

Untuk informasi selengkapnya tentang pemantauan Amazon Managed Service untuk metrik Prometheus, lihat. CloudWatch Menggunakan CloudWatch metrik untuk memantau Layanan Terkelola Amazon untuk sumber daya Prometheus

Bisakah saya memeriksa tagihan saya kapan saja?

AWS Cost and Usage Report Lacak 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 untuk penggunaan dalam suatu wilayah, bukan di seluruh wilayah. Penggunaan dalam suatu wilayah harus mencapai tingkat berikutnya untuk menggunakan tarif yang lebih rendah.
> - Dalam organisasi di AWS Organizations, penggunaan tingkat dihitung per akun pembayar, bukan per akun (akun pembayar selalu merupakan akun manajemen organisasi). Ketika total metrik yang dicerna (dalam suatu wilayah) untuk semua akun di organisasi mencapai tingkat berikutnya, semua akun dikenakan tarif yang lebih rendah.

Saya menghapus semua Layanan Terkelola Amazon saya untuk ruang kerja Prometheus, tetapi sepertinya saya masih dikenakan biaya. Apa yang mungkin terjadi?

Satu kemungkinan dalam kasus ini adalah Anda masih memiliki pencakar AWS terkelola yang disiapkan untuk mengirim metrik ke ruang kerja yang dihapus. Ikuti instruksi untukTemukan dan hapus pencakar.

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 cara menelan metrik dari penggunaan Amazon Data Firehose. CloudWatch Ini juga menjelaskan pengaturan dan pengelolaan Amazon Managed Service untuk Prometheus AWS dengan modul Observability Accelerator Terraform, atau dengan menggunakan Controller untuk Kubernetes. AWS

Topik

- Mengintegrasikan dengan pemantauan biaya Amazon EKS
- Siapkan Amazon Managed Service untuk Prometheus dengan Observability Accelerator AWS
- Kelola Layanan Terkelola Amazon untuk AWS Prometheus dengan Pengontrol untuk Kubernetes
- Mengintegrasikan CloudWatch metrik dengan Amazon Managed Service untuk Prometheus

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.

Siapkan Amazon Managed Service untuk Prometheus dengan Observability Accelerator AWS

AWS menyediakan 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 AWS lainnya. OpenTelemetry Untuk membantu Anda menggunakan alat ini bersama-sama, AWS sediakan modul Terraform yang mengonfigurasi observabilitas dengan layanan ini, yang disebut Observability Accelerator.AWS

AWS Observability 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 AWS Observability 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 AWS penyedia di dokumentasi Terraform.

Menggunakan contoh pemantauan infrastruktur

AWS Observability 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.
- AWS Distro 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.

Kelola Layanan Terkelola Amazon untuk AWS Prometheus dengan Pengontrol 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 (CRDs) 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 memiliki yang AWS CLIdiinstal. 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 yang benar *`region`* untuk sistem Anda, seperti`us-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, *`REGION`* dan dengan Wilayah AWS yang Anda gunakan. 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`, atau`RuleGroupNamespace`. 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 sajaprometheus-config.yaml. Ganti *account* dengan ID akun Anda, *workspaceURL/* dengan URL yang baru saja Anda temukan, dan *region* 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 (CRDs), 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 Amazon Managed Service untuk Prometheus

Ini dapat membantu untuk memiliki semua metrik Anda di satu tempat. Layanan Terkelola Amazon untuk Prometheus tidak secara otomatis menelan metrik Amazon. CloudWatch Namun, Anda dapat menggunakan Amazon Data Firehose dan AWS Lambda untuk mendorong CloudWatch metrik ke Amazon Managed Service untuk Prometheus.

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/rumah #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. Di 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 adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Managed Service for Prometheus, AWS lihat Layanan dalam Lingkup berdasarkan Layanan Program Kepatuhan dalam Lingkup oleh Kepatuhan.
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang 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. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host 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, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan 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 AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

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. AWS CLI AWS SDKs 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 meliputi:

- Nilai metrik
- Label metrik (atau pasangan nilai kunci arbitrer) yang membantu mengidentifikasi dan mengklasifikasikan data
- Stempel waktu untuk sampel data

Penyewa unik IDs mengisolasi data dari pelanggan yang berbeda. Ini IDs membatasi data pelanggan yang dapat diakses. Pelanggan tidak dapat mengubah penyewa. IDs

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 for Prometheus mendukung pembuatan kunci terkelola pelanggan untuk mengenkripsi data Anda. Untuk informasi selengkapnya tentang kunci yang digunakan

Amazon Managed Service for Prometheus secara default, dan cara menggunakan kunci terkelola pelanggan Anda sendiri, lihat. Enkripsi diam

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

- AWS kunci 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 tentang AWS KMS, lihat Apa itu AWS 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. CreateGrant AWS 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 yang dikelola pelanggan simetris dengan menggunakan AWS Management Console, atau. AWS KMS APIs 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 Service Pengembang.
- Untuk informasi selengkapnya tentang akses kunci pemecahan masalah, lihat Panduan AWS Key Management Service Pengembang.

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 KMS menggunakan 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 terkelola 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

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan 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 Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas

- Mengelola akses menggunakan kebijakan
- Bagaimana Amazon Managed Service untuk Prometheus bekerja dengan IAM
- Contoh kebijakan berbasis identitas 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 Anda akan memberi Anda kredensi 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 untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari selengkapnya tentang cara 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

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensi yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

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 kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar 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, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS 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 aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins 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 dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk

mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan

beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

- Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan 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 tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus

menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke 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 batasan 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 akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.
- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa

memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, 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	Ya
Tindakan kebijakan	Ya

Fitur IAM	Layanan Terkelola Amazon untuk dukungan Prometheus
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Tidak
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Sesi akses teruskan (FAS)	Tidak
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Layanan Terkelola Amazon untuk Prometheus dan layanan AWS lainnya dengan sebagian besar fitur IAM, AWS lihat 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, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, 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: Ya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan 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 tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke principal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon Managed Service untuk Prometheus

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki

nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi 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 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 AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk

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 menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar Layanan Terkelola Amazon untuk jenis sumber daya Prometheus beserta jenisnya, lihat Sumber daya yang ditentukan oleh Amazon Managed Service for Prometheus di Referensi Otorisasi Layanan. ARNs 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 kondisi kebijakan khusus layanan: Tidak

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi 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 menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global 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 (ACLs) di Amazon Managed Service untuk Prometheus

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka 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 menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensi sementara dengan Amazon Managed Service untuk Prometheus

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Teruskan sesi akses untuk Amazon Managed Service untuk Prometheus

Mendukung sesi akses maju (FAS): Tidak

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.

