aws

사용자 가이드

Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus: 사용자 가이드

Table of Contents

Amazon Managed Service for Prometheus란?

Amazon Managed Service for Prometheus는 컨테이너 지표에 대한 서버리스, Prometheus 호환 모니터링 서비스로 컨테이너 환경을 대규모로 더 쉽고 안전하게 모니터링할 수 있도록 합니다. Amazon Managed Service for Prometheus를 사용하면 컨테이너화된 워크로드의 성능을 모니터링하는 데 현재 사용하는 것과 동일한 오픈 소스 Prometheus 데이터 모델과 쿼리 언어를 사용할 수 있으며, 기본 인프라를 관리할 필요 없이 향상된 확장성, 가용성 및 보안도 누릴 수 있습니다.

Amazon Managed Service for Prometheus는 워크로드 크기가 확장 및 축소됨에 따라 운영 지표의 수집, 저장 및 쿼리를 자동으로 확장합니다. AWS 보안 서비스와 통합되어 데이터에 빠르고 안전하게 액세스할 수 있습니다.

Amazon Managed Service for Prometheus는 다중 가용 영역(다중 AZ) 배포를 사용하여 높은 가용성을 제공하도록 설계되었습니다. 워크스페이스에 수집된 데이터는 같은 리전의 세 가용 영역에 복제됩니다.

Amazon Managed Service for Prometheus는 Amazon Elastic Kubernetes Service 및 자체 관리형 Kubernetes 환경에서 실행되는 컨테이너 클러스터에 작동합니다.

Amazon Managed Service for Prometheus를 사용하면 Prometheus에서 사용하는 것과 동일한 오픈 소스 Prometheus 데이터 모델 및 PromQL 쿼리 언어를 사용할 수 있습니다. 엔지니어링 팀은 ProMQL을 사용하여 지표를 필터링 및 집계하고 경보를 발생하고, 코드 변경 없이 신속하게 성능 가시성을 확보할 수 있습니다. Amazon Managed Service for Prometheus는 운영 비용 및 복잡성 없이 유연한 쿼리 기능을 제공합니다.

워크스페이스에 수집된 지표는 150일 동안 저장되고 그런 후에 자동으로 삭제됩니다.

지원되는 리전

Amazon Managed Service for Prometheus는 현재 다음 리전을 지원합니다.

리전 이름	Region	Endpoint	프로토콜	
US East (Ohio)	us-east-2	aps.us-east-2.amazonaws.com	HTTPS	
		aps-workspaces.us-east-2.amazonaws.com	HTTPS	

리전 이름	Region	Endpoint	프로토콜
미국 동부(버지니아 북부)	us-east-1	aps.us-east-1.amazonaws.com aps-workspaces.us-east-1.amazonaws.com	HTTPS HTTPS
미국 서부(오리건)	us-west-2	aps.us-west-2.amazonaws.com aps-workspaces.us-west-2.amazonaws.com	HTTPS HTTPS
아시아 태평양(뭄바이)	ap-south-1	aps.ap-south-1.amazonaws.com aps-workspaces.ap-south-1.amazonaws.com	HTTPS HTTPS
Asia Pacific (Seoul)	ap-northeast-2	aps.ap-northeast-2.amazonaws.com aps-workspaces.ap-northeast-2.amazonaws.com	HTTPS HTTPS
아시아 태평양(싱가포르)	ap-southeast-1	aps.ap-southeast-1.amazonaws.com aps-workspaces.ap-southeast-1.amazonaws.com	HTTPS HTTPS
아시아 태평양(시드니)	ap-southeast-2	aps.ap-southeast-2.amazonaws.com aps-workspaces.ap-southeast-2.amazonaws.com	HTTPS HTTPS
아시아 태평양(도쿄)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com aps-workspaces.ap-northeast-1.amazonaws.com	HTTPS HTTPS
유럽(프랑크푸르트)	eu-central-1	aps.eu-central-1.amazonaws.com aps-workspaces.eu-central-1.amazonaws.com	HTTPS HTTPS
유럽(아일랜드)	eu-west-1	aps.eu-west-1.amazonaws.com aps-workspaces.eu-west-1.amazonaws.com	HTTPS HTTPS

리전 이름	Region	Endpoint	프로토콜
Europe (London)	eu-west-2	aps.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.amazonaws.com	HTTPS
Europe (Paris)	eu-west-3	aps.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.amazonaws.com	HTTPS
Europe (Stockholm)	eu-north-1	aps.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.amazonaws.com	HTTPS
남아메리카(상파울루)	sa-east-1	aps.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.amazonaws.com	HTTPS

요금

지표 수집 및 보관에 대한 요금이 발생합니다. 보관 요금은 지표 샘플 및 메타데이터의 압축된 크기를 기준으로 합니다. 자세한 내용은 Amazon Managed Service for Prometheus 요금제를 참조하세요.

Cost Explorer와 AWS 비용 및 사용 보고서를 사용하여 요금을 모니터링할 수 있습니다. 자세한 내용은 Cost Explorer를 사용하여 데이터 탐색 및 AWS 비용 및 사용 보고서란 무엇입니까?를 참조하세요.

프리미엄 지원

원하는 수준의 AWS 프리미엄 지원 플랜을 구독하는 경우 Amazon Managed Service for Prometheus에 프리미엄 지원이 적용됩니다.

시작하기

이 섹션에서는 Amazon Managed Service for Prometheus WorkSpace를 생성하고, 해당 WorkSpace에 Prometheus 지표를 수집하도록 설정하고, 해당 지표를 쿼리하는 방법을 설명합니다.

또한 AWS를 처음 사용하는 경우에 대비하여 AWS 계정 설정에 대한 정보도 포함되어 있습니다.

주제

설정

AWS를 처음으로 설정하려면 이 섹션의 태스크를 완료합니다. 이미 AWS 계정이 있으면 WorkSpace 생성 단계로 건너뜁니다.

AWS에 가입하면 AWS 계정에서 Amazon Managed Service for Prometheus를 포함한 AWS의 모든 서비스에 자동으로 액세스할 수 있습니다. 하지만 사용한 서비스에 대해서만 청구됩니다.

주제

AWS 계정에 등록

AWS 계정 항목이 없으면 다음 절차에 따라 생성하십시오.

AWS 계정에 가입하려면

1. https://portal.aws.amazon.com/billing/signup을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   가입 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

AWS 계정에 가입하면 AWS 계정 루트 사용자 항목이 생성됩니다. 루트 사용자에게 계정의 모든 AWS 서비스 및 리소스에 대한 액세스 권한이 있습니다. 보안 모범 사례는 관리 사용자에게 관리자 액세스 권한을 할당하고, 루트 사용자만 루트 사용자 액세스 권한이 필요한 태스크를 수행하는 것입니다.

AWS 항목은 가입 절차 완료된 후 사용자에게 확인 이메일을 전송합니다. 언제든지 https://aws.amazon.com/으로 이동하고 내 계정을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리 사용자 생성

AWS 계정에 가입하고, AWS 계정 루트 사용자에 보안 조치를 한 다음, AWS IAM Identity Center를 활성화하여 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 생성합니다.

귀하의 AWS 계정 루트 사용자 보호

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 AWS Management Console에 계정 소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 사용 설명서의 루트 사용자로 로그인을 참조하십시오.
2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자용 가상 MFA 디바이스 활성화(콘솔) 섹션을 참조하십시오.

관리 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center 설정을 참조하십시오.
2. IAM Identity Center에서 관리 사용자에게 관리 액세스 권한을 부여합니다.

   IAM Identity Center 디렉터리를 ID 소스로 사용하는 방법에 대한 자습서는 AWS IAM Identity Center 사용 설명서의 기본 IAM Identity Center 디렉터리로 사용자 액세스 구성을 참조하십시오.

관리 사용자로 로그인

- IAM 자격 증명 센터 사용자로 로그인하려면 IAM 자격 증명 센터 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

  IAM Identity Center 사용자로 로그인하는 데 도움이 필요한 경우 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하십시오.

WorkSpace 생성

WorkSpace는 Prometheus 지표 보관 및 쿼리를 위한 전용 논리 공간입니다. WorkSpace는 업데이트, 나열, 설명, 삭제, 지표 수집 및 쿼리와 같은 관리 작업을 승인하기 위한 세분화된 액세스 제어를 지원합니다. 계정의 각 리전에는 하나 이상의 WorkSpace가 있을 수 있습니다.

WorkSpace를 설정하려면 다음 단계를 따르십시오.

> **Note**
>
> WorkSpace 생성에 대한 자세한 내용은 WorkSpace 생성 섹션을 참조하십시오.

Amazon Managed Service for Prometheus WorkSpace를 생성하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. WorkSpace 별칭의 경우 새 WorkSpace의 별칭을 입력합니다.

   WorkSpace 별칭은 WorkSpace를 식별하는 데 도움이 되는 친숙한 이름입니다. 별칭은 고유하지 않아도 됩니다. 두 WorkSpace의 별칭이 동일할 수 있지만 모든 WorkSpace에는 Amazon Managed Service for Prometheus에서 생성되는 고유한 WorkSpace ID가 있습니다.
3. (선택 사항) 네임스페이스에 태그를 추가하려면 새 태그 추가를 선택합니다.

   그런 다음, 키에서 태그 이름을 입력합니다. 값에 태그의 선택적 값을 추가할 수 있습니다.

   다른 태그를 추가하려면 새 태그 추가를 다시 선택합니다.
4. WorkSpace 생성을 선택합니다.

   WorkSpace 세부 정보 페이지가 나타납니다. 여기에는 원격 쓰기 및 쿼리에 대한 이 WorkSpace의 상태, ARN, WorkSpace ID 및 엔드포인트 URL을 비롯한 정보가 표시됩니다.

처음에는 상태가 아마도 생성 중이 됩니다. 지표 수집 설정으로 넘어가기 전에 상태가 활성이 될 때까지 기다리십시오.

엔드포인트 - 원격 쓰기 URL 및 엔드포인트 - 쿼리 URL에 표시된 URL을 기록해 두십시오. 이 WorkSpace에 원격으로 지표를 쓰도록 Prometheus 서버를 구성하고 해당 지표를 쿼리할 때 필요합니다.

Prometheus 지표를 WorkSpace에 수집

지표를 수집하는 한 가지 방법은 독립형 Prometheus 에이전트(에이전트 모드에서 실행되는 Prometheus 인스턴스)를 사용하여 클러스터에서 지표를 스크래핑한 후 이를 Amazon Managed Service for Prometheus로 전달하여 저장 및 모니터링하도록 하는 것입니다. 이 섹션에서는 Helm을 사용하여 Prometheus 에이전트의 새 인스턴스를 설정하여 Amazon EKS에서 Amazon Managed Service for Prometheus WorkSpace로 지표를 수집하는 방법을 설명합니다.

지표를 보호하고 고가용성 지표를 생성하는 방법을 포함하여 Amazon Managed Service for Prometheus로 데이터를 수집하는 다른 방법에 대한 자세한 내용은 Prometheus 지표를 WorkSpace에 수집 섹션을 참조하십시오.

> **Note**
>
> WorkSpace에 수집된 지표는 150일 동안 저장되고 그런 후에 자동으로 삭제됩니다.

이 섹션의 지침을 통해 Amazon Managed Service for Prometheus를 빠르게 시작하고 실행할 수 있습니다. Amazon EKS 클러스터에 새 Prometheus 서버를 설정하면 새 서버는 기본 구성을 사용해 에이전트 역할을 하여 Amazon Managed Service for Prometheus로 지표를 전송합니다. 이 방법의 사전 조건은 다음과 같습니다.

- 새 Prometheus 서버가 지표를 수집할 Amazon EKS 클러스터가 있어야 합니다.
- Helm CLI 3.0 이상을 사용해야 합니다.
- 다음 섹션의 단계를 수행하려면 Linux 또는 macOS 컴퓨터를 사용해야 합니다.

1단계: 새 Helm 차트 리포지토리 추가

새 Helm 차트 리포지토리를 추가하려면 다음 명령을 입력합니다. 이러한 명령에 대한 자세한 내용은 Helm 리포지토리를 참조하십시오.

```
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
```

2단계: Prometheus 네임스페이스 생성

다음 명령을 입력하여 Prometheus 서버 및 기타 모니터링 구성 요소에 대한 Prometheus 네임스페이스를 생성합니다. *prometheus-agent-namespace*를 이 네임스페이스에 사용할 이름으로 바꿉니다.

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

3단계: 서비스 계정의 IAM 역할 설정

이 수집 방법에서는 Prometheus 에이전트가 실행되는 Amazon EKS 클러스터의 서비스 계정에 대한 IAM 역할을 사용해야 합니다.

서비스 계정에 대한 IAM 역할을 사용할 경우 IAM 역할을 Kubernetes 서비스 계정에 연결할 수 있습니다. 이렇게 하면 이 서비스 계정에서는 이 서비스 계정을 사용하는 모든 포드에 있는 컨테이너에 AWS 권한을 제공할 수 있습니다. 자세한 내용은 서비스 계정에 대한 IAM 역할을 참조하십시오.

이러한 역할을 아직 설정하지 않은 경우 Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정의 지침에 따라 역할을 설정하십시오. 해당 섹션의 지침에는 `eksctl`을 사용해야 합니다. 자세한 내용은 Amazon Elastic Kubernetes Service 시작 - `eksctl`을 참조하십시오.

> **Note**
>
> EKS 또는 AWS를 사용하지 않으며 액세스 키와 비밀 키만 사용하여 Amazon Managed Service for Prometheus에 액세스하는 경우에는 `EKS-IAM-ROLE` 기반 SigV4를 사용할 수 없습니다.

4단계: 새 서버 설정 및 지표 수집 시작

Amazon Managed Service for Prometheus WorkSpace로 지표를 전송하는 새 Prometheus 에이전트를 설치하려면 다음 단계를 따르십시오.

새 Prometheus 에이전트를 설치하여 Amazon Managed Service for Prometheus WorkSpace로 지표를 보내려면

1. 텍스트 편집기를 사용하여 다음 내용을 포함하는 my_prometheus_values_yaml이라는 파일을 생성합니다.

   - *IAM_PROXY_PROMETHEUS_ROLE_ARN*을 Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정에서 생성한 amp-iamproxy-inges-role의 ARN으로 바꿉니다.
   - *WORKSPACE_ID*를 Amazon Managed Service for Prometheus WorkSpace의 ID로 바꿉니다.
   - *REGION*을 Amazon Managed Service for Prometheus WorkSpace의 리전으로 바꿉니다.

```
## 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. 다음 명령을 입력하여 Prometheus 서버를 생성합니다.

   - *prometheus-chart-name*을 Prometheus 릴리스 이름으로 바꿉니다.
   - *prometheus-agent-namespace*를 Prometheus 네임스페이스의 이름으로 바꿉니다.

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

```
-f my_prometheus_values_yaml
```

Prometheus 지표 쿼리

이제 지표가 WorkSpace에 수집되었으므로 쿼리할 수 있습니다. 지표를 쿼리하는 일반적인 방법은 Grafana와 같은 서비스를 사용하여 지표를 쿼리하는 것입니다. 이 섹션에서는 Amazon Managed Grafana를 사용하여 Amazon Managed Service for Prometheus에서 지표를 쿼리하는 방법을 알아봅니다.

> **Note**
>
> Amazon 관리 서비스에서 Prometheus 지표를 쿼리하거나 Amazon Managed Service for Prometheus API를 사용하는 다른 방법에 대해 알아보려면 Prometheus 지표 쿼리을 참조하십시오.

쿼리는 표준 Prometheus 쿼리 언어인 PromQL을 사용하여 수행합니다. PromQL 및 해당 구문에 대한 자세한 내용은 Prometheus 설명서의 Prometheus 쿼리를 참조하십시오.

Amazon Managed Grafana는 오픈 소스 Grafana용 완전 관리형 서비스로 대규모로 데이터 소스를 시각화하고 분석할 수 있는 오픈 소스, 타사 ISV 및 AWS 서비스에 대한 연결을 간소화합니다.

Amazon Managed Service for Prometheus에서는 Amazon Managed Grafana를 사용하여 WorkSpace에서 지표를 쿼리할 수 있습니다. Amazon Managed Grafana 콘솔에서 기존 Amazon Managed Service for Prometheus 계정을 검색하여 Amazon Managed Service for Prometheus WorkSpace를 데이터 소스로 추가할 수 있습니다. Amazon Managed Grafana는 Amazon Managed Service for Prometheus에 액세스하는 데 필요한 인증 자격 증명의 구성을 관리합니다. Amazon Managed Grafana에서 Amazon Managed Service for Prometheus에 대한 연결을 생성하는 방법에 대한 자세한 지침은 Amazon Managed Grafana 사용 설명서의 지침을 참조하십시오.

Amazon Managed Grafana에서 Amazon Managed Service for Prometheus 알림을 확인할 수도 있습니다. 알림과의 통합을 설정하는 방법에 대한 지침은 Amazon Managed Grafana 또는 오픈 소스 Grafana와 알림 통합 섹션을 참조하십시오.

> **Note**
>
> 프라이빗 VPC를 사용하도록 Amazon Managed Grafana WorkSpace를 구성한 경우, Amazon Managed Service for Prometheus WorkSpace를 동일한 VPC에 연결해야 합니다. 자세한 내용은 프라이빗 VPC에서 Amazon Managed Grafana에 연결 섹션을 참조하십시오.

워크스페이스 관리

WorkSpace는 Prometheus 지표 보관 및 쿼리를 위한 전용 논리 공간입니다. 워크스페이스는 업데이트, 나열, 설명, 삭제, 지표 수집 및 쿼리와 같은 관리 작업을 승인하기 위한 세분화된 액세스 제어를 지원합니다. 계정의 각 리전에는 하나 이상의 WorkSpace가 있을 수 있습니다.

이 섹션의 절차에 따라 Amazon Managed Service for Prometheus WorkSpace를 생성하고 관리합니다.

주제

WorkSpace 생성

Amazon Managed Service for Prometheus 워크스페이스를 생성하려면 다음 단계를 따르세요.

를 사용하여 작업공간을 만들려면 AWS CLI

1. 다음 명령을 입력하여 WorkSpace를 생성합니다. 이 예제에서는 `my-first-workspace`라는 WorkSpace를 생성하지만 원할 경우 다른 별칭을 사용(또는 미사용)할 수 있습니다. 워크스페이스 별칭은 워크스페이스를 식별하는 데 도움이 되는 친숙한 이름입니다. 별칭은 고유하지 않아도 됩니다. 두 WorkSpace의 별칭이 동일할 수 있지만 모든 WorkSpace에는 Amazon Managed Service for Prometheus에서 생성되는 고유한 WorkSpace ID가 있습니다.

   (선택 사항) 자체 KMS 키를 사용하여 작업 공간에 저장된 데이터를 암호화하려면 사용할 AWS KMS 키와 함께 `kmsKeyArn` 파라미터를 포함시킬 수 있습니다. Prometheus용 Amazon Managed Service에서는 고객 관리 키 사용에 대해 비용을 청구하지 않지만, 에서 사용하는 키와 관련된 비용이 발생할 수 있습니다. AWS Key Management Service Amazon Managed Service for Prometheus 데이터 암호화 또는 자체 고객 관리형 키를 생성, 관리 및 사용하는 방법에 대한 자세한 내용은 저장 시 암호화을 참조하세요.

   대괄호(`[]`) 안의 매개 변수는 선택 사항이므로 명령에 대괄호를 포함하지 마세요.