Peran layanan untuk Amazon Managed Service untuk Prometheus

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah 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 terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait 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. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Layanan Terkelola Amazon untuk Prometheus, termasuk format untuk setiap jenis sumber daya, lihat Tindakan, sumber daya, dan kunci kondisi untuk Layanan Terkelola Amazon untuk Prometheus dalam Referensi Otorisasi Layanan. ARNs

Topik

- Praktik terbaik kebijakan
- Menggunakan Amazon Managed Service untuk konsol Prometheus
- Mengizinkan pengguna 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 membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam 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 melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua

permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.

- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam 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 mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka 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 Menambah izin untuk pengguna dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI 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": "*"
        }
    ]
}
```

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

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan 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 memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

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.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

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 memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

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 pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi 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 menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.

- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Izin dan kebijakan IAM

Akses ke Layanan Terkelola Amazon untuk tindakan dan data Prometheus memerlukan kredensi. Kredensional 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

Untuk melihat daftar kemungkinan Layanan Terkelola Amazon untuk tindakan Prometheus. jenis sumber daya, dan kunci kondisi, lihat Tindakan, sumber daya, dan kunci kondisi untuk Layanan Terkelola Amazon untuk Prometheus.

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.

JSON

```
{
    "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 tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat 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, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan

praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.

- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan dalam Layanan Terkelola Amazon untuk Prometheus

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

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 AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik 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 mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal 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 AWSService RoleForAmazonPrometheusScraper 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 terkait layanan, silakan lihat layanan AWS yang bisa digunakan dengan IAM dan carilah layanan yang memiliki opsi Ya di kolom Peran terkait layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait 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 AWSService RoleForAmazonPrometheusScraper terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `scraper.aps.amazonaws.com`

Kebijakan izin peran bernama AmazonPrometheusScraperServiceRolePolicy memungkinkan 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, lihat Izin peran terkait layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Anda tidak perlu membuat peran terkait 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 terkait 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. AWSService RoleForAmazonPrometheusScraper Setelah membuat peran terkait 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, lihat Mengedit peran terkait layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Amazon Managed Service untuk Prometheus

Anda tidak perlu menghapus AWSService RoleForAmazonPrometheusScraper peran secara manual. Saat Anda menghapus semua instance kolektor terkelola yang terkait dengan peran di AWS Management Console, API AWS 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. Lihat informasi yang lebih lengkap di Wilayah yang Didukung.

Logging Amazon Managed Service untuk panggilan API Prometheus menggunakan AWS CloudTrail

Amazon Managed Service untuk Prometheus terintegrasi AWS CloudTraildengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau. Layanan AWS CloudTrailmenangkap 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. 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, kapan dibuat, dan detail tambahan.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut hal ini:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau pengguna.
- Apakah permintaan dibuat atas nama pengguna IAM Identity Center.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan tersebut dibuat oleh Layanan AWS lain.

CloudTrail aktif di Anda Akun AWS ketika Anda membuat akun dan Anda secara otomatis memiliki akses ke riwayat CloudTrail Acara. Riwayat CloudTrail Acara menyediakan catatan yang dapat dilihat, dapat dicari, dapat diunduh, dan tidak dapat diubah dari 90 hari terakhir dari peristiwa manajemen yang direkam dalam file. Wilayah AWS Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna. Tidak ada CloudTrail biaya untuk melihat riwayat Acara.

Untuk catatan acara yang sedang berlangsung dalam 90 hari Akun AWS terakhir Anda, buat jejak atau penyimpanan data acara CloudTrail Danau.

CloudTrail jalan setapak

> Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Semua jalur yang dibuat menggunakan AWS Management Console Multi-region. Anda dapat membuat jalur Single-region atau Multi-region dengan menggunakan. AWS CLI Membuat jejak Multi-wilayah disarankan karena Anda menangkap aktivitas Wilayah AWS di semua akun Anda. Jika

Anda membuat jejak wilayah Tunggal, Anda hanya dapat melihat peristiwa yang dicatat di jejak. Wilayah AWS Untuk informasi selengkapnya tentang jejak, lihat Membuat jejak untuk Anda Akun AWS dan Membuat jejak untuk organisasi di Panduan AWS CloudTrail Pengguna.

Anda dapat mengirimkan satu salinan acara manajemen yang sedang berlangsung ke bucket Amazon S3 Anda tanpa biaya CloudTrail dengan membuat jejak, namun, ada biaya penyimpanan Amazon S3. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga. Untuk informasi tentang harga Amazon S3, lihat Harga Amazon S3.

CloudTrail Penyimpanan data acara danau

CloudTrail Lake memungkinkan Anda menjalankan kueri berbasis SQL pada acara Anda. CloudTrail Lake mengonversi peristiwa yang ada dalam format JSON berbasis baris ke format Apache ORC. ORC adalah format penyimpanan kolumnar yang dioptimalkan untuk pengambilan data dengan cepat. Peristiwa digabungkan ke dalam penyimpanan data peristiwa, yang merupakan kumpulan peristiwa yang tidak dapat diubah berdasarkan kriteria yang Anda pilih dengan menerapkan pemilih acara tingkat lanjut. Penyeleksi yang Anda terapkan ke penyimpanan data acara mengontrol peristiwa mana yang bertahan dan tersedia untuk Anda kueri. Untuk informasi lebih lanjut tentang CloudTrail Danau, lihat Bekerja dengan AWS CloudTrail Danau di Panduan AWS CloudTrail Pengguna.

CloudTrail Penyimpanan data acara danau dan kueri menimbulkan biaya. Saat Anda membuat penyimpanan data acara, Anda memilih opsi harga yang ingin Anda gunakan untuk penyimpanan data acara. Opsi penetapan harga menentukan biaya untuk menelan dan menyimpan peristiwa, dan periode retensi default dan maksimum untuk penyimpanan data acara. Untuk informasi selengkapnya tentang CloudTrail harga, lihat AWS CloudTrail Harga.

Layanan Terkelola Amazon untuk acara manajemen Prometheus di CloudTrail

Acara manajemen memberikan informasi tentang operasi manajemen yang dilakukan pada sumber daya di Anda Akun AWS. Ini juga dikenal sebagai operasi pesawat kontrol. Secara default, CloudTrail mencatat peristiwa manajemen.

Layanan Terkelola Amazon untuk Prometheus mencatat semua Layanan Terkelola Amazon untuk operasi pesawat kontrol Prometheus sebagai peristiwa manajemen. Untuk daftar operasi bidang kontrol Layanan Terkelola Amazon untuk Prometheus yang digunakan oleh Layanan Terkelola Amazon untuk Prometheus, lihat Layanan Terkelola Amazon untuk Referensi API Prometheus. CloudTrail

Layanan Terkelola Amazon untuk contoh acara Prometheus

Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang operasi API yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga peristiwa 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"
}
```


Untuk informasi tentang konten CloudTrail rekaman, lihat konten CloudTrail rekaman di Panduan AWS CloudTrail Pengguna.

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 memberikan AWS izin ke container di pod mana pun 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 di jaringan virtual yang Anda tentukan. 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 oleh AWS PrivateLink, sebuah AWS teknologi yang memungkinkan komunikasi pribadi antara AWS layanan menggunakan antarmuka jaringan elastis dengan alamat IP pribadi. Untuk informasi selengkapnya, lihat posting blog New — AWS PrivateLink for AWS Services.

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 APIs Prometheus kompatibel. Untuk informasi selengkapnya, lihat Kompatibel dengan Prometheus di APIs 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 APIs 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 untuk AWS Security Token Service, untuk memungkinkan sigv4 bekerja melalui titik akhir. Untuk informasi tentang membuat titik akhir VPC AWS STS, lihat Menggunakan titik akhir AWS STS VPC 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.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonManagedPrometheusPermissions",
            "Effect": "Allow",
            "Action": [
                "aps:DescribeWorkspace",
                "aps:DescribeRuleGroupsNamespace",
                "aps:ListRuleGroupsNamespaces",
                "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 Amazon Managed Service untuk kesalahan Prometheus

Gunakan bagian berikut untuk membantu memecahkan masalah dengan Amazon Managed Service for Prometheus.

Topik

429 atau batas melebihi 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
```

Jika Anda melihat kesalahan 429 yang mirip dengan contoh berikut, permintaan Anda telah melampaui kuota Layanan Terkelola Amazon untuk Prometheus untuk tarif (transaksi per detik) yang dapat Anda kirim data ke ruang kerja menggunakan API yang kompatibel dengan Prometheus. `RemoteWrite`

```
ts=2024-03-26T16:50:21.780708811Z caller=dedupe.go:112 component=remote level=error
 remote_name=ab123c
url=https://aps-workspaces.us-east-1.amazonaws.com/workspaces/workspace_id/api/v1/
remote_write
msg="non-recoverable error" count=1000 exemplarCount=0 err="server returned HTTP status
 429 Too Many Requests: {\"message\":\"Rate exceeded\"}"
```

Jika Anda melihat kesalahan 400 yang mirip dengan contoh berikut, permintaan Anda telah melebihi kuota Layanan Terkelola Amazon untuk Prometheus untuk rangkaian waktu aktif. Untuk detail tentang bagaimana kuota deret waktu aktif ditangani, lihat. Kuota default seri aktif

```
ts=2024-03-26T16:50:21.780708811Z caller=push.go:53 level=warn
url=https://aps-workspaces.us-east-1.amazonaws.com/workspaces/workspace_id/api/v1/
remote_write
msg="non-recoverable error" count=500 exemplarCount=0
err="server returned HTTP status 400 Bad Request: maxFailure (quorum) on a given error
 family, rpc error: code = Code(400)
desc = addr=10.1.41.23:9095 state=ACTIVE zone=us-east-1a, rpc error: code = Code(400)
desc = user=accountid_workspace_id: per-user series limit of 10000000 exceeded,
Capacity from 2,000,000 to 10,000,000 is automatically adjusted based on the last 30
 min of usage.
If throttled above 10,000,000 or in case of incoming surges, please contact
 administrator to raise it.
(local limit: 0 global limit: 10000000 actual local limit: 92879)"
```

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 cap 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 memeriksa penyiapan Anda, lihat posting blog Memahami Sampel Duplikat dan Kesalahan Out-of-order Timestamp 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 *<value>* terlampaui, silakan hubungi administrator untuk menaikkannya
- batas seri per metrik *<value>* terlampaui, silakan hubungi administrator untuk menaikkannya
- 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_tsdb_head_samples_ap`	

Jenis data	Kueri untuk digunakan	
	`pended_total[5m])`	
Most-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 Menggunakan CloudWatch metrik untuk memantau Layanan Terkelola Amazon untuk sumber daya Prometheus.

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

Alasan	Arti
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

Menandai Layanan Terkelola Amazon untuk Prometheus

Tag adalah label atribut kustom yang Anda atau AWS tetapkan ke AWS sumber daya. Setiap AWS tag memiliki dua bagian:

- 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 sebanyak 50 tag yang ditetapkan untuk setiap ruang kerja.

Tag membantu Anda mengidentifikasi dan mengatur AWS sumber daya Anda. Banyak AWS layanan mendukung penandaan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya terkait. Misalnya, Anda dapat menetapkan tag yang sama ke Layanan Terkelola Amazon untuk ruang kerja Prometheus yang ditetapkan 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 APIs, atau SDKs untuk menambah, mengelola, dan menghapus tag untuk sumber daya ini. Selain mengidentifikasi, mengatur, dan melacak ruang nama ruang kerja dan grup aturan dengan tag, Anda dapat menggunakan tag dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan sumber daya Amazon Managed Service for Prometheus Anda.