```
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]
```

이 명령은 다음 데이터를 반환합니다.

- `workspaceId`는 이 워크스페이스의 고유 ID입니다. 이 ID를 기록해 둡니다.
- `arn`은 이 워크스페이스의 ARN입니다.
- `status`는 워크스페이스의 현재 상태입니다. WorkSpace를 만든 직후에는 아마도 `CREATING`이 됩니다.
- `kmsKeyArn`은 WorkSpace 데이터를 암호화하는 데 사용되는 고객 관리형 키입니다(지정된 경우).

> **Note**
>
> 고객 관리형 키로 생성한 WorkSpace는 수집에AWS 관리형 수집기를 사용할 수 없습니다.
> 고객 관리 키를 사용할지 아니면 AWS 소유 키를 사용할지 신중하게 선택하십시오. 고객 관리 키로 생성한 작업 영역은 나중에 AWS 소유 키를 사용하도록 전환할 수 없으며, 그 반대의 경우도 마찬가지입니다.

- `tags`는 WorkSpace의 태그(있는 경우)를 나열합니다.

2. `create-workspace` 명령이 `CREATING` 상태를 반환하면 다음 명령을 입력하여 워크스페이스가 준비된 경우를 확인할 수 있습니다. `create-workspace`명령이 반환한 *`my-workspace-id`*값으로 바꾸세요. `workspaceId`

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

`describe-workspace` 명령이 `status`에 대해 `ACTIVE`를 반환하면 워크스페이스를 사용할 준비가 된 것입니다.

Amazon Managed Service for Prometheus 콘솔을 사용하여 워크스페이스를 생성하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 생성을 선택합니다.

3. 워크스페이스 별칭의 경우 새 워크스페이스의 별칭을 입력합니다.

   워크스페이스 별칭은 워크스페이스를 식별하는 데 도움이 되는 친숙한 이름입니다. 별칭은 고유하지 않아도 됩니다. 두 WorkSpace의 별칭이 동일할 수 있지만 모든 WorkSpace에는 Amazon Managed Service for Prometheus에서 생성되는 고유한 WorkSpace ID가 있습니다.

4. (선택 사항) 자체 KMS 키를 사용하여 작업 공간에 저장된 데이터를 암호화하려면 암호화 설정 사용자 지정을 선택하고 사용할 키를 선택 (또는 새 AWS KMS 키를 생성) 할 수 있습니다. 드롭다운 목록에서 계정의 키를 선택하거나 액세스 권한이 있는 모든 키의 ARN을 입력할 수 있습니다. Prometheus용 Amazon Managed Service에서는 고객 관리 키 사용에 대해 비용을 청구하지 않지만, 에서 사용하는 키와 관련된 비용이 발생할 수 있습니다. AWS Key Management Service

   Amazon Managed Service for Prometheus 데이터 암호화에 대한 자세한 내용 또는 고객 관리형 키를 직접 생성, 관리 및 사용하는 방법에 대한 자세한 내용은 저장 시 암호화 섹션을 참조하세요.

   > **Note**
   >
   > 고객 관리형 키로 생성한 WorkSpace는 수집에AWS 관리형 수집기를 사용할 수 없습니다. 고객 관리 키를 사용할지 아니면 AWS 소유 키를 사용할지 신중하게 선택하십시오. 고객 관리 키로 생성한 작업 영역은 나중에 AWS 소유 키를 사용하도록 전환할 수 없으며, 그 반대의 경우도 마찬가지입니다.

5. (선택 사항) WorkSpace에 하나 이상의 태그를 추가하려면 새 태그 추가를 선택합니다. 그런 다음, 키에 태그 이름을 입력합니다. 값에 태그의 선택적 값을 추가할 수 있습니다.

   다른 태그를 추가하려면 새 태그 추가를 다시 선택합니다.

6. 워크스페이스 생성을 선택합니다.

   워크스페이스 세부 정보 페이지가 나타납니다. 여기에는 원격 쓰기 및 쿼리에 대한 이 WorkSpace의 상태, ARN, WorkSpace ID 및 엔드포인트 URL을 비롯한 정보가 표시됩니다.

   WorkSpace가 준비될 때까지 상태가 CREATING으로 돌아갑니다. 지표 수집 설정으로 넘어가기 전에 상태가 활성이 될 때까지 기다리세요.

   엔드포인트 - 원격 쓰기 URL 및 엔드포인트 - 쿼리 URL에 표시된 URL을 기록해 두세요. 이 워크스페이스에 원격으로 지표를 쓰도록 Prometheus 서버를 구성하고 해당 지표를 쿼리할 때 필요합니다.

워크스페이스에 지표를 수집하는 방법에 대한 자세한 내용은 Prometheus 지표를 WorkSpace에 수집 섹션을 참조하세요.

워크스페이스 편집

워크스페이스를 편집하여 별칭을 변경할 수 있습니다. AWS CLI를 사용하여 워크스페이스 별칭을 변경하려면 다음 명령을 입력합니다.

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

Amazon Managed Service for Prometheus 콘솔을 사용하여 워크스페이스를 편집하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 페이지 왼쪽 상단에서 메뉴 아이콘을 선택한 다음, 모든 워크스페이스를 선택합니다.
3. 편집할 워크스페이스의 워크스페이스 ID를 선택한 후 편집을 선택합니다.
4. 워크스페이스의 새 별칭을 입력한 다음, 저장을 선택합니다.

워크스페이스 ARN 찾기

콘솔 또는 AWS CLI를 사용하여 Amazon Managed Service for Prometheus 워크스페이스의 ARN을 찾을 수 있습니다.

Amazon Managed Service for Prometheus 콘솔을 사용하여 워크스페이스 ARN을 찾으려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 페이지 왼쪽 상단에서 메뉴 아이콘을 선택한 다음, 모든 워크스페이스를 선택합니다.
3. 워크스페이스의 워크스페이스 ID를 선택합니다.

   워크스페이스 ARN은 ARN 아래에 표시됩니다.

를 사용하여 작업공간 ARN을 AWS CLI 찾으려면 다음 명령을 입력합니다.

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

결과에서 `arn` 값을 찾습니다.

워크스페이스 삭제

작업 공간을 삭제하면 해당 작업 공간에 인제스트된 데이터가 삭제됩니다.

> **Note**
>
> Amazon Managed Service for Prometheus 작업 영역을 삭제해도 지표를 스크랩하여 작업 공간으로 보내는 관리형 수집기는 자동으로 AWS 삭제되지 않습니다. 자세한 정보는 스크레이퍼 찾기 및 삭제을 참조하세요.

를 사용하여 작업 영역을 삭제하려면 AWS CLI

다음 명령을 사용합니다.

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

Amazon Managed Service for Prometheus 콘솔을 사용하여 워크스페이스를 삭제하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 페이지 왼쪽 상단에서 메뉴 아이콘을 선택한 다음, 모든 워크스페이스를 선택합니다.
3. 삭제할 워크스페이스의 워크스페이스 ID를 선택한 후 삭제를 선택합니다.
4. 확인 상자에 **delete**를 입력한 다음, 삭제를 선택합니다.

Prometheus 지표를 WorkSpace에 수집

이 섹션에서는 지표가 WorkSpace에 수집되도록 설정하는 방법을 설명합니다.

Amazon Managed Service for Prometheus WorkSpace에 지표를 수집하는 방법으로는 두 가지 옵션이 있습니다.

- AWS 관리형 컬렉터 사용 — Prometheus용 Amazon Managed Service는 에이전트가 필요 없는 완전 관리형 스크레이퍼를 제공하여 Amazon Elastic Kubernetes Service (Amazon EKS) 클러스터의 메트릭을 자동으로 스크레이핑합니다. 스크래핑은 Prometheus 호환 엔드포인트에서 지표를 자동으로 가져옵니다.
- 고객 관리형 수집기 사용 - 자체 수집기를 관리할 수 있는 다양한 옵션이 있습니다. 가장 일반적으로 사용되는 두 가지 수집기는 Prometheus 인스턴스를 직접 설치하거나 에이전트 모드에서 실행하거나 Distro for 를 사용하는 것입니다. AWS OpenTelemetry 이러한 수집기는 다음 섹션에 자세히 설명되어 있습니다.

  수집기는 Prometheus 원격 쓰기 기능을 사용하여 Amazon Managed Service for Prometheus에 지표를 전송합니다. 자체 애플리케이션에서 Prometheus 원격 쓰기 기능을 사용하여 Amazon Managed Service for Prometheus로 지표를 직접 보낼 수 있습니다. 원격 쓰기를 직접 사용하는 방법 및 원격 쓰기 구성에 대한 자세한 내용은 Prometheus 설명서의 remote_write를 참조하세요.

주제

AWS 매니지드 컬렉터

Amazon Managed Service for Prometheus의 일반적인 사용 사례는 Amazon Elastic Kubernetes Service(Amazon EKS)에서 관리되는 Kubernetes 클러스터를 모니터링하는 것입니다. Kubernetes 클러스터와 Amazon EKS 내에서 실행되는 많은 애플리케이션은 Prometheus 호환 스크레이퍼가 액세스할 수 있도록 지표를 자동으로 내보냅니다.

> **Note**
>
> Kubernetes 환경에서 실행되는 많은 기술과 애플리케이션은 Prometheus 호환 지표를 제공합니다. 체계적으로 문서화된 내보내기 목록은 Prometheus 문서에서 내보내기 및 통합을 참조하세요.

Amazon Managed Service for Prometheus는 에이전트 없는 완전 관리형 스크레이퍼 또는 수집기를 제공합니다. 이 스크레이퍼 또는 수집기는 Prometheus 호환 지표를 자동으로 검색하고 가져옵니다. 에이전트나 스크레이퍼를 관리, 설치, 패치 또는 유지 관리할 필요가 없습니다. Amazon Managed Service for Prometheus 수집기는 Amazon EKS 클러스터에 대해 신뢰할 수 있고 안정적이며 가용성 높고 자동으로 확장되는 지표 모음을 제공합니다. Prometheus용 아마존 매니지드 서비스 매니지드 컬렉터는 EC2 및 Fargate를 포함한 Amazon EKS 클러스터와 함께 작동합니다.

Amazon Managed Service for Prometheus 수집기는 스크레이퍼를 생성할 때 지정된 서브넷별로 탄력적 네트워크 인터페이스(ENI)를 생성합니다. 수집기는 이러한 ENI를 통해 지표를 스크래핑하고 `remote_write`를 사용하여 VPC 엔드포인트를 통해 Amazon Managed Service for Prometheus WorkSpace로 데이터를 푸시합니다. 스크래핑한 데이터는 퍼블릭 인터넷을 통해 전송되지 않습니다.

다음 주제에서는 Amazon EKS 클러스터에서 Amazon Managed Service for Prometheus 수집기를 사용하는 방법 및 수집된 지표에 대한 자세한 정보를 제공합니다.

주제

- AWS 관리형 컬렉터 사용
- Prometheus 호환 지표란 무엇입니까?

AWS 관리형 컬렉터 사용

Amazon Managed Service for Prometheus 수집기를 사용하려면 Amazon EKS 클러스터에서 지표를 검색하고 가져오는 스크레이퍼를 생성해야 합니다.

- Amazon EKS 클러스터 생성의 일부로 스크레이퍼를 생성할 수 있습니다. 스크레이퍼 생성을 포함한 Amazon EKS 클러스터를 생성하는 방법에 대한 자세한 내용은 Amazon EKS 사용 설명서의 Amazon EKS 클러스터 생성을 참조하세요.
- AWS API를 사용하거나 를 사용하여 프로그래밍 방식으로 자체 스크레이퍼를 만들 수 있습니다. AWS CLI

> **Note**
>
> 고객 관리 키로 생성한 Amazon Managed Service for Prometheus 작업 영역은 수집에 관리형 컬렉터를 사용할 수 없습니다. AWS

Amazon Managed Service for Prometheus 수집기는 Prometheus 호환 지표를 스크래핑합니다. Prometheus 호환 지표에 대한 자세한 내용은 Prometheus 호환 지표란 무엇입니까? 섹션을 참조하세요.

다음 주제에서는 스크레이퍼를 생성, 관리 및 구성하는 방법을 설명합니다.

주제

- 스크레이퍼 생성
- Amazon EKS 클러스터 구성
- 스크레이퍼 찾기 및 삭제
- 스크레이퍼 구성
- 스크레이퍼 구성 문제 해결
- 스크레이퍼 제한 사항

스크레이퍼 생성

Amazon Managed Service for Prometheus 수집기는 Amazon EKS 클러스터에서 지표를 검색하고 수집하는 스크레이퍼로 구성됩니다. Amazon Managed Service for Prometheus가 스크레이퍼를 관리하므로 인스턴스, 에이전트 또는 스크레이퍼를 직접 관리할 필요 없이 필요한 확장성, 보안 및 신뢰성을 제공합니다.

Amazon EKS 콘솔을 통해 Amazon EKS 클러스터를 생성하면 스크레이퍼가 자동으로 생성됩니다. 하지만 경우에 따라 스크레이퍼를 직접 생성하기를 원할 수도 있습니다. 예를 들어 기존 Amazon EKS 클러스터에 AWS 관리형 수집기를 추가하거나 기존 수집기의 구성을 변경하려는 경우가 이에 해당합니다.

AWS API 또는 를 사용하여 스크레이퍼를 생성할 수 있습니다. AWS CLI

나만의 스크레이퍼를 만들기 위한 몇 가지 사전 조건은 다음과 같습니다.

- Amazon EKS 클러스터가 생성되어 있어야 합니다.

- Amazon EKS 클러스터에 클러스터 엔드포인트 액세스 제어가 프라이빗 액세스를 포함하도록 설정되어 있어야 합니다. 프라이빗 및 퍼블릭을 포함할 수 있지만 프라이빗은 반드시 포함해야 합니다.

API를 사용하여 스크레이퍼를 만들려면 AWS

`CreateScraper` API 작업을 사용하여 AWS API로 스크레이퍼를 생성합니다. 다음 예제에서는 `us-west-2` 리전에서 스크레이퍼를 생성합니다. 작업 공간 AWS 계정, 보안 및 Amazon EKS 클러스터 정보를 자체 ID로 바꾸고 스크레이퍼에 사용할 구성을 제공해야 합니다.

> **Note**
>
> 2개 이상의 가용 영역에 있는 2개 이상의 서브넷을 포함해야 합니다.

`scrapeConfiguration`은 base64로 인코딩된 Prometheus 구성 YAML 파일입니다. `GetDefaultScraperConfiguration` API 작업을 통해 범용 구성을 다운로드할 수 있습니다. 다음 섹션에는 `scrapeConfiguration`의 형식에 대한 자세한 내용이 포함되어 있습니다.

```
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>
    }
}
```

를 사용하여 스크레이퍼를 만들려면 AWS CLI

`create-scraper` 명령을 사용하여 `us-west-2` 리전에 스크레이퍼를 생성합니다. API 예시와 같이 필요한 정보를 자체 계정의 정보로 바꿔야 합니다.

```
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'}"
```

다음은 AWS API와 함께 사용할 수 있는 스크레이퍼 작업의 전체 목록입니다.

- CreateScraperAPI 작업으로 스크레이퍼를 생성합니다.
- ListScrapersAPI 작업을 사용하여 기존 스크레이퍼를 나열하십시오.
- DeleteScraperAPI 작업으로 스크레이퍼를 삭제합니다.
- DescribeScraperAPI 작업을 통해 스크레이퍼에 대한 자세한 내용을 확인하세요.
- GetDefaultScraperConfigurationAPI 작업을 통해 스크레이퍼의 범용 구성을 확보하십시오.

> **Note**
>
> 스크래핑하려는 Amazon EKS 클러스터는 Amazon Managed Service for Prometheus가 지표에 액세스하는 것을 허용하도록 구성해야 합니다. 다음 주제에서는 클러스터를 구성하는 방법을 설명합니다.

Amazon EKS 클러스터 구성

Amazon EKS 클러스터는 스크레이퍼가 지표에 액세스하는 것을 허용하도록 구성해야 합니다. 다음 단계를 수행하면 액세스가 허용됩니다. 이 절차에서는 `kubectl` 및 AWS CLI를 사용합니다. `kubectl` 설치에 대한 자세한 내용은 Amazon EKS 사용 설명서의 kubectl 설치를 참조하세요.

관리형 지표 스크래핑을 위해 Amazon EKS 클러스터를 구성하려면

1. 다음 텍스트를 사용하여 `clusterrole-binding.yml`이라는 파일을 생성합니다.

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

2. 클러스터에서 다음 명령을 실행합니다.

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

   그러면 클러스터 역할 바인딩 및 규칙이 생성됩니다. 이 예제는 역할 이름으로 `aps-collector-role`을 사용하고, 사용자 이름으로 `aps-collector-user`를 사용합니다.

3. 다음 명령은 ID가 *`scraper-id`*인 스크레이퍼에 대한 정보를 제공합니다. 이 스크레이퍼는 이전 섹션의 명령을 사용하여 생성한 스크레이퍼입니다.

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

4. describe-scraper의 결과에서 다음과 같은 형식의 roleArn을 찾습니다.

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

Amazon EKS에서는 이 ARN에 대해 다른 형식이 필요합니다. 다음 단계에서 사용할 반환된 ARN의 형식을 조정해야 합니다. 다음 형식에 맞게 편집합니다.

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

예를 들어 이 ARN은

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

다음과 같이 작성해야 합니다.

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

5. 이전 단계에서 수정한 roleArn과 클러스터 이름 및 리전을 사용하여 클러스터에서 다음 명령을 실행합니다.

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

이렇게 하면 스크레이퍼가 clusterrole-binding.yml 파일에서 생성한 역할과 사용자를 사용하여 클러스터에 액세스할 수 있습니다.

스크레이퍼 찾기 및 삭제

AWS API 또는 를 사용하여 계정의 스크래퍼를 나열하거나 삭제할 수 있습니다. AWS CLI

> **Note**
>
> 최신 버전의 AWS CLI 또는 SDK를 사용하고 있는지 확인하세요. 최신 버전은 보안 업데이트뿐 아니라 최신 특징과 기능을 제공합니다. 또는 항상 up-to-date 명령줄 환경을 자동으로 제공하는 AWS Cloudshell을 사용할 수도 있습니다.

계정의 모든 스크레이퍼를 나열하려면 API 작업을 사용하세요. ListScrapers

또는 를 사용하여 다음과 같이 호출할 AWS CLI수 있습니다.

```
aws amp list-scrapers
```

ListScrapers가 계정의 모든 스크레이퍼를 반환합니다. 예를 들면 다음과 같습니다.

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

스크레이퍼를 삭제하려면 작업을 사용하여 삭제하려는 스크레이퍼를 찾은 다음 `ListScrapers` 작업을 사용하여 삭제하십시오. `scraperId` DeleteScraper

또는 를 사용하여 다음과 AWS CLI같이 호출하십시오.

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

스크레이퍼 구성

Prometheus 호환 스크레이퍼 구성을 사용하여 스크레이퍼가 지표를 검색하고 수집하는 방법을 제어할 수 있습니다. 예를 들어 지표가 WorkSpace로 전송되는 간격을 변경할 수 있습니다. 레이블 재지정을 사용하여 지표의 레이블을 동적으로 다시 작성할 수도 있습니다. 스크레이퍼 구성은 스크레이퍼 정의의 일부인 YAML 파일입니다.

가능한 값에 대한 자세한 분석을 포함하여 스크레이퍼 구성 형식에 대한 자세한 내용은 Prometheus 설명서의 구성을 참조하세요. 글로벌 구성 옵션 및 `<scrape_config>` 옵션은 가장 일반적으로 필요한 옵션을 설명합니다.

새 스크레이퍼가 생성되면 API 호출에서 base64로 인코딩된 YAML 파일을 제공하여 구성을 지정합니다. Amazon Managed Service for Prometheus API에서 `GetDefaultScraperConfiguration` 작업이 포함된 범용 구성 파일을 다운로드할 수 있습니다.

스크레이퍼의 구성을 수정하려면 스크레이퍼를 삭제하고 새 구성으로 다시 생성합니다.

샘플 구성 파일

다음은 스크래핑 간격이 30초인 샘플 YAML 구성 파일입니다.

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

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

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

AWS 관리 컬렉터에는 다음과 같은 두 가지 제한 사항이 있습니다.

- 스크래핑 간격 - 스크레이퍼 구성에서는 스크래핑 간격을 30초 미만으로 지정할 수 없습니다.
- 대상 - `static_config`의 대상을 IP 주소로 지정해야 합니다.

스크레이퍼 구성 문제 해결

Amazon Managed Service for Prometheus 수집기는 자동으로 지표를 검색하고 스크래핑합니다. 하지만 Amazon Managed Service for Prometheus WorkSpace에 표시될 것으로 예상되는 지표가 표시되지 않는 경우 어떻게 문제를 해결할 수 있습니까?

up 지표는 유용한 도구입니다. Amazon Managed Service for Prometheus 수집기가 발견한 각 엔드포인트에 대해 이 지표를 자동으로 제공합니다. 이 지표에는 수집기 내에서 발생하는 문제를 해결하는 데 도움이 되는 세 가지 상태가 있습니다.

- up이 존재하지 않음 - 엔드포인트에 대한 up 지표가 없는 경우 수집기가 엔드포인트를 찾을 수 없었음을 의미합니다.

  엔드포인트가 존재한다고 확신하는 경우 스크래핑 구성을 조정해야 할 수 있습니다. 검색 `relabel_config`를 조정해야 할 수도 있고, 검색에 사용되는 `role`에 문제가 있을 수도 있습니다.
- up이 존재하지만 항상 0임 - up이 존재하지만 0인 경우 수집기는 엔드포인트를 검색할 수 있지만 Prometheus 호환 지표를 찾을 수 없습니다.

  이 경우 엔드포인트에 대한 `curl` 명령을 직접 사용해 볼 수 있습니다. 사용 중인 프로토콜 (`http`또는 엔드포인트 또는 `https` 포트) 과 같은 세부 정보가 정확한지 확인할 수 있습니다. 또한 엔드포인트가 유효한 `200` 응답으로 응답하고 있으며 Prometheus 형식을 따르고 있는지 확인할 수 있습니다. 마지막으로, 응답 본문은 최대 허용 크기보다 클 수 없습니다. (AWS 관리형 컬렉터에 대한 제한은 다음 섹션을 참조하십시오.)
- up이 존재하고 0보다 큰 경우 - up이 존재하고 0보다 크면 지표가 Amazon Managed Service for Prometheus로 전송됩니다.

Amazon Managed Service for Prometheus(또는 Amazon Managed Grafana와 같은 대체 대시보드)에서 올바른 지표를 찾고 있는지 확인합니다. curl을 다시 사용하여 `/metrics` 엔드포인트에서 예상 데이터를 확인할 수 있습니다. 또한 스크레이퍼당 엔드포인트 수와 같은 다른 한도를 초과하지 않았는지도 확인합니다.

스크레이퍼 제한 사항

Amazon Managed Service for Prometheus에서 제공하는 완전 관리형 스크레이퍼에는 몇 가지 제한이 있습니다.

- 리전 - EKS 클러스터, 관리형 스크레이퍼 및 Amazon Managed Service for Prometheus WorkSpace가 모두 동일한 AWS 리전에 있어야 합니다.
- 계정 - EKS 클러스터, 관리형 스크레이퍼 및 Amazon Managed Service for Prometheus WorkSpace가 모두 동일한 AWS 계정에 있어야 합니다.
- 수집기 - 계정별로 리전당 최대 10개의 Amazon Managed Service for Prometheus 스크레이퍼를 보유할 수 있습니다.

> **Note**
>
> 할당량 증가를 요청하여 이 한도에 대한 증가를 요청할 수 있습니다.

- 지표 응답 - 한 `/metrics` 엔드포인트 요청의 응답 본문은 50메가바이트(MB)를 초과할 수 없습니다.
- 스크레이퍼당 엔드포인트 - 스크레이퍼는 최대 30,000개의 엔드포인트를 스크래핑할 수 있습니다.
- 스크래핑 간격 - 스크레이퍼 구성에서는 스크래핑 간격을 30초 미만으로 지정할 수 없습니다.

Prometheus 호환 지표란 무엇입니까?

Amazon Managed Service for Prometheus에서 사용하기 위해 애플리케이션과 인프라에서 Prometheus 지표를 스크래핑하려면 Prometheus 호환 `/metrics` 엔드포인트에서 Prometheus 호환 지표를 계측하여 공개해야 합니다. 자체 지표를 구현할 수 있지만 반드시 그럴 필요는 없습니다. Kubernetes(Amazon EKS 포함) 및 기타 여러 라이브러리 및 서비스는 이러한 지표를 직접 구현합니다.

Amazon EKS의 지표를 Prometheus 호환 엔드포인트로 내보내는 경우 Amazon Managed Service for Prometheus 수집기가 해당 지표를 자동으로 스크래핑하도록 할 수 있습니다.

자세한 정보는 다음 주제를 참조하세요.

- 지표를 Prometheus 지표로 내보내는 기존 라이브러리 및 서비스에 대한 자세한 내용은 Prometheus 설명서의 내보내기 및 통합을 참조하세요.
- 자체 코드에서 Prometheus 호환 지표를 내보내는 방법에 대한 자세한 내용은 Prometheus 설명서의 내보내기 작성을 참조하세요.
- Amazon EKS 클러스터의 지표를 자동으로 스크래핑하도록 Amazon Managed Service for Prometheus 수집기를 설정하는 방법에 대한 자세한 내용은 AWS 관리형 컬렉터 사용 섹션을 참조하세요.

고객 관리형 수집기

이 섹션에는 Prometheus 원격 쓰기를 사용하여 Amazon Managed Service for Prometheus로 지표를 보내는 자체 수집기를 설정하여 데이터를 수집하는 방법에 대한 정보가 포함되어 있습니다.

자체 수집기를 사용하여 Amazon Managed Service for Prometheus로 지표를 보내는 경우, 지표를 보호하고 수집 프로세스가 가용성 요구 사항을 충족하도록 확인해야 합니다.

대부분의 고객 관리형 수집기는 다음 도구 중 하나를 사용합니다.

- AWS Distro for OpenTelemetry (ADOT) — ADOT는 에이전트가 메트릭을 수집할 수 있도록 OpenTelemetry 하는 완전히 지원되고 안전한 프로덕션 준비가 된 오픈 소스 배포입니다. ADOT를 사용해 지표를 수집하여 Amazon Managed Service for Prometheus WorkSpace로 보낼 수 있습니다. ADOT 컬렉터에 대한 자세한 내용은 배포판을 참조하십시오.AWS OpenTelemetry
- Prometheus 에이전트 - 에이전트로 실행되는 오픈 소스 Prometheus 서버의 자체 인스턴스를 설정하여 지표를 수집하고 이를 Amazon Managed Service for Prometheus WorkSpace에 전달할 수 있습니다.

다음 주제에서는 두 도구를 모두 사용하는 방법을 설명하고 자체 수집기 설정에 대한 일반적인 정보를 포함합니다.

주제

- 지표 수집 보호
- AWS Distro for를 OpenTelemetry 컬렉터로 사용
- Prometheus 인스턴스를 수집기로 사용
- 고가용성 데이터를 위해 Amazon Managed Service for Prometheus 설정

지표 수집 보호

Amazon Managed Service for Prometheus는 지표 수집을 보호하는 데 도움이 되는 방법을 제공합니다.

Prometheus용 Amazon 매니지드 서비스와 AWS PrivateLink 함께 사용

Amazon Managed Service for Prometheus로 지표를 수집하는 네트워크 트래픽은 퍼블릭 인터넷 엔드포인트를 통해 또는 VPC 엔드포인트를 통해 수행될 수 있습니다. AWS PrivateLink AWS PrivateLink를 사용하면 VPC의 네트워크 트래픽을 퍼블릭 인터넷을 거치지 않고 AWS 네트워크 내에서 보호할 수 있습니다. Prometheus용 Amazon 관리형 서비스를 위한 AWS PrivateLink VPC 엔드포인트를 생성하려면 을 참조하십시오. 인터페이스 VPC 엔드포인트에서 Amazon Managed Service for Prometheus 사용

인증 및 권한 부여

AWS Identity 및 Access Management (IAM) 는 리소스에 대한 액세스를 안전하게 제어할 수 있는 웹 서비스입니다. AWS IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상을 제어합니다. Amazon Managed Service for Prometheus는 IAM과 통합되어 데이터를 안전하게 유지하는 데 도움이 됩니다. Amazon Managed Service for Prometheus를 설정할 때는 Prometheus 서버에서 지표를 수집하고 Grafana 서버가 Amazon Managed Service for Prometheus WorkSpace에 저장된 지표를 쿼리할 수 있도록 하는 몇 가지 IAM 역할을 생성해야 합니다. IAM에 대한 자세한 내용은 IAM이란? 섹션을 참조하세요.

Prometheus용 Amazon 관리 서비스를 설정하는 데 도움이 되는 또 다른 AWS 보안 기능은 서명 버전 4 서명 프로세스 (SigV4) 입니다 AWS .AWS 서명 버전 4는 HTTP로 전송된 요청에 인증 정보를 추가하는 AWS 프로세스입니다. 보안을 위해 대부분의 요청에는 액세스 키 ID와 보안 액세스 키로 구성된 액세스 키로 AWS 서명해야 합니다. 이 두 키는 일반적으로 보안 자격 증명이라고 합니다. SigV4에 대한 자세한 내용은 서명 버전 4 서명 프로세스를 참조하세요.

AWS Distro for를 OpenTelemetry 컬렉터로 사용

다음 항목에서는 AWS Distro를 지표용 OpenTelemetry 컬렉터로 설정하는 다양한 방법을 설명합니다.

주제

- Amazon Elastic Kubernetes Service 클러스터에서 오픈 텔레메트리용 AWS 배포판을 사용하여 지표 수집을 설정합니다.
- 오픈 텔레메트리용 배포판을 사용하여 AWS Amazon ECS에서 지표 수집 설정
- 원격 쓰기를 사용한 Amazon EC2 인스턴스에서의 지표 수집 설정

Amazon Elastic Kubernetes Service 클러스터에서 오픈 텔레메트리용 AWS 배포판을 사용하여 지표 수집을 설정합니다.

이 섹션에서는 Prometheus 계측 애플리케이션에서 스크랩하도록 AWS 배포판 OpenTelemetry (ADOT) Collector를 구성하고, 지표를 Prometheus용 Amazon Managed Service로 보내는 방법을 설명합니다. ADOT 컬렉터에 대한 자세한 내용은 배포판을 참조하십시오.AWS OpenTelemetry

ADOT를 통한 Prometheus 메트릭 수집에는 Prometheus 수신기, Prometheus 원격 쓰기 익스포터 및 Sigv4 인증 확장이라는 세 가지 OpenTelemetry 구성 요소가 포함됩니다.

기존 Prometheus 구성을 사용하여 서비스 검색 및 지표 스크래핑을 수행하도록 Prometheus Receiver를 구성할 수 있습니다. Prometheus Receiver는 Prometheus 표시 형식으로 지표를 스크래핑합니다. 스크래핑하려는 모든 애플리케이션 또는 엔드포인트는 Prometheus 클라이언트 라이브러리로 구성해야 합니다. Prometheus Receiver는 Prometheus 설명서의 구성에 설명된 Prometheus 스크래핑 및 레이블 재지정 구성의 전체 세트를 지원합니다. 이러한 구성을 ADOT Collector 구성에 직접 붙여 넣을 수 있습니다.

Prometheus Remote Write Exporter는 `remote_write` 엔드포인트를 사용하여 스크래핑된 지표를 관리 포털 워크스페이스로 보냅니다. 데이터 내보내기를 위한 HTTP 요청은 Sigv4 인증 확장을 통해 보안 인증을 위한 프로토콜인 AWS SigV4로 서명됩니다. AWS 자세한 내용은 서명 버전 4 서명 프로세스를 참조하세요.

수집기는 Amazon EKS에서 Prometheus 지표 엔드포인트를 자동으로 검색하고 <kubernetes_sd_config>에 있는 구성을 사용합니다.

다음 데모는 Amazon Elastic Kubernetes Service 또는 자체 관리형 Kubernetes를 실행하는 클러스터에서 사용되는 이러한 구성의 예입니다. 이 단계를 수행하려면 기본 자격 증명 체인에 있는 잠재적 옵션의 AWS 자격 증명이 있어야 합니다. AWS 자세한 내용은 Go용 AWS SDK 구성을 참조하십시오. 이 데모에서는 프로세스의 통합 테스트에 사용되는 샘플 앱을 사용합니다. 샘플 앱은 Prometheus 클라이언트 라이브러리처럼 `/metrics` 엔드포인트에서 지표를 노출합니다.

필수 조건

아래 수집 설정 단계를 시작하기 전에 서비스 계정 및 신뢰 정책에 대한 IAM 역할을 설정해야 합니다.

서비스 계정 및 신뢰 정책에 대한 IAM 역할을 설정하려면

1. Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정의 단계에 따라 서비스 계정의 IAM 역할을 생성합니다.

   ADOT Collector는 지표를 스크래핑하고 내보낼 때 이 역할을 사용합니다.

2. 다음으로 신뢰 정책을 편집합니다. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
3. 왼쪽 탐색 창에서 역할을 선택하고 1단계에서 amp-iamproxy-ingest-role생성한 역할을 찾습니다.
4. 신뢰 관계 탭을 선택한 후 신뢰 관계 편집을 선택합니다.
5. 신뢰 관계 정책 JSON에서 `aws-amp`를 `adot-col`로 바꾼 다음, 신뢰 정책 업데이트를 선택합니다. 결과 신뢰 정책은 다음과 같아야 합니다.

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

6. 권한 탭을 선택하고 다음 권한 정책이 역할에 연결되어 있는지 확인합니다.

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

```
}
```

Prometheus 지표 수집 활성화

> **Note**
>
> Amazon EKS에서 네임스페이스를 생성하면 `alertmanager` 및 노드 내보내기가 기본적으로 비활성화됩니다.

Amazon EKS 또는 Kubernetes 클러스터에서 Prometheus 수집을 활성화하려면

1. 의 리포지토리에서 샘플 앱을 포크 및 aws-otel-community복제합니다.

   그런 후 다음 명령을 실행합니다.

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

2. 이 이미지를 Amazon ECR 또는 DockerHub 같은 레지스트리에 푸시하십시오.
3. 이 Kubernetes 구성을 복사하고 적용하여 클러스터에 샘플 앱을 배포합니다. `prometheus-sample-app.yaml` 파일에서 `{{PUBLIC_SAMPLE_APP_IMAGE}}`를 대체하여 이미지를 방금 푸시한 이미지로 변경합니다.

   ```
   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. 다음 명령을 입력하여 샘플 앱이 시작되었는지 확인합니다. 명령 출력의 `NAME` 열에 `prometheus-sample-app`이 표시됩니다.