Batasan tag

Batasan dasar berikut berlaku untuk tanda:

- Setiap sumber daya dapat memiliki maksimum 50 tag.
- Untuk setiap sumber daya, setiap kunci tag harus unik, dan setiap kunci tag hanya dapat memiliki satu nilai.
- Panjang kunci tag maksimum adalah 128 karakter Unicode dalam UTF-8.
- Panjang nilai tag maksimum adalah 256 karakter Unicode dalam UTF-8.

- Jika skema penandaan Anda digunakan di beberapa AWS layanan dan sumber daya, ingatlah bahwa layanan lain mungkin memiliki batasan pada karakter yang diizinkan. Karakter yang diizinkan secara umum adalah huruf, angka, spasi yang dapat direpresentasikan dalam UTF-8, dan karakter berikut: .: + = @ _/- (tanda hubung).
- Kunci dan nilai tag peka huruf besar dan kecil. Sebagai praktik terbaik, tentukan strategi untuk menulis tag dalam huruf kapital 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 hanya disediakan untuk AWS digunakan. Anda tidak dapat menyunting atau menghapus kunci atau nilai tag dengan awalan ini. Tag dengan awalan ini tidak dihitung terhadap tags-per-resource batas Anda.

Topik

Menandai Layanan Terkelola Amazon untuk ruang kerja Prometheus

Tag adalah label khusus yang dapat ditetapkan ke sumber daya. Mereka termasuk kunci unik dan nilai opsional (dalam pasangan kunci-nilai). Tag membantu Anda mengidentifikasi dan mengatur 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,, atau SDKs untuk menambah APIs, mengelola, dan menghapus tag untuk sumber daya ini. Selain mengidentifikasi, mengatur, dan melacak ruang kerja Anda dengan tag, Anda dapat menggunakan tag dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan sumber daya Amazon Managed Service for Prometheus Anda.

Gunakan prosedur di bagian ini untuk bekerja dengan tag untuk Amazon Managed Service untuk ruang kerja Prometheus.

Topik

- Lihat tag untuk ruang kerja
- Mengedit tag untuk ruang kerja
- Menghapus tag dari ruang kerja

Menambahkan tag ke ruang kerja

Menambahkan tag ke Layanan Terkelola Amazon untuk ruang kerja Prometheus dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya serta mengelola akses ke sana. Pertama, Anda menambahkan satu atau beberapa tag (pasangan nilai kunci) ke ruang kerja. Setelah Anda memiliki tag, Anda dapat membuat kebijakan IAM untuk mengelola akses ke ruang kerja berdasarkan tag ini. Anda dapat menggunakan konsol atau menambahkan tag AWS CLI ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.

> **Important**
>
> Menambahkan tag ke ruang kerja dapat memengaruhi akses ke ruang kerja tersebut. Sebelum menambahkan tag ke ruang kerja, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya.

Untuk informasi selengkapnya tentang menambahkan tag ke Layanan Terkelola Amazon untuk ruang kerja Prometheus saat Anda membuatnya, lihat. Buat Layanan Terkelola Amazon untuk ruang kerja Prometheus

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 tag ke Layanan Terkelola Amazon untuk ruang kerja Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.

3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Tanda.
6. Jika tidak ada tag yang ditambahkan ke Amazon Managed Service untuk ruang kerja Prometheus, pilih Buat tag. Jika tidak, pilih Kelola 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 Simpan perubahan.

Tambahkan tag ke ruang kerja ()AWS CLI

Ikuti langkah-langkah berikut untuk menggunakan untuk menambahkan tag AWS CLI ke Layanan Terkelola Amazon untuk ruang kerja Prometheus. Untuk menambahkan tag ke ruang kerja saat Anda membuatnya, lihatBuat Layanan Terkelola Amazon untuk ruang kerja Prometheus.

Dalam langkah-langkah ini, kami berasumsi bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi saat ini. Untuk informasi selengkapnya, silakan lihat Menginstal AWS Command Line Interface.

Di terminal atau baris perintah, jalankan tag-resource perintah, tentukan Nama Sumber Daya Amazon (ARN) ruang kerja tempat Anda ingin menambahkan tag dan kunci serta nilai tag yang ingin Anda tambahkan. Anda dapat menambahkan lebih dari satu tag ke ruang kerja. Misalnya, untuk menandai Layanan Terkelola Amazon untuk ruang kerja Prometheus bernama My-Workspace dengan dua tag, kunci tag yang diberi nama dengan nilai tag, dan kunci tag *Status* bernama dengan *Secret* nilai tag: *Team My-Team*

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspaces/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 AWS sumber daya Anda dan mengelola akses ke sana. Untuk informasi selengkapnya tentang strategi penandaan, lihat AWS Menandai Sumber Daya.

Lihat tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus (konsol)

Anda dapat menggunakan konsol untuk melihat tag yang terkait dengan Layanan Terkelola Amazon untuk ruang kerja Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Tanda.

Lihat tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus ()AWS CLI

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk melihat AWS tag 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 tag dan nilai tag 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 tag 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 tag untuk ruang kerja, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag 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 tag yang terkait dengan Layanan Terkelola Amazon untuk ruang kerja Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Tanda.
6. Jika tidak ada tag yang ditambahkan ke ruang kerja, pilih Buat tag. Jika tidak, pilih Kelola 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 Simpan perubahan.

Edit tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus ()AWS CLI

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk memperbarui tag untuk ruang kerja. Anda dapat mengubah nilai untuk kunci yang ada, atau menambahkan kunci lain.