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

5. ADOT Collector의 기본 인스턴스를 시작합니다. 이렇게 하려면 먼저 다음 명령을 입력하여 ADOT Collector의 Kubernetes 구성을 끌어옵니다.

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

그런 다음, 템플릿 파일을 편집하여 Amazon Managed Service for Prometheus 워크스페이스에 대한 remote_write 엔드포인트를 YOUR_ENDPOINT로 바꾸고 리전을 YOUR_REGION으로 바꿉니다. 워크스페이스 세부 정보를 확인할 때 Amazon Managed Service for Prometheus 콘솔에 표시되는 remote_write 엔드포인트를 사용합니다.

또한 Kubernetes 구성의 서비스 계정 섹션에서 계정 ID로 YOUR_ACCOUNT_ID 변경해야 합니다. AWS

이 예제에서 ADOT Collector 구성은 주석(scrape=true)을 사용하여 스크래핑할 대상 엔드포인트를 알려줍니다. 이를 통해 ADOT Collector는 샘플 앱 엔드포인트를 클러스터의 kube-system 엔드포인트와 구별할 수 있습니다. 다른 샘플 앱을 스크래핑하려는 경우 레이블 재지정 구성에서 이 앱을 제거할 수 있습니다.

6. 다음 명령을 입력하여 ADOT Collector를 배포합니다.

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

7. 다음 명령을 입력하여 ADOT Collector가 시작되었는지 확인합니다. NAMESPACE 열에서 adot-col을 찾아봅니다.

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

8. 로깅 내보내기를 사용하여 파이프라인이 작동하는지 확인합니다. 예제 템플릿은 로깅 내보내기와 이미 통합되어 있습니다. 다음 명령을 입력합니다.

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

샘플 앱에서 스크래핑한 지표 중 일부는 다음 예와 같습니다.

```
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. Amazon Managed Service for Prometheus가 지표를 수신했는지 테스트하려면 `awscurl`을 사용합니다. 이 도구를 사용하면 AWS Sigv4 인증을 통해 명령줄을 통해 HTTP 요청을 보낼 수 있으므로 Prometheus용 Amazon Managed Service에서 쿼리하려면 올바른 권한을 가진 AWS 자격 증명을 로컬에 설정해야 합니다. 설치 지침은 awscurl을 참조하십시오. `awscurl`

   다음 명령에서 `AMP_REGION`과 `AMP_ENDPOINT`를 사용자의 Amazon Managed Service for Prometheus WorkSpace에 대한 정보로 바꿉니다.

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

   응답으로 지표가 수신되면 파이프라인 설정이 성공적으로 완료되었고 지표가 샘플 앱에서 Amazon Managed Service for Prometheus로 성공적으로 전파되었음을 의미합니다.

정리

이 데모를 정리하려면 다음 명령을 입력합니다.

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

고급 구성

Prometheus Receiver는 Prometheus 설명서의 구성에 설명된 Prometheus 스크래핑 및 레이블 재지정 구성의 전체 세트를 지원합니다. 이러한 구성을 ADOT Collector 구성에 직접 붙여 넣을 수 있습니다.

Prometheus Receiver의 구성에는 서비스 검색, 스크래핑 구성 및 레이블 재지정 구성이 포함됩니다. 수신기 구성은 다음과 같습니다.

```
receivers:
  prometheus:
```

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

다음은 예제 구성입니다.

```
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
```

기존 Prometheus 구성이 있는 경우 값이 환경 변수로 바뀌지 않도록 $ 문자를 $$로 바꿔야 합니다. *이 작업은 relabel_configurations의 대체 값에 특히 중요합니다. 예를 들어 다음과 같이 relabel_configuration으로 시작하는 경우

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

다음과 같이 됩니다.

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

Prometheus Remote Write Exporter 및 Sigv4 Authentication Extension

Prometheus Remote Write Exporter와 Sigv4 Authentication Extension의 구성은 Prometheus Receiver보다 간단합니다. 파이프라인의 이 단계에서는 이미 지표가 수집되었으며 이 데이터를 Amazon Managed Service for Prometheus로 내보낼 준비가 되었습니다. Amazon Managed Service for Prometheus와 통신하기 위한 성공적인 구성의 최소 요구 사항은 다음 예제에 나와 있습니다.

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

이 구성은 기본 자격 증명 체인의 자격 증명을 사용하여 SigV4에서 서명한 AWS HTTPS 요청을 전송합니다. AWS AWS 자세한 내용은 AWS SDK for Go구성 섹션을 참조하세요. 서비스를 aps로 지정해야 합니다.

배포 방법에 관계없이 ADOT 수집기는 기본 자격 증명 체인에 나열된 옵션 중 하나에 액세스할 수 있어야 합니다. AWS Sigv4 인증 확장은 에 AWS SDK for Go 의존하며 이를 사용하여 자격 증명을 가져오고 인증합니다. 이러한 보안 인증에 Amazon Managed Service for Prometheus에 대한 원격 쓰기 권한이 있는지 확인해야 합니다.

오픈 텔레메트리용 배포판을 사용하여 AWS Amazon ECS에서 지표 수집 설정

이 섹션에서는 오픈 텔레메트리용 배포판 (ADOT) 을 사용하여 Amazon Elastic Container Service (Amazon ECS) 에서 지표를 수집하고 이를 AWS 프로메테우스용 아마존 매니지드 서비스에 수집하는 방법을 설명합니다. 또한 Amazon Managed Grafana에서 지표를 시각화하는 방법도 설명합니다.

필수 조건

> **Important**
>
> 시작하기 전에 기본 설정이 적용된 AWS Fargate 클러스터의 Amazon ECS 환경, Amazon Managed Service for Prometheus 워크스페이스, Amazon Managed Grafana 워크스페이스가 있어야 합니다. 컨테이너 워크로드, Amazon Managed Service for Prometheus, Amazon Managed Grafana에 대해 잘 알고 있다고 가정합니다.

자세한 내용은 다음 링크를 참조하십시오.

- 기본 설정이 적용된 Fargate 클러스터에서 Amazon ECS 환경을 생성하는 방법에 대한 자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 생성을 참조하세요.
- Amazon Managed Service for Prometheus 워크스페이스를 생성하는 방법에 대한 자세한 내용은 Amazon Managed Service for Prometheus 사용 설명서에서 워크스페이스 생성을 참조하세요.
- Amazon Managed Grafana 워크스페이스를 생성하는 방법에 대한 자세한 내용은 Amazon Managed Grafana 사용 설명서에서 워크스페이스 생성을 참조하세요.

사용자 지정 ADOT Collector 컨테이너 이미지 정의

다음 구성 파일을 템플릿으로 사용하여 고유한 ADOT Collector 컨테이너 이미지를 정의합니다. *my-remote-URL* 및 *my-region*을 각각 `endpoint` 및 `region` 값으로 바꿉니다. 구성을 adot-config.yaml이라는 파일에 저장합니다.

> **Note**
>
> 이 구성에서는 `sigv4auth` 확장 프로그램을 사용하여 Amazon Managed Service for Prometheus에 대한 호출을 인증합니다. 구성에 대한 자세한 내용은 Authenticator - Sigv4 on `sigv4auth` GitHub을 참조하십시오.

```
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]
```

ADOT Collector 컨테이너 이미지를 Amazon ECR 리포지토리에 푸시

Dockerfile을 사용하여 컨테이너 이미지를 생성하고 Amazon Elastic Container Registry(ECR) 리포지토리에 푸시합니다.

1. Dockerfile을 빌드하여 컨테이너 이미지를 복사하고 OTEL 도커 이미지에 추가합니다.

```
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. Amazon ECR 리포지토리를 생성합니다.

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

3. 컨테이너 이미지를 생성합니다.

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

> **Note**
>
> 여기서는 컨테이너가 실행될 환경과 동일한 환경에서 컨테이너를 빌드한다고 가정합니다. 그렇지 않은 경우 이미지를 빌드할 때 `--platform` 파라미터를 사용해야 할 수 있습니다.

4. Amazon ECR 리포지토리에 로그인합니다. *my-region*을 `region` 값으로 바꿉니다.

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

5. 컨테이너 이미지를 푸시합니다.

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

Amazon Managed Service for Prometheus를 스크래핑할 Amazon ECS 태스크 정의 생성

Amazon Managed Service for Prometheus를 스크래핑할 Amazon ECS 태스크 정의를 생성합니다. 태스크 정의에는 이름이 `adot-collector`인 컨테이너와 이름이 `prometheus`인 컨테이너가 포함되어야 합니다. `prometheus`는 지표를 생성하고 `adot-collector`는 `prometheus`를 스크래핑합니다.

> **Note**
>
> Amazon Managed Service for Prometheus는 서비스로 실행되며 컨테이너에서 지표를 수집합니다. 이 경우 컨테이너는 Prometheus를 로컬에서 에이전트 모드로 실행하여 로컬 지표는 Amazon Managed Service for Prometheus로 전송됩니다.

예제: 태스크 정의

다음은 태스크 정의의 모양을 보여 주는 예제입니다. 이 예제를 템플릿으로 사용하여 자체 태스크 정의를 생성할 수 있습니다. `adot-collector`의 `image` 값을 리포지토리 URL 및 이미지 태그(`$COLLECTOR_REPOSITORY:ecs`)로 바꿉니다. `adot-collector` 및 `prometheus`의 `region` 값을 `region` 값으로 바꿉니다.

```
{
  "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"
}
```

AWS 관리형 정책 **AmazonPrometheusRemoteWriteAccess**를 태스크에 대한 IAM 역할에 연결

스크랩된 지표를 Prometheus용 Amazon Managed Service로 보내려면 Amazon ECS 작업에 API 작업을 호출할 수 있는 올바른 권한이 있어야 합니다. AWS 태스크에 대한 IAM 역할을 생성하고 이 역할에 `AmazonPrometheusRemoteWriteAccess` 정책을 연결해야 합니다. 이 역할을 생성하고 정책을 연결하는 방법에 대한 자세한 내용은 태스크에 대한 IAM 역할 및 정책 생성을 참조하세요.

`AmazonPrometheusRemoteWriteAccess`를 IAM 역할에 연결하고 해당 역할을 태스크에 사용하면 Amazon ECS에서 스크래핑한 지표를 Amazon Managed Service for Prometheus로 보낼 수 있습니다.

Amazon Managed Grafana에서 지표 시각화

> **Important**
>
> 시작하기 전에 Amazon ECS 태스크 정의에서 Fargate 태스크를 실행해야 합니다. 그렇지 않으면 Amazon Managed Service for Prometheus에서 지표를 사용할 수 없습니다.

1. Amazon Managed Grafana 워크스페이스의 탐색 창에서 아이콘 아래에 있는 데이터 소스를 선택합니다. AWS
2. 데이터 소스 탭의 서비스에서 Amazon Managed Service for Prometheus를 선택하고 기본 리전을 선택합니다.
3. 데이터 소스 추가를 선택합니다.
4. `ecs` 및 `prometheus` 접두사를 사용하여 지표를 쿼리하고 확인합니다.

원격 쓰기를 사용한 Amazon EC2 인스턴스에서의 지표 수집 설정

이 섹션에서는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에서 원격 쓰기로 Prometheus 서버를 실행하는 방법을 설명합니다. Go로 작성된 데모 애플리케이션에서 지표를 수집한 후 Amazon Managed Service for Prometheus 워크스페이스로 보내는 방법을 설명합니다.

필수 조건

> **Important**
>
> 시작하기 전에 Prometheus v2.26 이상을 설치해야 합니다. Prometheus, Amazon EC2 및 Amazon Managed Service for Prometheus에 대해 잘 알고 있다고 가정합니다. Prometheus 설치 방법에 대한 자세한 내용은 Prometheus 웹 사이트에서 시작하기를 참조하세요.

Amazon EC2 또는 Amazon Managed Service for Prometheus에 익숙하지 않은 경우 먼저 다음 섹션을 읽는 것이 좋습니다.

- Amazon Elastic Compute Cloud란?
- Amazon Managed Service for Prometheus란?

Amazon EC2의 IAM 역할 생성

지표를 스트리밍하려면 먼저 관리형 정책을 사용하여 IAM 역할을 생성해야 합니다. AWS AmazonPrometheusRemoteWriteAccess 그런 다음, 역할과 함께 인스턴스를 시작하고 지표를 Amazon Managed Service for Prometheus WorkSpace로 스트리밍할 수 있습니다.

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.
2. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.
3. 신뢰할 수 있는 엔터티 유형에서 AWS 서비스를 선택합니다. 사용 사례로 EC2를 선택합니다. 다음: 권한을 선택합니다.
4. 검색 창에 AmazonPrometheusRemoteWriteAccess를 입력합니다. 정책 이름에서 을 선택한 AmazonPrometheusRemoteWriteAccess다음 Attach policy (정책 연결) 를 선택합니다. 다음: 태그를 선택합니다.
5. (선택 사항) IAM 역할을 위한 IAM 태그를 생성합니다. 다음: 검토를 선택합니다.
6. 역할의 이름을 입력합니다. 정책 생성(Create policy)을 선택합니다.

Amazon EC2 인스턴스 시작하기

Amazon EC2 인스턴스를 시작하려면 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스 시작의 지침을 따르세요.

데모 애플리케이션 실행

1. 다음 템플릿을 사용하여 `main.go`라는 Go 파일을 생성합니다.

```
package main

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

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

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

2. 다음 명령을 실행하여 올바른 종속성을 설치합니다.

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

3. 데모 애플리케이션을 실행합니다.

```
go run main.go
```

   데모 애플리케이션은 포트 8000에서 실행되어야 하며 노출된 모든 Prometheus 지표를 표시합니다. 다음은 이러한 지표의 예입니다.

```
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
```

Amazon Managed Service for Prometheus WorkSpace 생성

Amazon Managed Service for Prometheus WorkSpace를 생성하려면 Amazon Managed Service for Prometheus 사용 설명서에서 WorkSpace 생성의 지침을 따르세요.

Prometheus 서버 실행

1. 다음 예제 YAML 파일을 템플릿으로 사용하여 `prometheus.yaml`이라는 새 파일을 생성합니다. 의 경우 *`my-region# ##`* 값과 Prometheus용 `url` Amazon Managed Service에서 생성한 작업 공간 *`my-workspace-id`*ID로 바꾸십시오. `region`의 경우 *`my-region`*을 리전 값으로 바꿉니다.

   예제: YAML 파일

   ```
   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. Prometheus 서버를 실행하여 데모 애플리케이션의 지표를 Amazon Managed Service for Prometheus 워크스페이스로 전송합니다.

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

이제 Prometheus 서버가 데모 애플리케이션의 지표를 Amazon Managed Service for Prometheus WorkSpace로 전송합니다.

Prometheus 인스턴스를 수집기로 사용

다음 주제에서는 에이전트 모드에서 실행되는 Prometheus 인스턴스를 지표 수집기로 설정하는 다양한 방법을 설명합니다.

> **Warning**
> 보안 기능을 활성화하여 Prometheus Scrape 엔드포인트를 공용 인터넷에 노출시키지 마세요.