Di terminal atau baris perintah, jalankan tag-resource perintah, tentukan Nama Sumber Daya Amazon (ARN) dari Amazon Managed Service untuk ruang kerja Prometheus tempat Anda ingin memperbarui tag dan menentukan kunci tag dan nilai tag:

```
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 tag yang terkait dengan ruang kerja. Menghapus tag tidak menghapus tag dari AWS sumber daya lain yang terkait dengan tag tersebut.

> **Important**
>
> Menghapus tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus dapat memengaruhi akses ke ruang kerja tersebut. Sebelum menghapus tag dari ruang kerja, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya seperti repositori.

Menghapus tag dari Layanan Terkelola Amazon untuk ruang kerja Prometheus (konsol)

Anda dapat menggunakan konsol untuk menghapus asosiasi antara tag dan ruang kerja.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Tanda.
6. Pilih Kelola tanda.
7. Temukan tag yang ingin Anda hapus, dan pilih Hapus.

Menghapus tag dari Layanan Terkelola Amazon untuk ruang kerja Prometheus ()AWS CLI

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk menghapus tag dari ruang kerja. Menghapus tag tidak menghapusnya, tetapi hanya menghapus hubungan antara tag dan ruang kerja.

> **Note**
>
> Jika Anda menghapus Layanan Terkelola Amazon untuk ruang kerja Prometheus, semua asosiasi tag akan dihapus dari ruang kerja yang dihapus. Anda tidak perlu menghapus tag sebelum menghapus ruang kerja.

Di terminal atau baris perintah, jalankan untag-resource perintah, tentukan Nama Sumber Daya Amazon (ARN) ruang kerja tempat Anda ingin menghapus tag dan kunci tag tag yang ingin Anda hapus. Misalnya, untuk menghapus tag pada ruang kerja bernama My-Workspace dengan kunci 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, jalankan list-tags-for-resource perintah.

Menandai ruang nama grup aturan

Tag adalah label khusus yang dapat ditetapkan ke sumber daya. Mereka termasuk kunci unik dan nilai opsional (dalam pasangan kunci-nilai). Tag membantu Anda mengidentifikasi dan mengatur sumber daya AWS . Di Amazon Managed Service untuk Prometheus, ruang nama grup aturan (dan ruang kerja) dapat diberi tag. Anda dapat menggunakan konsol, AWS CLI,, atau SDKs untuk menambah APIs, mengelola, dan menghapus tag untuk sumber daya ini. Selain mengidentifikasi, mengatur, dan melacak ruang nama grup aturan dengan tag, Anda dapat menggunakan tag dalam kebijakan IAM untuk membantu mengontrol siapa yang dapat melihat dan berinteraksi dengan sumber daya Layanan Terkelola Amazon untuk Prometheus.

Gunakan prosedur di bagian ini untuk bekerja dengan tag untuk Amazon Managed Service untuk ruang nama grup aturan Prometheus.

Topik

Menambahkan tag ke namespace grup aturan

Menambahkan tag ke ruang nama grup aturan Amazon Managed Service untuk Prometheus dapat membantu Anda mengidentifikasi dan mengatur AWS sumber daya serta mengelola akses ke sana. Pertama, Anda menambahkan satu atau beberapa tag (pasangan nilai kunci) ke namespace grup aturan. Setelah Anda memiliki tag, Anda dapat membuat kebijakan IAM untuk mengelola akses ke namespace berdasarkan tag ini. Anda dapat menggunakan konsol atau AWS CLI untuk menambahkan tag ke namespace grup aturan Amazon Managed Service untuk Prometheus.

> **Important**
>
> Menambahkan tag ke namespace grup aturan dapat memengaruhi akses ke namespace grup aturan tersebut. Sebelum menambahkan tag, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan tag untuk mengontrol akses ke sumber daya.

Untuk informasi selengkapnya tentang menambahkan tag ke namespace grup aturan saat Anda membuatnya, lihat. Buat file aturan

Topik

Tambahkan tag ke namespace grup aturan (konsol)

Anda dapat menggunakan konsol untuk menambahkan satu atau beberapa tag ke namespace grup aturan Amazon Managed Service untuk Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/

2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Manajemen Aturan.
6. Pilih tombol di sebelah nama namespace dan pilih Edit.
7. Pilih Buat 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 Simpan perubahan.

Tambahkan tag ke namespace grup aturan ()AWS CLI

Ikuti langkah-langkah berikut untuk menggunakan untuk menambahkan tag AWS CLI ke namespace grup aturan Amazon Managed Service untuk Prometheus. Untuk menambahkan tag ke namespace grup aturan saat Anda membuatnya, lihat. Unggah file konfigurasi aturan ke Amazon Managed Service untuk Prometheus

Dalam langkah-langkah ini, kami berasumsi bahwa Anda telah menginstal versi terbaru dari AWS CLI atau diperbarui ke versi saat ini. Untuk informasi selengkapnya, silakan lihat Menginstal AWS Command Line Interface.

Di terminal atau baris perintah, jalankan tag-resource perintah, tentukan Nama Sumber Daya Amazon (ARN) dari namespace grup aturan tempat Anda ingin menambahkan tag dan kunci serta nilai tag yang ingin Anda tambahkan. Anda dapat menambahkan lebih dari satu tag ke namespace grup aturan. Misalnya, untuk menandai Layanan Terkelola Amazon untuk namespace Prometheus bernama My-Workspace dengan dua tag, kunci tag yang diberi nama dengan nilai tag, dan kunci tag *`Status`* bernama dengan *`Secret`* nilai tag: *`Team My-Team`*

```
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 AWS sumber daya Anda dan mengelola akses ke sana. Untuk informasi selengkapnya tentang strategi penandaan, lihat AWS Menandai Sumber Daya.

Melihat tag untuk namespace grup aturan (konsol) Layanan Terkelola Amazon untuk Prometheus

Anda dapat menggunakan konsol untuk melihat tag yang terkait dengan namespace grup aturan Amazon Managed Service untuk Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Manajemen Aturan.
6. Pilih nama namespace.