동일한 지표 세트를 모니터링하는 여러 Prometheus 인스턴스를 설정하고 고가용성을 위해 Amazon Managed Service for Prometheus 단일 WorkSpace로 전송하는 경우 중복 제거를 설정해야 합니다. 중복 제거를 설정하는 단계를 따르지 않으면 Amazon Managed Service for Prometheus로 전송된 모든 데이터 샘플(중복 샘플 포함)에 대한 요금이 부과됩니다. 중복 제거 설정에 대한 지침은 Amazon Managed Service for Prometheus로 전송된 고가용성 지표 중복 제거 섹션을 참조하세요.

주제

- Helm을 사용하여 새 Prometheus 서버에서 수집 설정
- EC2의 Kubernetes에 있는 기존 Prometheus 서버에서의 수집 설정
- Fargate의 Kubernetes에 있는 기존 Prometheus 서버에서의 수집 설정

Helm을 사용하여 새 Prometheus 서버에서 수집 설정

이 섹션의 지침을 통해 Amazon Managed Service for Prometheus를 빠르게 시작하고 실행할 수 있습니다. Amazon EKS 클러스터에 새 Prometheus 서버를 설정하면 새 서버는 기본 구성을 사용하여 Amazon Managed Service for Prometheus로 지표를 전송합니다. 이 방법의 사전 조건은 다음과 같습니다.

- 새 Prometheus 서버가 지표를 수집할 Amazon EKS 클러스터가 있어야 합니다.
- Helm CLI 3.0 이상을 사용해야 합니다.
- 다음 섹션의 단계를 수행하려면 Linux 또는 macOS 컴퓨터를 사용해야 합니다.

1단계: 새 차트 Helm 리포지토리 추가

새 차트 Helm 리포지토리를 추가하려면 다음 명령을 입력합니다. 이러한 명령에 대한 자세한 내용은 Helm 리포지토리를 참조하세요.

```
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
```

2단계: Prometheus 네임스페이스 생성

다음 명령을 입력하여 Prometheus 서버 및 기타 모니터링 구성 요소에 대한 Prometheus 네임스페이스를 생성합니다. *prometheus-namespace*를 이 네임스페이스에 사용할 이름으로 바꿉니다.

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

3단계: 서비스 계정의 IAM 역할 설정

문서화하는 온보딩 방법에 대해서는 Prometheus 서버가 실행되는 Amazon EKS 클러스터에서 서비스 계정에 대한 IAM 역할을 사용해야 합니다.

서비스 계정에 대한 IAM 역할을 사용할 경우 IAM 역할을 Kubernetes 서비스 계정에 연결할 수 있습니다. 이렇게 하면 이 서비스 계정에서는 이 서비스 계정을 사용하는 모든 포드에 있는 컨테이너에 AWS 권한을 제공할 수 있습니다. 자세한 내용은 서비스 계정에 대한 IAM 역할을 참조하세요.

이러한 역할을 아직 설정하지 않은 경우 Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정의 지침에 따라 역할을 설정하세요. 해당 섹션의 지침에는 `eksctl`을 사용해야 합니다. 자세한 내용은 Amazon Elastic Kubernetes Service 시작 - `eksctl`을 참조하세요.

> **Note**
>
> EKS를 사용하지 않거나 AWS 액세스 키와 비밀 키만 사용하여 Prometheus용 Amazon Managed Service에 액세스하는 경우에는 기반 SigV4를 사용할 수 없습니다. `EKS-IAM-ROLE`

4단계: 새 서버 설정 및 지표 수집 시작

Amazon Managed Service for Prometheus 워크스페이스로 지표를 전송하는 새 Prometheus 서버를 설치하려면 다음 단계를 따르세요.

새 Prometheus 서버를 설치하여 Amazon Managed Service for Prometheus 워크스페이스로 지표를 보내려면

1. 텍스트 편집기를 사용하여 다음 내용을 포함하는 `my_prometheus_values_yaml`이라는 파일을 생성합니다.

   - *IAM_PROXY_PROMETHEUS_ROLE_ARN# ## ### ARN##* 바꾸십시오. amp-iamproxy-ingest-roleAmazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정
   - *WORKSPACE_ID*를 Amazon Managed Service for Prometheus 워크스페이스의 ID로 바꿉니다.
   - *REGION*을 Amazon Managed Service for Prometheus 워크스페이스의 리전으로 바꿉니다.

```
## 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. 다음 명령을 입력하여 Prometheus 서버를 생성합니다.

   - Prometheus 출시 *prometheus-chart-name*이름으로 바꾸십시오.
   - *prometheus-namespace*를 Prometheus 네임스페이스의 이름으로 바꿉니다.

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

> **Note**
>
> 여러 가지 방법으로 `helm install` 명령을 사용자 지정할 수 있습니다. 자세한 내용은 Helm 설명서의 Helm 설치를 참조하세요.

EC2의 Kubernetes에 있는 기존 Prometheus 서버에서의 수집 설정

Amazon Managed Service for Prometheus는 Amazon EKS를 실행하는 클러스터와 Amazon EC2에서 실행되는 자체 관리형 Kubernetes 클러스터의 Prometheus 서버에서 지표 수집을 지원합니다. 이 섹션의 세부 지침은 Amazon EKS 클러스터의 Prometheus 서버를 위한 것입니다. Amazon EC2의 자체 관리형 Kubernetes 클러스터의 경우 서비스 계정의 OIDC 공급자 및 IAM 역할을 직접 설정해야 한다는 점을 제외하고 수집 설정 단계는 동일합니다.

이 섹션의 지침에서는 Helm을 Kubernetes 패키지 관리자로 사용합니다.

주제

- 1단계: 서비스 계정의 IAM 역할 설정
- 2단계: Helm을 사용하여 기존 Prometheus 서버 업그레이드

1단계: 서비스 계정의 IAM 역할 설정

문서화하는 온보딩 방법에 대해서는 Prometheus 서버가 실행되는 Amazon EKS 클러스터에서 서비스 계정에 대한 IAM 역할을 사용해야 합니다. 이러한 역할을 서비스 역할이라고도 합니다.

서비스 역할을 사용하면 IAM 역할을 Kubernetes 서비스 계정에 연결할 수 있습니다. 그러면 이 서비스 계정을 사용하여 해당 서비스 계정을 사용하는 모든 포드의 컨테이너에 AWS 권한을 제공할 수 있습니다. 자세한 내용은 서비스 계정에 대한 IAM 역할을 참조하세요.

이러한 역할을 아직 설정하지 않은 경우 Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정의 지침에 따라 역할을 설정하세요.

2단계: Helm을 사용하여 기존 Prometheus 서버 업그레이드

이 섹션의 지침에는 Prometheus 서버가 Amazon Managed Service for Prometheus WorkSpace에 원격 쓰기를 수행할 수 있도록 인증하고 권한을 부여하기 위한 원격 쓰기 및 sigv4 설정 방법이 포함되어 있습니다.

Prometheus 버전 2.26.0 이상 사용

버전 2.26.0 이상의 Prometheus 서버 이미지에서 차트 Helm을 사용하는 경우 다음 단계를 따르세요.

차트 Helm을 사용하여 Prometheus 서버에서 원격 쓰기를 설정하려면

1. Helm 구성 파일에 새 원격 쓰기 섹션을 생성합니다.

   - 에서 amp-iamproxy-ingest-role생성한 `${IAM_PROXY_PROMETHEUS_ROLE_ARN}` ARN으로 대체합니다. 1단계: 서비스 계정의 IAM 역할 설정 역할 ARN의 형식은 `arn:aws:iam::`*`your account ID`*`:role/amp-iamproxy-ingest-role`입니다.
   - `${WORKSPACE_ID}`를 Amazon Managed Service for Prometheus WorkSpace ID로 바꿉니다.
   - `${REGION}`을 Amazon Managed Service for Prometheus WorkSpace의 리전(예: `us-west-2`)으로 바꿉니다.

```
## 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. Helm을 사용하여 기존 Prometheus 서버 구성을 업데이트합니다.

   - `prometheus-chart-name`을 Prometheus 릴리스 이름으로 바꿉니다.
   - `prometheus-namespace`를 Prometheus 서버가 설치된 Kubernetes 네임스페이스로 바꿉니다.
   - `my_prometheus_values_yaml`을 Helm 구성 파일의 경로로 바꿉니다.
   - `current_helm_chart_version`을 Prometheus 서버 차트 Helm의 현재 버전으로 바꿉니다. helm list 명령을 사용하여 현재 차트 버전을 찾을 수 있습니다.

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

이전 버전의 Prometheus 사용

2.26.0 이전의 Prometheus 버전을 사용하는 경우 다음 단계를 따르세요. 이전 버전의 Prometheus에서는 서명 버전 4 서명 프로세스 (SigV4) 를 기본적으로 AWS 지원하지 않기 때문에 이러한 단계에서는 사이드카 접근 방식을 사용합니다.AWS

이 지침에서는 Helm을 사용하여 Prometheus를 배포한다고 가정합니다.

Prometheus 서버에서 원격 쓰기를 설정하려면

1. Prometheus 서버에서 새 원격 쓰기 구성을 생성합니다. 먼저 새 업데이트 파일을 생성합니다. `amp_ingest_override_values.yaml` 파일을 호출합니다.

   YAML 파일에 다음 값을 추가합니다.

```
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
```

`${REGION}`을 Amazon Managed Service for Prometheus 워크스페이스의 리전으로 바꿉니다.

에서 amp-iamproxy-ingest-role생성한 `${SERVICE_ACCOUNT_IAM_INGEST_ROLE_ARN}` ARN으로 대체합니다. 1단계: 서비스 계정의 IAM 역할 설정 역할 ARN의 형식은 `arn:aws:iam::`*`your account ID`*`:role/amp-iamproxy-ingest-role`입니다.

`${WORKSPACE_ID}`를 워크스페이스 ID로 바꿉니다.

2. Prometheus 차트 Helm을 업그레이드합니다. 먼저 다음 명령을 입력하여 차트 Helm 이름을 찾습니다. 이 명령의 출력에서 이름에 `prometheus`가 포함된 차트를 찾아보세요.

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

이어서 다음 명령을 입력합니다.

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

이전 *prometheus-helm-chart-name*명령에서 반환된 Prometheus helm 차트의 이름으로 바꾸십시오. *prometheus-namespace*를 네임스페이스의 이름으로 바꿉니다.

차트 Helm 다운로드

차트 Helm을 아직 로컬로 다운로드하지 않은 경우, 다음 명령을 사용하여 다운로드할 수 있습니다.

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

Fargate의 Kubernetes에 있는 기존 Prometheus 서버에서의 수집 설정

Amazon Managed Service for Prometheus는 Fargate에서 실행되는 자체 관리형 Kubernetes 클러스터의 Prometheus 서버의 지표 수집을 지원합니다. Fargate에서 실행되는 Amazon EKS 클러스터의 Prometheus 서버에서 지표를 수집하려면 다음과 같이 amp_ingest_override_values.yaml이라는 구성 파일의 기본 구성을 재정의하세요.

```
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
```

다음 명령을 사용하여 재정의하여 Prometheus를 설치합니다.

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

차트 Helm 구성에서는 노드 내보내기와 알림 관리자를 비활성화하고 Prometheus 서버 배포를 실행하지 않도록 설정했습니다.

다음 예제 테스트 쿼리를 사용하여 설치를 확인할 수 있습니다.

```
$ 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
```

고가용성 데이터를 위해 Amazon Managed Service for Prometheus 설정

Amazon Managed Service for Prometheus로 데이터를 전송하면 해당 리전의 AWS 가용성 영역 전체에 데이터가 자동으로 복제되며 확장성, 가용성 및 보안을 제공하는 호스트 클러스터에서 사용자에게 제공됩니다. 특정 설정에 따라 고가용성 유사 시 대기를 더 추가할 수 있습니다. 설정에 고가용성 안전 기능을 추가하는 두 가지 일반적인 방법은 다음과 같습니다.

- 동일한 데이터를 포함하는 컨테이너 또는 인스턴스가 여러 개 있는 경우, 해당 데이터를 Amazon Managed Service for Prometheus로 전송하면 데이터의 중복을 자동으로 제거할 수 있습니다. 이렇게 하면 데이터가 Amazon Managed Service for Prometheus 워크스페이스로 전송되도록 할 수 있습니다.

  고가용성 데이터 중복 제거에 대한 자세한 내용은 Amazon Managed Service for Prometheus로 전송된 고가용성 지표 중복 제거 섹션을 참조하세요.
- AWS 리전을 사용할 수 없는 경우에도 데이터에 액세스할 수 있도록 하려면 지표를 다른 리전의 또 다른 WorkSpace로 보낼 수 있습니다.

지표 데이터를 여러 워크스페이스로 보내는 방법에 대한 자세한 내용은 교차 리전 가용성 섹션을 참조하세요.

주제

- Amazon Managed Service for Prometheus로 전송된 고가용성 지표 중복 제거
- Prometheus를 사용하여 Amazon Managed Service for Prometheus로 고가용성 데이터 전송
- Prometheus Operator를 사용하여 Amazon Managed Service for Prometheus로 고가용성 데이터 전송
- 오픈 텔레메트리용 배포판을 사용하여 AWS Prometheus용 Amazon 매니지드 서비스로 고가용성 데이터를 전송
- Prometheus 커뮤니티 Helm 차트를 사용하여 Amazon Managed Service for Prometheus로 고가용성 데이터 전송
- FAQ: 고가용성 구성
- 교차 리전 가용성

Amazon Managed Service for Prometheus로 전송된 고가용성 지표 중복 제거

여러 Prometheus 에이전트(에이전트 모드에서 실행되는 Prometheus 인스턴스)에서 Amazon Managed Service for Prometheus 워크스페이스로 데이터를 보낼 수 있습니다. 이러한 인스턴스 중 일부가 동일한 지표를 기록하고 전송하는 경우 데이터의 가용성이 높아집니다(에이전트 중 하나가 데이터 전송을 중단하더라도 Amazon Managed Service for Prometheus 워크스페이스는 다른 인스턴스에서 데이터를 계속 수신함). 하지만 지표가 여러 번 표시되지 않고 데이터 모으기 및 저장 요금이 여러 번 청구되지 않도록 Amazon Managed Service for Prometheus 워크스페이스에서 지표가 자동으로 중복 제거되도록 할 수 있습니다.

Amazon Managed Service for Prometheus에서 여러 Prometheus 에이전트의 데이터가 자동으로 중복 제거되도록 하려면 중복 데이터를 보내는 에이전트 세트에 단일 클러스터 이름을 지정하고 각 인스턴스에 복제본 이름을 지정합니다. 클러스터 이름은 인스턴스를 공유 데이터가 있는 것으로 식별하며, 복제본 이름은 Amazon Managed Service for Prometheus가 각 지표의 소스를 식별할 수 있도록 합니다. 최종 저장된 지표에는 클러스터 레이블이 포함되지만 복제본은 포함되지 않으므로 지표는 단일 소스에서 가져온 것으로 나타납니다.

> **Note**
>
> 특정 버전의 쿠버네티스 (1.28 및 1.29) 는 레이블이 있는 자체 메트릭을 내보낼 수 있습니다. `cluster` 이로 인해 Prometheus용 Amazon 매니지드 서비스 중복 제거에 문제가 발생할 수 있습니다. 자세한 내용은 고가용성 FAQ를 참조하십시오.

다음 주제에서는 Prometheus용 Amazon Managed Service에서 자동으로 데이터 중복을 제거하도록 데이터를 `cluster` 전송하고 및 `__replica__` 레이블을 포함하는 방법을 보여줍니다.

> **Important**
>
> 중복 제거를 설정하지 않으면 Amazon Managed Service for Prometheus로 전송되는 모든 데이터 샘플에 대해 요금이 부과됩니다. 이러한 데이터 샘플에는 중복 샘플이 포함되어 있습니다.

Prometheus를 사용하여 Amazon Managed Service for Prometheus로 고가용성 데이터 전송

Prometheus를 사용하여 고가용성 구성을 설정하려면 Amazon Managed Service for Prometheus에서 식별할 수 있도록 고가용성 그룹의 모든 인스턴스에 외부 레이블을 적용해야 합니다. Prometheus 인스턴스 에이전트를 고가용성 그룹의 일부로 식별하려면 `cluster` 레이블을 사용합니다. 그룹 내 각 복제본을 개별적으로 식별하려면 `__replica__` 레이블을 사용합니다. 중복 제거가 제대로 작동하려면 `__replica__` 및 `cluster` 레이블을 모두 적용해야 합니다.

> **Note**
>
> `__replica__` 레이블은 단어 `replica` 앞뒤에 두 개의 밑줄 기호를 사용하여 서식이 지정되어 있습니다.

예제: 코드 조각

다음 코드 조각에서 `cluster` 레이블은 Prometheus 인스턴스 에이전트 `prom-team1`을 식별하고 `_replica_` 레이블은 복제본 `replica1` 및 `replica2`를 식별합니다.

```
cluster: prom-team1
```

```
__replica__: replica1
```

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

Amazon Managed Service for Prometheus는 이러한 레이블과 함께 고가용성 복제본의 데이터 샘플을 저장하므로 샘플이 승인되면 `replica` 레이블이 제거됩니다. 즉, 현재 시리즈에 대해 복제본당 시리즈가 아닌 1:1 시리즈 매핑만 사용할 수 있습니다. `cluster` 레이블은 유지됩니다.

> **Note**
>
> 특정 버전의 Kubernetes (1.28 및 1.29) 는 레이블이 있는 자체 지표를 내보낼 수 있습니다. `cluster` 이로 인해 Prometheus용 Amazon 매니지드 서비스 중복 제거에 문제가 발생할 수 있습니다. 자세한 내용은 고가용성 FAQ를 참조하십시오.

Prometheus Operator를 사용하여 Amazon Managed Service for Prometheus로 고가용성 데이터 전송

Prometheus Operator를 사용하여 고가용성 구성을 설정하려면 Amazon Managed Service for Prometheus에서 식별할 수 있도록 고가용성 그룹의 모든 인스턴스에 외부 레이블을 적용해야 합니다. 또한 Prometheus Operator 차트 Helm에서도 `replicaExternalLabelName` 및 `externalLabels` 속성을 설정해야 합니다.

예제: YAML 헤더

다음 YAML 헤더에서는 Prometheus 인스턴스 에이전트를 고가용성 그룹의 일부로 식별하기 위해 `externalLabel`에 `cluster`가 추가되고 `replicaExternalLabels`는 그룹 내의 각 복제본을 식별합니다.

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

> **Note**
>
> 특정 버전의 쿠버네티스 (1.28 및 1.29) 는 레이블이 있는 자체 메트릭을 내보낼 수 있습니다. `cluster` 이로 인해 Prometheus용 Amazon 매니지드 서비스 중복 제거에 문제가 발생할 수 있습니다. 자세한 내용은 고가용성 FAQ를 참조하십시오.

오픈 텔레메트리용 배포판을 사용하여 AWS Prometheus용 Amazon 매니지드 서비스로 고가용성 데이터를 전송

AWS 오픈 텔레메트리용 배포판 (ADOT) 은 프로젝트를 안전하게 배포하고 프로덕션에 바로 사용할 수 있는 배포판입니다. OpenTelemetry ADOT는 소스 API, 라이브러리 및 에이전트를 제공하므로 애플리케이션 모니터링을 위한 분산 추적 및 지표를 수집할 수 있습니다. ADOT에 대한 자세한 내용은 개방형 텔레메트리용 배포판에 대한 정보를 참조하십시오. AWS

고가용성 구성으로 ADOT를 설정하려면 ADOT 컬렉터 컨테이너 이미지를 구성하고 외부 `cluster` 레이블을 Prometheus AWS 원격 쓰기 `__replica__` 내보내기에 적용해야 합니다. 이 내보내기는 스크래핑한 지표를 `remote_write` 엔드포인트를 통해 Amazon Managed Service for Prometheus WorkSpace로 보냅니다. 원격 쓰기 내보내기에서 이러한 레이블을 설정하면 중복 복제본이 실행되는 동안 중복 지표가 유지되는 것을 방지할 수 있습니다. AWS Prometheus 원격 쓰기 익스포터에 대한 자세한 내용은 Prometheus용 아마존 매니지드 서비스용 Prometheus 원격 쓰기 익스포터 시작하기를 참조하십시오.

> **Note**
>
> 특정 버전의 쿠버네티스 (1.28 및 1.29) 는 레이블이 붙은 자체 메트릭을 내보낼 수 있습니다. `cluster` 이로 인해 Prometheus용 Amazon 매니지드 서비스 중복 제거에 문제가 발생할 수 있습니다. 자세한 내용은 고가용성 FAQ를 참조하십시오.

Prometheus 커뮤니티 Helm 차트를 사용하여 Amazon Managed Service for Prometheus로 고가용성 데이터 전송

Prometheus 커뮤니티 Helm 차트를 사용하여 고가용성 구성을 설정하려면 Amazon Managed Service for Prometheus에서 식별할 수 있도록 고가용성 그룹의 모든 인스턴스에 외부 레이블을 적용해야 합니다. 다음은 Prometheus 커뮤니티 차트 Helm에서 Prometheus의 단일 인스턴스에 `external_labels`를 추가하는 방법의 예입니다.

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

> **Note**
>
> Prometheus 커뮤니티 Helm 차트에서는 컨트롤러 그룹에서 직접 복제본 수를 늘릴 때 복제본 값을 동적으로 설정할 수 없으므로 여러 복제본을 원하는 경우 다른 복제본 값을 사용하여 차트를 여러 번 배포해야 합니다. `replica` 레이블이 자동으로 설정되도록 하려면 prometheus-operator Helm 차트를 사용하세요.

> **Note**
>
> 특정 버전의 쿠버네티스 (1.28 및 1.29) 는 레이블이 있는 자체 메트릭을 내보낼 수 있습니다. `cluster` 이로 인해 Prometheus용 Amazon 매니지드 서비스 중복 제거에 문제가 발생할 수 있습니다. 자세한 내용은 고가용성 FAQ를 참조하십시오.

FAQ: 고가용성 구성

샘플 포인트를 추적하려면 __replica__ 값을 다른 레이블에 포함해야 합니까?

고가용성 설정에서 Amazon Managed Service for Prometheus는 Prometheus 인스턴스 클러스터의 리더를 선택하여 데이터 샘플이 중복되지 않도록 합니다. 리더 복제본이 30초 동안 데이터 샘플 전송을 중단하면 Amazon Managed Service for Prometheus는 자동으로 다른 Prometheus 인스턴스를 리더 복제본으로 만들고 새 리더로부터 누락된 데이터를 비롯한 데이터를 수집합니다. 따라서 대답은 '아니요'로, 이 작업은 권장되지 않습니다. 이렇게 하면 다음과 같은 문제가 발생할 수 있습니다.

- 새 리더를 선택하는 기간 동안 PromQL에서 `count`를 쿼리하면 예상보다 높은 값이 반환될 수 있습니다.
- 새 리더를 선택하는 기간 동안 `active series` 수가 증가하여 `active series limits`에 도달합니다. 자세한 내용은 AMP 할당량을 참조하세요.

Kubernetes에는 자체 클러스터 레이블이 있는 것 같고 메트릭의 중복을 제거하지 않습니다. 이 문제를 해결하려면 어떻게 해야 하나요?

쿠버네티스 1.28에 레이블이 붙은 새 메트릭이 `apiserver_storage_size_bytes` 도입되었습니다. `cluster` 이로 인해 Prometheus용 Amazon Managed Service for Prometheus의 중복 제거에 문제가 발생할 수 있으며, 이 문제는 레이블에 따라 다릅니다. `cluster` Kubernetes 1.3에서는 레이블 이름이 로 변경됩니다 `storage-cluster_id` (이후 패치 1.28 및 1.29에서는 이름도 변경됨). 클러스터가 `cluster` 레이블과 함께 이 지표를 내보내는 경우 Prometheus용 Amazon Managed Service에서는 관련 시계열을 중복 제거할 수 없습니다. 이 문제를 방지하려면 Kubernetes 클러스터를 최신 패치 버전으로 업그레이드하는 것이 좋습니다. 또는, Amazon Managed Service for `cluster` Prometheus에 수집하기 전에 `apiserver_storage_size_bytes` 메트릭의 레이블을 다시 지정할 수도 있습니다.

> **Note**
>
> 쿠버네티스 변경에 대한 자세한 내용은 쿠버네티스 프로젝트의 apiserver_storage_size_bytes 지표에 대한 레이블 클러스터 이름을 storage_cluster_id로 변경을 참조하십시오. GitHub

교차 리전 가용성

데이터에 지역 간 가용성을 추가하려면 AWS 여러 지역의 여러 작업 공간에 메트릭을 전송할 수 있습니다. Prometheus는 다중 작성자와 교차 리전 쓰기를 모두 지원합니다.

다음 예제는 에이전트 모드에서 실행되는 Prometheus 서버가 Helm을 사용하여 서로 다른 리전의 두 WorkSpace에 지표를 보내도록 설정하는 방법을 보여 줍니다.

```
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]
```

Prometheus 지표 쿼리

이제 지표가 WorkSpace에 수집되었으므로 쿼리할 수 있습니다. Grafana와 같은 서비스를 사용하여 지표를 쿼리하거나 Amazon Managed Service for Prometheus API를 사용할 수 있습니다.

쿼리는 표준 Prometheus 쿼리 언어인 PromQL을 사용하여 수행합니다. PromQL 및 해당 구문에 대한 자세한 내용은 Prometheus 설명서의 Prometheus 쿼리를 참조하십시오.

주제

지표 쿼리 보호

Amazon Managed Service for Prometheus는 지표 쿼리를 보호하는 데 도움이 되는 방법을 제공합니다.

Amazon Managed Service for Prometheus에서 AWS PrivateLink 사용

Amazon Managed Service for Prometheus에서 지표를 쿼리하는 네트워크 트래픽은 퍼블릭 인터넷 엔드포인트를 통해 또는 AWS PrivateLink를 통해 VPC 엔드포인트에 의해 수행할 수 있습니다. AWS PrivateLink를 사용하면 VPC의 네트워크 트래픽을 퍼블릭 인터넷을 거치지 않고 AWS 네트워크 내에서 보호할 수 있습니다. Amazon Managed Service for Prometheus용 AWS PrivateLink VPC 엔드포인트를 생성하려면 인터페이스 VPC 엔드포인트에서 Amazon Managed Service for Prometheus 사용 섹션을 참조하십시오.

인증 및 권한 부여

AWS Identity and Access Management는 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있는 웹 서비스입니다. IAM을 사용하여 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 있음)된 대상

을 제어합니다. Amazon Managed Service for Prometheus는 IAM과 통합되어 데이터를 안전하게 유지하는 데 도움이 됩니다. Amazon Managed Service for Prometheus를 설정할 때는 Grafana 서버가 Amazon Managed Service for Prometheus WorkSpace에 저장된 지표를 쿼리할 수 있도록 하는 몇 가지 IAM 역할을 생성해야 합니다. IAM에 대한 자세한 내용은 IAM이란? 섹션을 참조하십시오.

Amazon Managed Service for Prometheus를 설정하는 데 도움이 되는 또 다른 AWS 보안 기능은 AWS 서명 버전 4 서명 프로세스(AWS SigV4)입니다. 서명 버전 4는 HTTP로 전송된 AWS 요청에 인증 정보를 추가하는 프로세스입니다. 보안을 위해 대부분의 AWS 요청은 액세스 키 ID와 보안 액세스 키로 구성된 액세스 키로 서명해야 합니다. 이 두 키는 일반적으로 보안 자격 증명이라고 합니다. SigV4에 대한 자세한 내용은 서명 버전 4 서명 프로세스를 참조하십시오.

Amazon Managed Grafana를 Amazon Managed Service for Prometheus와 함께 사용하도록 설정

Amazon Managed Grafana는 오픈 소스 Grafana용 완전 관리형 서비스로 대규모로 데이터 소스를 시각화하고 분석할 수 있는 오픈 소스, 타사 ISV 및 AWS 서비스에 대한 연결을 간소화합니다.

Amazon Managed Service for Prometheus에서는 Amazon Managed Grafana를 사용하여 WorkSpace에서 지표를 쿼리할 수 있습니다. Amazon Managed Grafana 콘솔에서 기존 Amazon Managed Service for Prometheus 계정을 검색하여 Amazon Managed Service for Prometheus WorkSpace를 데이터 소스로 추가할 수 있습니다. Amazon Managed Grafana는 Amazon Managed Service for Prometheus에 액세스하는 데 필요한 인증 자격 증명의 구성을 관리합니다. Amazon Managed Grafana에서 Amazon Managed Service for Prometheus에 대한 연결을 생성하는 방법에 대한 자세한 지침은 Amazon Managed Grafana 사용 설명서의 지침을 참조하십시오.

Amazon Managed Grafana에서 Amazon Managed Service for Prometheus 알림을 확인할 수도 있습니다. 알림과의 통합을 설정하는 방법에 대한 지침은 Amazon Managed Grafana 또는 오픈 소스 Grafana와 알림 통합 섹션을 참조하십시오.

프라이빗 VPC에서 Amazon Managed Grafana에 연결

Amazon Managed Service for Prometheus는 Amazon Managed Grafana가 지표 및 알림을 쿼리할 때 연결할 수 있는 서비스 엔드포인트를 제공합니다.

프라이빗 VPC를 사용하도록 Amazon Managed Grafana를 구성할 수 있습니다(Grafana에서 프라이빗 VPC를 설정하는 방법에 대한 자세한 내용은 Amazon Managed Grafana 사용 설명서의 Amazon VPC에 연결 참조). 설정에 따라 이 VPC는 Amazon Managed Service for Prometheus 서비스 엔드포인트에 액세스하지 못할 수도 있습니다.

특정 프라이빗 VPC를 사용하도록 구성된 Amazon Managed Grafana WorkSpace에 Amazon Managed Service for Prometheus를 데이터 소스로 추가하려면 먼저 VPC 엔드포인트를 생성하여 Amazon Managed Service for Prometheus를 동일한 VPC에 연결해야 합니다. VPC 엔드포인트 생성에 대한 자세한 내용은 Amazon Managed Service for Prometheus에 대한 인터페이스 VPC 엔드포인트 생성 섹션을 참조하십시오.

Amazon Managed Service for Prometheus와 함께 사용할 Grafana 오픈 소스 또는 Grafana Enterprise를 설정하십시오.

Amazon Managed Service for Prometheus에서는 Grafana 버전 7.3.5 이상을 사용하여 WorkSpace에서 지표를 쿼리할 수 있습니다. 버전 7.3.5 이상에는 AWS 서명 버전 4(SigV4) 인증에 대한 지원이 포함됩니다.

tar.gz 또는 zip 파일을 사용하여 독립형 Grafana를 설정하는 방법에 대한 지침은 Grafana 설명서에서 Grafana 설치를 참조하십시오. 새 독립형 Grafana를 설치하면 사용자 이름과 암호를 입력하라는 메시지가 표시됩니다. 기본값은 **admin/admin**입니다. 처음 로그인하면 암호를 변경하라는 메시지가 표시됩니다. 자세한 내용은 Grafana 설명서에서 Grafana 시작하기를 참조하십시오.

Grafana 버전을 확인하려면 다음 명령을 실행합니다.

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

Prometheus용 Amazon 관리 서비스와 함께 작동하도록 Grafana를 설정하려면 정책 또는,, 및 권한이 AmazonPrometheusQueryAccess있는 계정에 로그인해야 합니다. `aps:QueryMetrics` `aps:GetMetricMetadata` `aps:GetSeries` `aps:GetLabels` 자세한 설명은 IAM 권한 및 정책 섹션을 참조하세요.

AWS SIGv4 설정

Amazon Managed Service for Prometheus는 AWS Identity and Access Management(IAM)과 연동하여 Prometheus API에 대한 모든 호출을 IAM 자격 증명으로 보호합니다. 기본적으로 Grafana의 Prometheus 데이터 소스는 Prometheus에 인증이 필요하지 않다고 가정합니다. Grafana가 Amazon Managed Service for Prometheus 인증 및 권한 부여 기능을 활용할 수 있도록 하려면 Grafana 데이터 소스에서 SigV4 인증 지원을 활성화해야 합니다. 자체 관리형 Grafana 오픈 소스 또는 Grafana 엔터프라이즈 서버를 사용하는 경우 이 페이지의 단계를 따르십시오. Amazon Managed Grafana를 사용하는 경우 SigV4 인증은 완전히 자동화됩니다. Amazon Managed Grafana에 대한 자세한 내용은 Amazon Managed Grafana란 무엇입니까?를 참조하십시오.

Grafana에서 SigV4를 활성화하려면 `AWS_SDK_LOAD_CONFIG` 및 `GF_AUTH_SIGV4_AUTH_ENABLED` 환경 변수를 `true`로 설정한 상태에서 Grafana를 시작하십시오. `GF_AUTH_SIGV4_AUTH_ENABLED` 환경 변수는 SigV4 지원을 활성화하기 위해 Grafana의 기본 구성을 재정의합니다. 자세한 내용은 Grafana 설명서의 구성을 참조하십시오.

Linux

Linux의 독립형 Grafana 서버에서 SigV4를 활성화하려면 다음 명령을 입력합니다.

```
export AWS_SDK_LOAD_CONFIG=true
```

```
export GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

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

Windows

Windows 명령 프롬프트를 사용하여 Windows의 독립형 Grafana에서 SigV4를 활성화하려면 다음 명령을 입력합니다.

```
set AWS_SDK_LOAD_CONFIG=true
```

```
set GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

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

Grafana에 Prometheus 데이터 소스 추가

다음 단계는 Amazon Managed Service for Prometheus 지표를 쿼리하도록 Grafana의 Prometheus 데이터 소스를 설정하는 방법을 설명합니다.