Lihat tag untuk Layanan Terkelola Amazon untuk ruang kerja Prometheus ()AWS CLI

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk melihat AWS tag 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 tag dan nilai tag untuk namespace grup aturan:

```
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 grup aturan

Anda dapat mengubah nilai untuk tag yang terkait dengan namespace grup aturan. 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 tag untuk sumber daya, pastikan untuk meninjau kebijakan IAM apa pun yang mungkin menggunakan kunci atau nilai tag untuk mengontrol akses ke sumber daya.

Mengedit tag untuk Layanan Terkelola Amazon untuk namespace grup aturan Prometheus (konsol)

Anda dapat menggunakan konsol untuk mengedit tag yang terkait dengan namespace grup aturan Amazon Managed Service untuk Prometheus.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Manajemen Aturan.
6. Pilih nama namespace.
7. Pilih Kelola tag, Tambahkan tag baru.
8. Untuk mengubah nilai tag yang ada, masukkan nilai baru untuk Nilai.
9. o tambahkan tag tambahan, pilih Tambahkan tag baru.
10. Setelah selesai menambahkan dan mengedit tag, pilih Simpan perubahan.

Mengedit tag untuk Layanan Terkelola Amazon untuk namespace grup aturan Prometheus ()AWS CLI

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk memperbarui tag untuk namespace grup aturan. Anda dapat mengubah nilai untuk kunci yang ada, atau menambahkan kunci lain.

Di terminal atau baris perintah, jalankan tag-resource perintah, tentukan Nama Sumber Daya Amazon (ARN) dari sumber daya tempat Anda ingin memperbarui tag dan tentukan kunci tag dan nilai tag:

```
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 grup aturan

Anda dapat menghapus satu atau beberapa tag yang terkait dengan namespace grup aturan. Menghapus tag tidak menghapus tag dari AWS sumber daya lain yang terkait dengan tag tersebut.

> **Important**
>
> Menghapus tag untuk sumber daya dapat memengaruhi akses ke sumber daya tersebut. Sebelum Anda menghapus tag dari sumber daya, pastikan untuk meninjau kebijakan IAM yang mungkin menggunakan kunci atau nilai untuk tag 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 asosiasi antara tag dan namespace grup aturan.

1. Buka Layanan Terkelola Amazon untuk konsol Prometheus di. https://console.aws.amazon.com/prometheus/
2. Di panel navigasi, pilih ikon menu.
3. Pilih Semua ruang kerja.
4. Pilih ID ruang kerja ruang kerja yang ingin Anda kelola.
5. Pilih tab Manajemen Aturan.

6. Pilih nama namespace.
7. Pilih Kelola tanda.
8. Di samping tag yang ingin Anda hapus, pilih Hapus.
9. Setelah selesai, pilih Simpan perubahan.

Menghapus tag dari Layanan Terkelola Amazon untuk namespace grup aturan Prometheus ()AWS CLI

Ikuti langkah-langkah ini untuk menggunakan AWS CLI untuk menghapus tag dari namespace grup aturan. Menghapus tag tidak menghapusnya, tetapi hanya menghapus hubungan antara tag dan namespace grup aturan.

> **Note**
>
> Jika Anda menghapus namespace grup aturan Amazon Managed Service untuk Prometheus, semua asosiasi tag akan dihapus dari nnamespace yang dihapus. Anda tidak perlu menghapus tag sebelum menghapus namespace.

Di terminal atau baris perintah, jalankan untag-resource perintah, tentukan Nama Sumber Daya Amazon (ARN) dari namespace grup aturan tempat Anda ingin menghapus tag dan kunci tag tag yang ingin Anda hapus. Misalnya, untuk menghapus tag pada ruang kerja bernama My-Workspace dengan kunci 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 . Dengan menggunakan fitur alarm metrik CloudWatch penggunaan Amazon, 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 Kuota default seri aktif. 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 mengetahui informasi selengkapnya, lihat Pelambatan konsumsi.

> **Note**
>
> Kecuali dinyatakan lain, kuota ini per ruang kerja. Nilai maksimum untuk seri aktif per ruang kerja adalah satu miliar.

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. Catatan: Jika batas tercapai, sampel metrik dicatat, tetapi metadata di atas batas dijatuhkan.
Seri aktif per ruang kerja	Setiap Wilayah yang didukung: 50.000.000	Ya	Jumlah seri aktif unik per ruang kerja (hingga maksimal 1 miliar). Serangkaian aktif jika sampel telah dilaporkan dalam 2 jam terakhir. Kapasitas dari 2 M hingga 50 M 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	Tidak	Ukuran maksimum file definisi manajer peringatan, dalam megabyte.

Nama	Default	Dapat disesuaikan	Deskripsi
Ukuran payload peringatan di Alert Manager	Setiap Wilayah yang didukung: 20	Tidak	Ukuran payload peringatan maksimum dari semua peringatan Alert Manager per ruang kerja, dalam megabyte. 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.
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.
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	Tidak	Ukuran gabungan maksimum dari semua label dan nilai label diterima untuk seri, dalam kilobyte.

Nama	Default	Dapat disesuaikan	Deskripsi
LabelSet batas per ruang kerja	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum batas labelset yang dapat dibuat per ruang kerja.
Label per seri metrik	Setiap Wilayah yang didukung: 150	Ya	Jumlah label per seri metrik.
Panjang metadata	Setiap Wilayah yang didukung: 1	Tidak	Panjang maksimum yang diterima untuk metadata metrik, dalam kilobyte. Metadata mengacu pada Nama Metrik, Jenis, Unit dan Teks Bantuan.
Metadata per metrik	Setiap Wilayah yang didukung: 10	Tidak	Jumlah metadata per metrik. Catatan: Jika batas tercapai, sampel metrik dicatat, tetapi metadata di atas batas dijatuhkan.
Node di pohon perutean manajer peringatan	Setiap Wilayah yang didukung: 100	Ya	Jumlah maksimum node di pohon routing manajer peringatan.

Nama	Default	Dapat disesuaikan	Deskripsi
Jumlah operasi API per wilayah dalam transaksi per detik	Setiap Wilayah yang didukung: 10	Ya	Jumlah maksimum operasi API per detik per wilayah untuk semua Layanan Terkelola Amazon untuk APIs Prometheus, termasuk CRUD ruang kerja, APIs penandaan, namespace grup aturan APIs CRUD, dan definisi manajer peringatan CRUD. APIs APIs
Jumlah GetSeries, GetLabels dan operasi GetMetricMetadata API per ruang kerja dalam transaksi per detik	Setiap Wilayah yang didukung: 10	Tidak	Jumlah maksimum GetSeries, GetLabels dan operasi API yang GetMetricMetadata kompatibel dengan Prometheus per detik per ruang kerja.
Jumlah operasi QueryMetrics API per ruang kerja dalam transaksi per detik	Setiap Wilayah yang didukung: 300	Tidak	Jumlah maksimum operasi API yang QueryMetrics kompatibel dengan Prometheus per detik per ruang kerja.
Jumlah operasi RemoteWrite API per ruang kerja dalam transaksi per detik	Setiap Wilayah yang didukung: 3.000	Tidak	Jumlah maksimum operasi API yang RemoteWrite kompatibel dengan Prometheus per detik per ruang kerja.