Grafana 서버에 Prometheus 데이터 소스를 추가하려면

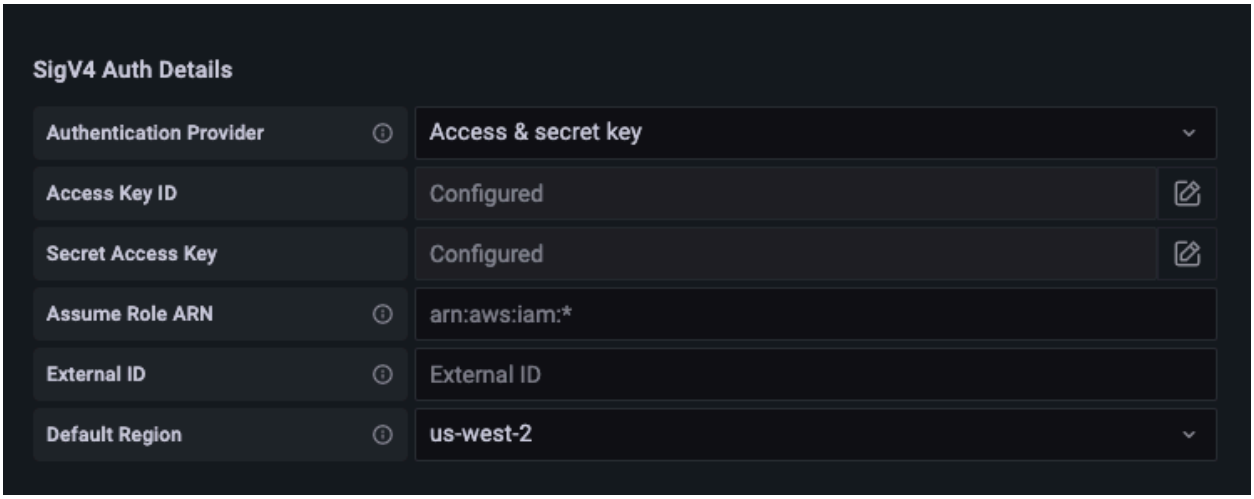
1. Grafana 콘솔을 엽니다.

2. 구성에서 데이터 소스를 선택합니다.
3. 데이터 소스 추가를 선택합니다.
4. Prometheus를 선택합니다.
5. HTTP URL의 경우 Amazon Managed Service for Prometheus 콘솔의 WorkSpace 세부 정보 페이지에 표시된 엔드포인트 - 쿼리 URL을 지정합니다.
6. 방금 지정한 HTTP URL에서 URL에 추가된 `/api/v1/query` 문자열을 제거합니다. Prometheus 데이터 소스에서 자동으로 추가하기 때문입니다.

   올바른 URL은 https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-1234a5b6-78cd-901e-2fgh-3i45j6k178l9와 비슷합니다.
7. 인증에서 SigV4 인증 토글을 선택하여 활성화합니다.
8. Grafana에서 직접 장기 자격 증명을 지정하거나 기본 공급자 체인을 사용하여 SigV4 인증을 구성할 수 있습니다. 장기 자격 증명을 직접 지정하면 더 빨리 시작할 수 있으며, 다음 단계에서는 이러한 지침이 먼저 제공됩니다. Amazon Managed Service for Prometheus에서 Grafana를 사용하는 데 익숙해지면 더 나은 유연성과 보안을 제공하는 기본 공급자 체인을 사용하는 것이 좋습니다. 기본 제공자 체인 설정에 대한 자세한 내용은 자격 증명 지정을 참조하십시오.
   - 장기 자격 증명을 직접 사용하려면 다음을 수행합니다.
     a. SigV4 인증 세부 정보에서 인증 공급자에 대해 액세스 및 보안 키를 선택합니다.
     b. 액세스 키 ID에 AWS 액세스 키 ID를 입력합니다.
     c. 보안 액세스 키에 AWS 보안 액세스 키를 입력합니다.
     d. 역할 수임 ARN 및 외부 ID 필드를 비워 둡니다.
     e. 기본 리전에 대해 Amazon Managed Service for Prometheus WorkSpace의 리전을 선택합니다. 이 리전은 5단계에서 나열한 URL에 포함된 리전과 일치해야 합니다.
     f. 저장 및 테스트를 선택합니다.

        데이터 소스가 작동 중입니다. 메시지가 표시됩니다.

        다음 스크린샷은 액세스 키, 보안 키 SigV4 인증 세부 정보 설정을 보여 줍니다.



- 기본 공급자 체인을 대신 사용하려면(프로덕션 환경에 권장), 다음을 수행하십시오.

  a. SigV4 인증 세부 정보에서 인증 공급자로 AWS SDK 기본값을 선택합니다.

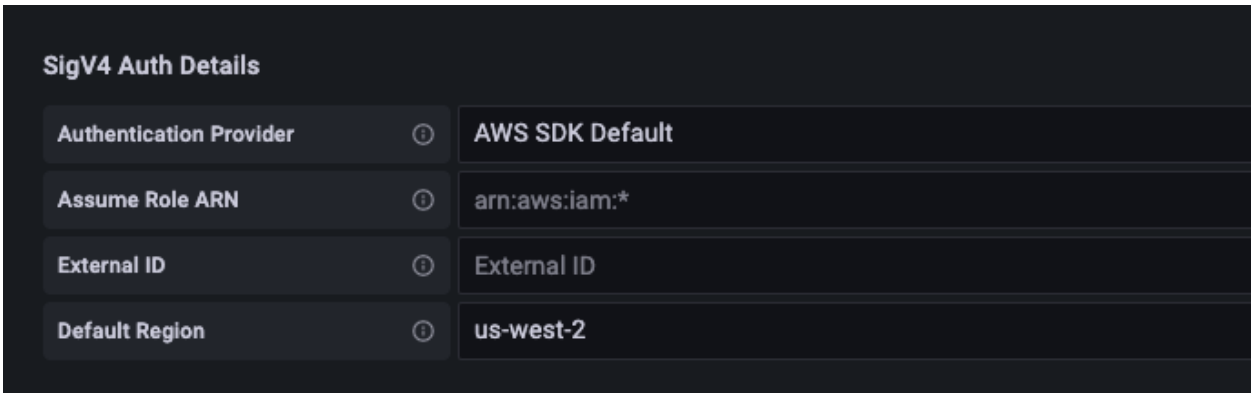
  b. 역할 수임 ARN 및 외부 ID 필드를 비워 둡니다.

  c. 기본 리전에 대해 Amazon Managed Service for Prometheus WorkSpace의 리전을 선택합니다. 이 리전은 5단계에서 나열한 URL에 포함된 리전과 일치해야 합니다.

  d. 저장 및 테스트를 선택합니다.

  데이터 소스가 작동 중입니다. 메시지가 표시됩니다.

  다음 스크린샷은 SDK 기본 SigV4 인증 세부 정보 설정을 보여 줍니다.



9. 새 데이터 소스에 대해 PromQL 쿼리를 테스트합니다.

   a. 탐색을 선택합니다.

   b. 다음과 같은 샘플 PromQL 쿼리를 실행합니다.

```
prometheus_tsdb_head_series
```

저장 및 테스트가 작동하지 않는 경우 문제 해결

이전 절차에서 저장 및 테스트를 선택할 때 오류가 표시되면 다음을 확인하십시오.

HTTP 오류 찾을 수 없음

URL의 WorkSpace ID가 올바른지 확인합니다.

HTTP 오류 금지

이 오류는 자격 증명이 유효하지 않음을 의미합니다. 다음을 확인하십시오.

- 기본 리전에 지정된 리전이 올바른지 확인합니다.
- 자격 증명에 입력 오류가 있는지 확인합니다.
- 사용 중인 자격 증명에 정책이 있는지 확인하십시오. AmazonPrometheusQueryAccess 자세한 설명은 IAM 권한 및 정책 섹션을 참조하세요.
- 사용 중인 자격 증명에 이 Amazon Managed Service for Prometheus WorkSpace에 대한 액세스 권한이 있는지 확인합니다.

HTTP 오류 잘못된 게이트웨이

이 오류를 해결하려면 Grafana 서버 로그를 확인하십시오. 자세한 내용은 Grafana 설명서에서 문제 해결을 참조하십시오.

`Error http: proxy error: NoCredentialProviders: no valid providers in chain`이 표시되면 기본 자격 증명 공급자 체인이 사용할 유효한 AWS 자격 증명을 찾지 못한 것입니다. 자격 증명 지정에 설명된 대로 자격 증명을 설정했는지 확인합니다. 공유 구성을 사용하려면 `AWS_SDK_LOAD_CONFIG` 환경이 `true`로 설정되어 있는지 확인합니다.

Amazon EKS 클러스터에서 실행 중인 Grafana를 사용한 쿼리

Amazon Managed Service for Prometheus에서는 Grafana 버전 7.3.5 이상을 사용하여 Amazon Managed Service for Prometheus WorkSpace에서 지표를 쿼리할 수 있습니다. 버전 7.3.5 이상에는 AWS 서명 버전 4(SigV4) 인증에 대한 지원이 포함됩니다.

Prometheus용 Amazon 관리 서비스와 함께 작동하도록 Grafana를 설정하려면 정책 또는,, 및 권한이 AmazonPrometheusQueryAccess있는 계정에 로그인해야 합니다. `aps:QueryMetrics`

`aps:GetMetricMetadata` `aps:GetSeries` `aps:GetLabels` 자세한 설명은 IAM 권한 및 정책 섹션을 참조하세요.

AWS SIGv4 설정

Grafana는 AWS 서명 버전 4(SigV4) 인증을 지원하는 새로운 기능을 추가했습니다. 자세한 내용은 서명 버전 4 서명 프로세스를 참조하십시오. Grafana 서버에서는 이 기능이 기본적으로 비활성화되어 있습니다. 이 기능을 활성화하기 위한 다음 지침은 Helm을 사용하여 Kubernetes 클러스터에 Grafana를 배포한다고 가정합니다.

Grafana 7.3.5 이상 버전의 서버에서 SigV4를 활성화하려면

1. Grafana 구성을 재정의하는 새 업데이트 파일을 만들고 이름을 `amp_query_override_values.yaml`로 지정합니다.
2. 다음 콘텐츠를 복사하고 파일에 입력한 후 파일을 저장합니다. *`account-id`*를 Grafana 서버가 실행 중인 AWS 계정 ID로 바꿉니다.

```
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
```

   해당 YAML 파일 내용에서 `amp-iamproxy-query-role`은 다음 섹션 서비스 계정에 대한 IAM 역할 설정에서 생성할 역할의 이름입니다. WorkSpace 쿼리를 위한 역할을 이미 생성한 경우 이 역할을 자체 역할 이름으로 바꿀 수 있습니다.

이 파일은 나중에 Helm을 사용하여 Grafana 서버 업그레이드에서 사용합니다.

서비스 계정에 대한 IAM 역할 설정

Amazon EKS 클러스터에서 Grafana 서버를 사용하는 경우 액세스 제어를 위해 서비스 계정의 IAM 역할(서비스 역할이라고도 함)을 사용하는 것이 좋습니다. 이렇게 하면 IAM 역할을 Kubernetes 서비스 계정과 연결하면 서비스 계정에서는 해당 서비스 계정을 사용하는 모든 포드에 있는 컨테이너에 대해 AWS 권한을 제공할 수 있습니다. 자세한 내용은 서비스 계정에 대한 IAM 역할을 참조하십시오.

쿼리를 위해 이러한 서비스 역할을 아직 설정하지 않은 경우 지표 쿼리를 위해 서비스 계정에 대한 IAM 역할 설정의 지침에 따라 역할을 설정하십시오.

그런 다음, 신뢰 관계 조건에 Grafana 서비스 계정을 추가해야 합니다.

신뢰 관계 조건에 Grafana 서비스 계정을 추가하려면

1. 터미널 창에서 Grafana 서버의 네임스페이스와 서비스 계정 이름을 확인합니다. 예를 들어, 다음 명령을 사용할 수 있습니다.

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

2. Amazon EKS 콘솔에서 EKS 클러스터와 연결된 서비스 계정의 IAM 역할을 엽니다.
3. 신뢰 관계 편집을 선택합니다.
4. 1단계의 명령 출력에서 찾은 Grafana 네임스페이스와 Grafana 서비스 계정 이름을 포함하도록 조건을 업데이트합니다. 다음은 예입니다.

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

5. 신뢰 정책 업데이트를 선택합니다.

Helm을 사용하여 Grafana 서버 업그레이드

이 단계는 이전 섹션에서 `amp_query_override_values.yaml` 파일에 추가한 항목을 사용하도록 Grafana 서버를 업그레이드합니다.

다음 명령을 실행합니다. Grafana용 Helm 차트에 대한 자세한 내용은 Grafana 커뮤니티 Kubernetes Helm 차트를 참조하십시오.

```
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
```

Grafana에 Prometheus 데이터 소스 추가

다음 단계는 Amazon Managed Service for Prometheus 지표를 쿼리하도록 Grafana의 Prometheus 데이터 소스를 설정하는 방법을 설명합니다.

Grafana 서버에 Prometheus 데이터 소스를 추가하려면

1. Grafana 콘솔을 엽니다.
2. 구성에서 데이터 소스를 선택합니다.
3. 데이터 소스 추가를 선택합니다.
4. Prometheus를 선택합니다.
5. HTTP URL의 경우 Amazon Managed Service for Prometheus 콘솔의 WorkSpace 세부 정보 페이지에 표시된 엔드포인트 - 쿼리 URL을 지정합니다.
6. 방금 지정한 HTTP URL에서 URL에 추가된 `/api/v1/query` 문자열을 제거합니다. Prometheus 데이터 소스에서 자동으로 추가하기 때문입니다.
7. 인증에서 SigV4 인증 토글을 선택하여 활성화합니다.

   역할 수임 ARN 및 외부 ID 필드를 비워 둡니다. 그런 다음, 기본 리전으로 Amazon Managed Service for Prometheus WorkSpace가 있는 리전을 선택합니다.
8. 저장 및 테스트를 선택합니다.

   데이터 소스가 작동 중입니다. 메시지가 표시됩니다.
9. 새 데이터 소스에 대해 PromQL 쿼리를 테스트합니다.

a. 탐색을 선택합니다.

b. 다음과 같은 샘플 PromQL 쿼리를 실행합니다.

```
prometheus_tsdb_head_series
```

Prometheus 호환 API를 사용한 쿼리

Amazon Managed Grafana와 같은 도구를 사용하는 것이 지표를 보고 쿼리하는 가장 쉬운 방법이지만, Amazon Managed Service for Prometheus는 지표를 쿼리하는 데 사용할 수 있는 몇 가지 Prometheus 호환 API도 지원합니다. 사용 가능한 Prometheus 호환 API에 대한 자세한 내용은 Prometheus 호환 API 섹션을 참조하십시오.

이러한 API를 사용하여 지표를 쿼리할 때는 AWS 서명 버전 4 서명 프로세스로 요청에 서명해야 합니다. AWS 서명 버전 4를 설정하여 서명 프로세스를 간소화할 수 있습니다. 자세한 내용은 aws-sigv4-proxy를 참조하십시오.

`awscurl`을 사용하여 AWS SigV4 프록시를 통한 서명을 수행할 수 있습니다. 다음 항목 awscurl을 사용하여 Prometheus 호환 API 쿼리에서는 `awscurl`을 사용하여 AWS SigV4를 설정하는 방법을 설명합니다.

awscurl을 사용하여 Prometheus 호환 API를 쿼리

Amazon Managed Service for Prometheus에 대한 API 요청은 SigV4로 서명해야 합니다. awscurl을 사용하여 쿼리 프로세스를 간소화할 수 있습니다.

`awscurl`을 설치하려면 Python 3와 pip 패키지 관리자가 설치되어 있어야 합니다.

Linux 기반 인스턴스에서는 다음 명령이 `awscurl`을 설치합니다.

```
$ pip3 install awscurl
```

macOS 시스템에서는 다음 명령이 `awscurl`을 설치합니다.

```
$ brew install awscurl
```

다음 예는 샘플 쿼리입니다. `awscurl` *##*, *`Workspace-ID`* 및 *`QUERY`* 입력을 사용 사례에 적합한 값으로 바꾸십시오.

```
# 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**
>
> 쿼리 문자열은 url로 인코딩되어야 합니다.

다음과 같은 query=up 쿼리의 경우 다음과 같은 결과를 얻을 수 있습니다.

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

awscurl이 제공된 요청에 서명하도록 하려면 다음 방법 중 하나로 유효한 자격 증명을 전달해야 합니다.

- IAM 역할의 액세스 키 ID와 보안 키를 제공합니다. https://console.aws.amazon.com/iam/에서 역할에 대한 액세스 키와 보안 키를 찾을 수 있습니다.

  예:

```
$ 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>"
```

- .aws/credentials 및 /aws/config 파일에 저장된 구성 파일을 참조합니다. 또한 사용할 프로파일의 이름을 지정하도록 선택할 수 있습니다. 지정하지 않으면 default 파일이 사용됩니다. 예:

```
$ 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>"
```

- EC2 인스턴스와 연결된 인스턴스 프로파일을 사용합니다.

awscurl 컨테이너를 사용하여 쿼리 요청 실행

다른 버전의 Python을 설치하는데 관련 종속성을 실행할 수 없는 경우 컨테이너를 사용하여 awscurl 애플리케이션과 해당 종속성을 패키징할 수 있습니다. 다음 예제에서는 Docker 런타임을 사용하여 awscurl을 배포하지만 OCI 호환 런타임과 이미지가 모두 잘 작동합니다.