Nama	Default	Dapat disesuaikan	Deskripsi
Jumlah operasi API lain yang kompatibel dengan Prometheus per ruang kerja dalam transaksi per detik	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum operasi API per detik per ruang kerja untuk semua Prometheus lain yang kompatibel APIs termasuk,, dll. ListAlerts ListRules
Byte kueri untuk kueri instan	Setiap Wilayah yang didukung: 5	Tidak	Byte maksimum yang dapat dipindai oleh satu kueri instan, dalam gigabyte.
Byte kueri untuk kueri rentang	Setiap Wilayah yang didukung: 5	Tidak	Byte maksimum yang dapat dipindai per interval 24 jam dalam kueri rentang tunggal, dalam gigabyte.
Sampel kueri	Setiap Wilayah yang didukung: 50.000.000	Tidak	Jumlah maksimum sampel yang dapat dipindai selama satu kueri.
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: 95	Tidak	Rentang waktu maksimum QueryMetrics, GetSeries, dan GetLabels APIs.

Nama	Default	Dapat disesuaikan	Deskripsi
Ukuran permintaan	Setiap Wilayah yang didukung: 1	Tidak	Ukuran permintaan maksimum untuk konsumsi atau kueri, dalam megabyte.
Interval evaluasi aturan	Setiap Wilayah yang didukung: 30	Ya	Interval evaluasi aturan minimum dari kelompok aturan per ruang kerja, dalam hitungan detik.
Ukuran file definisi namespace grup aturan	Setiap Wilayah yang didukung: 1	Tidak	Ukuran maksimum file definisi namespace grup aturan, dalam megabyte.
Aturan per ruang kerja	Setiap Wilayah yang didukung: 2.000	Ya	Jumlah maksimum aturan per ruang kerja.
Keheningan per ruang kerja	Setiap Wilayah yang didukung: 1.000	Ya	Jumlah maksimum keheningan, termasuk keheningan yang kedaluwarsa, aktif, dan tertunda, per ruang kerja.
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.

Kuota default seri aktif

Layanan Terkelola Amazon untuk ruang kerja Prometheus secara otomatis beradaptasi dengan penggunaan konsumsi Anda. Ketika penggunaan Anda meningkat, layanan secara otomatis meningkatkan kapasitas rangkaian waktu Anda hingga kuota default.

Layanan Terkelola Amazon untuk ruang kerja Prometheus Anda menskalakan secara otomatis, berdasarkan penggunaan Anda, dengan dua cara:

1. Ketika penggunaan rata-rata 30 menit Anda di bawah 5 juta seri, kapasitasnya berlipat ganda (misalnya, ruang kerja dengan penggunaan 3,5 juta mendapat kapasitas 7M).
2. Ketika penggunaan melebihi 5 juta seri, ruang kerja menambahkan 10 juta buffer (misalnya, ruang kerja dengan penggunaan 25 juta mendapat kapasitas 35 juta).

Layanan Terkelola Amazon untuk Prometheus secara otomatis mengalokasikan lebih banyak kapasitas saat konsumsi Anda meningkat, hingga kuota Anda. Ini membantu memastikan beban kerja Anda tidak mengalami pelambatan yang berkelanjutan. Namun, pelambatan dapat terjadi jika Anda menggandakan atau melebihi 10 juta di atas baseline sebelumnya yang dihitung selama 30 menit terakhir. Untuk menghindari pembatasan, Amazon Managed Service for Prometheus merekomendasikan peningkatan konsumsi secara bertahap saat meningkat melampaui baseline Anda sebelumnya.

> **Note**
>
> Kapasitas minimum untuk deret waktu aktif adalah 2 juta, dan tidak ada pelambatan ketika Anda memiliki kurang dari 2 juta seri.
> Untuk melampaui kuota default Anda, Anda dapat meminta peningkatan kuota.

Penskalaan di atas kuota default

Saat Anda meminta peningkatan kuota di atas kuota seri aktif default, Amazon Managed Service for Prometheus menyesuaikan kapasitas ruang kerja Anda. Jika Anda tidak sepenuhnya memanfaatkan peningkatan kapasitas, layanan akan merebut kembali bagian yang tidak terpakai dari waktu ke waktu. Seiring bertambahnya penggunaan Anda, ruang kerja akan meningkat lagi secara otomatis.

Namun, pelambatan dapat terjadi jika Anda lebih dari dua kali lipat atau melebihi 50 juta deret waktu aktif di atas baseline sebelumnya yang dihitung dari 2 jam terakhir. Misalnya:

- Jika kuota Anda 100 juta dan baseline Anda 30 juta, Anda dapat menskalakan hingga 60 juta dalam waktu 2 jam tanpa pembatasan.
- Jika kuota Anda 100 juta dan baseline Anda 50 juta, Anda dapat menskalakan hingga 100 juta penuh dalam waktu 2 jam tanpa pembatasan.

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 CloudWatch untuk memantau konsumsi Anda, dan untuk membuat alarm untuk memperingatkan Anda ketika Anda hampir mencapai batas pelambatan. Untuk informasi selengkapnya, lihat Menggunakan CloudWatch metrik untuk memantau Layanan Terkelola Amazon untuk sumber daya Prometheus.

Amazon Managed Service untuk Prometheus menggunakan algoritma token bucket untuk mengimplementasikan 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 tersebut kemudian diisi ulang oleh 170.000 token setiap detik, hingga mencapai kapasitas maksimumnya 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.

Layanan Terkelola Amazon untuk Referensi API Prometheus

Amazon Managed Service untuk Prometheus menawarkan dua jenis: APIs

1. Layanan Terkelola Amazon untuk APIs Prometheus — APIs Ini memungkinkan Anda membuat dan mengelola Layanan Terkelola Amazon untuk ruang kerja Prometheus, termasuk operasi untuk ruang kerja, pencakar, definisi manajer peringatan, ruang nama grup aturan, dan pencatatan. Anda menggunakan AWS SDKs, tersedia untuk berbagai bahasa pemrograman, untuk berinteraksi dengan ini APIs.
2. Kompatibel dengan Prometheus APIs - Amazon Managed Service untuk Prometheus mendukung HTTP yang kompatibel dengan Prometheus. APIs Ini APIs memungkinkan pembuatan aplikasi khusus, mengotomatiskan alur kerja, mengintegrasikan dengan layanan atau alat lain, dan melakukan kueri serta berinteraksi dengan data pemantauan Anda menggunakan bahasa kueri Prometheus (PromQL).

Bagian ini mencantumkan operasi API dan struktur data yang didukung oleh Amazon Managed Service untuk Prometheus.

Untuk informasi tentang kuota untuk seri, label, dan permintaan API, lihat Layanan Terkelola Amazon untuk kuota layanan Prometheus di Panduan Pengguna Layanan Terkelola Amazon untuk Prometheus.

Topik

- Layanan Dikelola Amazon untuk Prometheus APIs
- Kompatibel dengan Prometheus APIs

Layanan Dikelola Amazon untuk Prometheus APIs

Layanan Terkelola Amazon untuk Prometheus menyediakan operasi API yang membuat dan memelihara Layanan Terkelola Amazon Anda untuk ruang kerja Prometheus. Ini termasuk APIs untuk ruang kerja, pencakar, definisi manajer peringatan, ruang nama grup aturan, dan pencatatan.

Untuk informasi selengkapnya tentang Layanan Terkelola Amazon untuk APIs Prometheus, lihat Referensi API Amazon Managed Service for Prometheus.

Menggunakan Amazon Managed Service untuk Prometheus dengan SDK AWS

AWS kit pengembangan perangkat lunak (SDKs) 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 SDKs dan alat menurut bahasa, lihat Alat untuk Dibangun AWS di Pusat AWS Pengembang.

> **Versi SDK**
>
> Kami menyarankan Anda menggunakan versi AWS SDK terbaru, dan lainnya SDKs, yang Anda gunakan dalam proyek Anda, dan untuk tetap SDKs up to date. AWS SDK memberi Anda fitur dan fungsionalitas terbaru, dan juga pembaruan keamanan.

Kompatibel dengan Prometheus APIs

Amazon Managed Service untuk Prometheus mendukung Prometheus berikut yang kompatibel dengan Prometheus. APIs

Untuk informasi selengkapnya tentang penggunaan Prometheus yang kompatibel dengan Prometheus APIs, lihat. Kueri menggunakan Prometheus-kompatibel APIs

Topik

- CreateAlertManagerAlerts
- DeleteAlertManagerSilence
- GetAlertManagerStatus
- GetAlertManagerSilence
- GetLabels
- GetMetricMetadata
- GetSeries
- ListAlerts
- ListAlertManagerAlerts
- ListAlertManagerAlertGroups

- ListAlertManagerReceivers
- ListAlertManagerSilences
- ListRules
- PutAlertManagerSilences
- QueryMetrics
- RemoteWrite

CreateAlertManagerAlerts

`CreateAlertManagerAlerts`Operasi membuat peringatan di ruang kerja.

Kata kerja HTTP yang valid:

`POST`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

- `/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`

Berlaku URIs:

- `/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

- `/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

`/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`

Berlaku URIs:

- `/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.

`max_samples_processed_warning_threshold=<integer>`(Opsional) Menetapkan ambang peringatan untuk Sampel Kueri yang Diproses (QSP). Saat kueri mencapai ambang batas ini, pesan peringatan akan dikembalikan dalam respons API.

`max_samples_processed_error_threshold=<integer>>`(Opsional) Menetapkan ambang kesalahan untuk Sampel Kueri yang Diproses (QSP). Kueri yang melebihi ambang batas ini akan ditolak dengan kesalahan dan tidak akan dikenakan biaya. Digunakan untuk mencegah biaya permintaan yang berlebihan.

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
- `w`minggu, 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`

Berlaku URIs:

`/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
Menambahkan dukungan kebijakan berbasis sumber daya	Tindakan API berikut sekarang tersedia: • DeleteResourcePolicy • DescribeResourcePolicy • PutResourcePolicy	Agustus 15, 2025
Perbarui ke kebijakan IAM AmazonPrometheusConsoleFullAccess terkelola.	AmazonPrometheusConsoleFullAccessKebijakan telah diperbarui. `aps:DescribeQueryLoggingConfiguration` Izin `aps:CreateQueryLoggingConfiguration` `aps:UpdateQueryLoggingConfiguration` `aps:DeleteQueryLoggingConfiguration` ,, ditambahkan ke kebijakan.	5 Mei 2025
Menambahkan pengeditan file definisi aturan dan file konfigurasi manajer Peringatan di konsol	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk mengedit file konfigurasi manajer Alert dan file definisi aturan dari dalam Amazon	16 Mei 2024

	Managed Service untuk konsol Prometheus.	
Menambahkan penyiapan kolektor AWS terkelola yang lebih sederhana dengan entri akses untuk Amazon EKS	Amazon Managed Service untuk Prometheus menambahkan dukungan untuk entri akses Amazon EKS untuk menyederhanakan pengaturan kolektor terkelola.AWS Kebijakan AmazonPrometheusScraperServiceRolePolicy AWS terkelola untuk pengumpul terkelola diperbarui untuk memungkinkan penghapusan entri akses yang tidak lagi digunakan.	2 Mei 2024
Pindahkan AWS API ke panduan referensi API terpisah	Amazon Managed Service untuk AWS APIs Prometheus sekarang tersedia dalam referensi mereka sendiri, Amazon Managed Service untuk Prometheus API Referensi. Kompatibel dengan Prometheus APIs 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 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 Mengintegrasikan 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 memodifikasi 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

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