```
$ 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"
```

쿼리 API 응답의 쿼리 통계 정보

쿼리 요금은 실행된 쿼리에서 한 달 동안 처리된 총 쿼리 샘플 수를 기준으로 합니다. query 또는 queryRange API에 대한 쿼리 응답에는 처리된 쿼리 샘플에 대한 통계 데이터가 포함됩니다. 요청에서 쿼리 파라미터 stats=all이 전송되면 samples 객체가 stats 객체가 생성되고 응답으로 stats 데이터가 반환됩니다.

samples 객체는 다음 속성으로 구성됩니다.

속성	설명
totalQueryableSamples	처리된 쿼리 샘플의 총 수입니다. 청구에 사용할 정보입니다.
totalQueryableSamplesPerStep	각 단계에서 처리된 쿼리 샘플 수입니다. 이는 Epoch 단위의 타임스탬프와 특정 단계에서 로드된 샘플 수를 포함하는 배열로 구성된 구조입니다.

응답에 stats 정보가 포함된 샘플 요청 및 응답은 다음과 같습니다.

query 예제:

GET

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

응답

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

queryRange 예제:

GET

```
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
```

응답

```
{
```

```
    "status": "success",
    "data": {
        "resultType": "matrix",
        "result": [
            {
                "metric": {},
                "values": [
                    [
                        1652383000,
                        "0"
                    ],
                    [
                        1652384000,
                        "0"
                    ]
                ]
            }
        ],
        "stats": {
            "samples": {
                "totalQueryableSamples": 8,
                "totalQueryableSamplesPerStep": [
                    [
                        1652382000,
                        0
                    ],
                    [
                        1652383000,
                        4
                    ],
                    [
                        1652384000,
                        4
                    ]
                ]
            }
        }
    }
}
```

기록 규칙 및 알림 규칙

Amazon Managed Service for Prometheus는 정기적으로 평가하는 두 가지 유형의 규칙을 지원합니다.

- 기록 규칙을 사용하면 자주 필요하거나 계산 비용이 많이 드는 식을 미리 계산하고, 해당 결과를 새로운 시계열 세트로 저장할 수 있습니다. 미리 계산된 결과를 쿼리하는 것이 필요할 때마다 원래 식을 실행하는 것보다 훨씬 빠른 경우가 많습니다.
- 알림 규칙을 사용하면 PromQL 및 임곗값을 기준으로 알림 조건을 정의할 수 있습니다. 규칙이 임곗값을 트리거하면 알림 관리자에게 알림이 전송되고 알림 관리자는 이 알림을 다운스트림으로 Amazon Simple Notification Service 등의 수신기에 전달합니다.

Amazon Managed Service for Prometheus에서 규칙을 사용하려면 규칙을 정의하는 하나 이상의 YAML 규칙 파일을 생성합니다. Amazon Managed Service for Prometheus 규칙 파일은 독립형 Prometheus의 규칙 파일과 형식이 동일합니다. 자세한 내용은 Prometheus 설명서의 기록 규칙 정의 및 알림 규칙을 참조하세요.

하나의 워크스페이스에 여러 규칙 파일을 둘 수 있습니다. 각각의 개별 규칙 파일은 별도의 네임스페이스 내에 포함됩니다. 규칙 파일이 여러 개 있으면 기존 Prometheus 규칙 파일을 변경하거나 결합하지 않고도 워크스페이스로 가져올 수 있습니다. 규칙 그룹 네임스페이스마다 태그가 다를 수도 있습니다.

규칙 시퀀싱

규칙 파일 내에서 규칙은 규칙 그룹 내에 포함됩니다. 규칙 파일의 단일 규칙 그룹 내 규칙은 항상 위에서 아래로 평가됩니다. 따라서 기록 규칙에서 하나의 기록 규칙 결과를 이후 기록 규칙을 계산할 때 사용하거나 동일한 규칙 그룹의 알림 규칙에 사용할 수 있습니다. 하지만 별도의 규칙 파일을 실행하는 순서는 지정할 수 없으므로 한 기록 규칙의 결과를 사용하여 다른 규칙 그룹이나 다른 규칙 파일의 규칙을 계산할 수는 없습니다.

주제

- 필요한 IAM 권한
- 규칙 파일 생성
- Amazon Managed Service for Prometheus에 규칙 구성 파일 업로드
- 규칙 구성 파일 편집
- 규칙 관리자 문제 해결

필요한 IAM 권한

Amazon Managed Service for Prometheus에서 사용자에게 규칙을 사용할 수 있는 권한을 부여해야 합니다. 다음 권한으로 AWS Identity and Access Management(IAM) 정책을 생성하고 사용자, 그룹 또는 역할에 정책을 할당합니다.

> **Note**
>
> IAM에 대한 자세한 내용은 Amazon Managed Service for Prometheus용 Identity and Access Management 단원을 참조하십시오.

사용 규칙에 대한 액세스 권한을 부여하는 정책

다음 정책은 계정의 모든 리소스에 대한 규칙을 사용하기 위한 액세스 권한을 부여합니다.

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

한 네임스페이스에만 액세스 권한을 부여하는 정책

특정 정책에 대해서만 액세스 권한을 부여하는 정책을 생성할 수도 있습니다. 다음 샘플 정책은 지정된 RuleGroupNamespace에 대해서만 액세스 권한을 부여합니다. 이 정책을 사용하려면, *<account>*, *<region>*, *<workspace-id>* 및 *<namespace-name>*을 계정에 적합한 값으로 바꿉니다.

```
{
```

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

규칙 파일 생성

Amazon Managed Service for Prometheus에서 규칙을 사용하려면 규칙을 정의하는 규칙 파일을 생성합니다. Amazon Managed Service for Prometheus 규칙 파일은 독립형 Prometheus의 규칙 파일과 형식이 동일합니다. 자세한 내용을 알아보려면 기록 규칙 정의 및 알림 규칙을 참조하세요.

다음은 규칙 파일의 기본 예제입니다.

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

알림 규칙 예제에 대한 자세한 내용은 알림 규칙 예제를 참조하세요.

Amazon Managed Service for Prometheus에 규칙 구성 파일 업로드

이제 Amazon Managed Service for Prometheus에 이 규칙 구성 파일을 업로드해야 합니다. 콘솔 또는 AWS CLI를 사용하여 로드할 수 있습니다.

Amazon Managed Service for Prometheus 콘솔을 사용하여 규칙 구성을 업로드하고 네임스페이스를 생성하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 페이지 왼쪽 상단에서 메뉴 아이콘을 선택한 다음, 모든 워크스페이스를 선택합니다.
3. 워크스페이스의 워크스페이스 ID를 선택한 다음, 규칙 관리 탭을 선택합니다.
4. 네임스페이스 추가를 선택합니다.
5. 파일 선택을 선택하고 규칙 정의 파일을 선택합니다.
6. (선택 사항) 네임스페이스에 태그를 추가하려면 새 태그 추가를 선택합니다.

   그런 다음, 키(Key)에서 태그 이름을 입력합니다. 값(Value)에 태그의 선택적 값을 추가할 수 있습니다.

   다른 태그를 추가하려면 새 태그 추가를 선택합니다.
7. 계속을 선택합니다. Amazon Managed Service for Prometheus는 선택한 규칙 파일과 동일한 이름을 가진 새 네임스페이스를 생성합니다.

AWS CLI를 사용하여 알림 관리자 구성을 새 네임스페이스의 워크스페이스에 업로드하려면

1. Base64는 알림 관리자 파일의 내용을 인코딩합니다. Linux에서 다음 명령을 사용할 수 있습니다.

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

   macOS에서 다음 명령을 사용할 수 있습니다.

   ```
   openssl base64 input-file output-file
   ```
2. 다음 명령 중 하나를 입력하여 네임스페이스를 생성하고 파일을 업로드합니다.

   AWS CLI 버전 2에서는 다음을 입력합니다.

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

AWS CLI 버전 1에서는 다음을 입력합니다.

```
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. 알림 관리자 구성이 활성화되는 데 몇 초 정도 걸립니다. 상태를 확인하려면 다음 명령을 입력합니다.

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

`status`가 `ACTIVE`이면 규칙 파일이 적용된 것입니다.

규칙 구성 파일 편집

콘솔에서 직접 규칙 구성 파일을 편집할 수는 없습니다. 대신 새 규칙 파일을 업로드하여 대체합니다. 선택적으로 현재 파일을 다운로드하고 텍스트 편집기에서 편집한 다음, 새 버전을 업로드할 수 있습니다.

Amazon Managed Service for Prometheus 콘솔을 사용하여 규칙 구성을 편집하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 페이지 왼쪽 상단에서 메뉴 아이콘을 선택한 다음, 모든 워크스페이스를 선택합니다.
3. 워크스페이스의 워크스페이스 ID를 선택한 다음, 규칙 관리 탭을 선택합니다.
4. (선택 사항) 먼저 현재 규칙 구성 파일을 편집하려면 다운로드 또는 복사를 선택합니다.
5. 새 규칙 파일이 준비되면 바꾸기를 선택합니다.
6. 파일 선택을 선택하고 새 규칙 정의 파일을 선택한 다음, 계속을 선택합니다.

AWS CLI를 사용하여 규칙 구성 파일을 편집하려면

1. Base64는 규칙 파일의 내용을 인코딩합니다. Linux에서 다음 명령을 사용할 수 있습니다.

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

macOS에서 다음 명령을 사용할 수 있습니다.

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

2. 다음 명령 중 하나를 입력하여 새 파일을 업로드합니다.

   AWS CLI 버전 2에서는 다음을 입력합니다.

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

   AWS CLI 버전 1에서는 다음을 입력합니다.

```
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. 규칙 파일이 활성화되는 데 몇 초 정도 걸립니다. 상태를 확인하려면 다음 명령을 입력합니다.

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

   `status`가 `ACTIVE`이면 규칙 파일이 적용된 것입니다. 그때까지는 이 규칙 파일의 이전 버전이 계속 활성 상태입니다.

규칙 관리자 문제 해결

CloudWatch 로그을 사용하여 알림 관리자 및 눈금자 관련 문제를 해결할 수 있습니다. 이 섹션에는 규칙 관리자 관련 문제 해결 항목이 포함되어 있습니다.

로그에 다음과 같은 규칙 관리자 실패 오류가 포함된 경우

```
{
    "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"
}
```

규칙을 실행하는 동안 오류가 발생했음을 의미합니다.

취할 조치

오류 메시지를 사용하여 규칙 실행 문제를 해결합니다.

알림 관리자

Amazon Managed Service for Prometheus에서 실행하는 알림 규칙이 발생하면 알림 관리자가 전송된 알림을 처리합니다. 알림을 중복 제거하고 그룹화하여 다운스트림 수신기에 라우팅합니다. Amazon Managed Service for Prometheus는 Amazon Simple Notification Service만 수신기로 지원하며, 동일한 계정의 Amazon SNS 주제로 메시지를 라우팅할 수 있습니다. 알림 관리자를 사용하여 알림을 무음으로 설정하거나 금지할 수도 있습니다.

알림 관리자는 Prometheus의 Alertmanager와 유사한 기능을 제공합니다.

알림 관리자의 구성 파일은 다음을 위해 사용할 수 있습니다.

- 그룹화 - 그룹화하면 유사한 알림을 하나의 알림으로 모을 수 있습니다. 이 기능은 한 번에 많은 시스템에 장애가 발생하고 수백 개의 알림이 동시에 발생할 수 있는 대규모 장애 발생 시 특히 유용합니다. 예를 들어 네트워크 장애로 인해 많은 노드에 동시에 장애가 발생한다고 가정해 보겠습니다. 이러한 유형의 알림이 그룹화되어 있으면 알림 관리자가 단일 알림을 보냅니다.

  알림 그룹화 및 그룹화된 알림의 타이밍은 알림 관리자 구성 파일의 라우팅 트리로 구성됩니다. 자세한 내용은 <route>를 참조하십시오.
- 금지 - 금지는 다른 특정 알림이 이미 발신되고 있는 경우 특정 알림에 대한 알림을 억제합니다. 예를 들어 클러스터에 연결할 수 없다는 알림이 발생하는 경우 이 클러스터와 관련된 다른 모든 알림을 음소거하도록 알림 관리자를 구성할 수 있습니다. 이렇게 하면 실제 문제와 관련이 없는 수백 개 또는 수천 개의 알림을 방지할 수 있습니다. 금지 규칙을 작성하는 방법에 대한 자세한 내용은 <inhibit_rule>을 참조하십시오.
- 무음 - 무음은 지정된 시간 동안(예: 유지 관리 기간 동안) 알림을 음소거합니다. 수신되는 알림이 활성 무음의 모든 등식 또는 정규식 매처와 일치하는지 확인됩니다. 일치하는 경우 해당 알림에 대한 메시지가 전송되지 않습니다.

  무음을 만들려면 `PutAlertManagerSilences` API를 사용합니다. 자세한 내용은 PutAlertManagerSilences 섹션을 참조하십시오.

Prometheus 템플릿

독립형 Prometheus는 별도의 템플릿 파일을 사용하여 템플릿 작성을 지원합니다. 템플릿은 무엇보다도 조건문 및 형식 데이터를 사용할 수 있습니다.

Amazon Managed Service for Prometheus에서는 알림 관리자 구성과 동일한 알림 관리자 구성 파일에 템플릿을 배치합니다.

주제

필요한 IAM 권한

Amazon Managed Service for Prometheus에서 사용자에게 규칙을 사용할 수 있는 권한을 부여해야 합니다. 다음 권한으로 AWS Identity and Access Management(IAM) 정책을 생성하고 사용자, 그룹 또는 역할에 정책을 할당합니다.

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

```
    ]
}
```

알림 관리자 구성 파일 생성

Amazon Managed Service for Prometheus에서 알림 관리자 및 템플릿을 사용하려면 알림 관리자 구성 YAML 파일을 생성합니다. Amazon Managed Service for Prometheus 알림 관리자 파일에는 다음과 같은 두 가지 기본 섹션이 있습니다.

- `template_files:`에는 수신기에서 보내는 메시지에 사용되는 템플릿이 들어 있습니다. 자세한 내용은 Prometheus 설명서의 템플릿 참조 및 템플릿 예제를 참조하십시오.
- `alertmanager_config:`에는 알림 관리자 구성이 포함되어 있습니다. 이 섹션에서는 독립형 Prometheus의 알림 관리자 구성 파일과 동일한 구조를 사용합니다. 자세한 내용을 알아보려면 Alertmanager 설명서의 구성을 참조하십시오.

> **Note**
>
> 위의 Prometheus 설명서에 나와 있는 `repeat_interval` 구성에는 Amazon Managed Service for Prometheus의 추가 제한 사항이 있습니다. 허용되는 최댓값은 5일입니다. 5일보다 높게 설정하면 5일로 처리되며 5일이 경과한 후 알림이 다시 전송됩니다.

Amazon Managed Service for Prometheus에서 알림 관리자 구성 파일은 YAML 파일의 루트에 있는 `alertmanager_config` 키 내에 모든 알림 관리자 구성 콘텐츠를 포함해야 합니다.

다음은 기본 예제 알림 관리자 구성 파일입니다.

```
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
```

현재 지원되는 유일한 수신기는 Amazon Simple Notification Service(Amazon SNS)입니다. 구성에 다른 유형의 수신기가 나열되어 있는 경우 거부됩니다.

다음은 `template_files` 블록과 `alertmanager_config` 블록을 모두 사용하는 또 다른 샘플 알림 관리자 구성 파일입니다.

```
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
```

기본 Amazon SNS 템플릿 블록

명시적으로 재정의하지 않는 한, 기본 Amazon SNS 구성은 다음 템플릿을 사용합니다.

```
{{ 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 }}
```

알림 수신기 설정

Amazon Managed Service for Prometheus에서 현재 지원되는 유일한 알림 수신기는 Amazon Simple Notification Service(Amazon SNS)입니다. 자세한 내용은 Amazon SNS란 무엇입니까?를 참조하십시오.

주제

- (선택 사항) 새 Amazon SNS 주제 생성
- Amazon Managed Service for Prometheus에 Amazon SNS 주제로 메시지를 전송할 수 있는 권한 부여
- 알림 관리자 구성 파일에 Amazon SNS 주제 지정
- (선택 사항) Amazon SNS에 JSON을 출력하도록 알림 관리자 구성
- (선택 사항) Amazon SNS에서 다른 대상으로 전송
- SNS 수신기 메시지 검증 및 잘림 규칙

(선택 사항) 새 Amazon SNS 주제 생성

기존 Amazon SNS 주제를 사용하거나 새로운 주제를 생성할 수 있습니다. 주제의 알림을 이메일, SMS 또는 HTTP로 전달할 수 있도록 표준 유형의 주제를 사용하는 것이 좋습니다.

알림 관리자 수신기로 사용할 새 Amazon SNS 주제를 생성하려면 1단계: 주제 생성의 단계를 따르십시오. 주제 유형으로는 표준을 선택해야 합니다.

해당 Amazon SNS 주제로 메시지가 전송될 때마다 이메일을 수신하려면 2단계: 주제 구독 생성의 단계를 따르십시오.

Amazon Managed Service for Prometheus에 Amazon SNS 주제로 메시지를 전송할 수 있는 권한 부여

Amazon Managed Service for Prometheus에 Amazon SNS 주제로 메시지를 전송할 수 있는 권한을 부여해야 합니다. 다음 정책 문에는 혼동된 대리자 보안 문제를 방지하는 데 도움이 되는 `Condition` 문이 포함되어 있습니다. 이 `Condition` 문은 Amazon SNS 주제에 대한 액세스를 제한하여 이 특정 계정 및 Amazon Managed Service for Prometheus WorkSpace에서 발생하는 작업만 허용하도록 합니다. 혼동된 대리자 문제에 대한 자세한 내용은 교차 서비스 혼동된 대리자 예방를 참조하십시오.

Amazon Managed Service for Prometheus에 Amazon SNS 주제에 메시지를 전송할 수 있는 권한을 부여하려면

1. https://console.aws.amazon.com/sns/v3/home에서 Amazon SNS 콘솔을 엽니다.
2. 탐색 창에서 주제를 선택합니다.
3. Amazon Managed Service for Prometheus에서 사용하는 주제의 이름을 선택합니다.
4. 편집을 선택합니다.
5. 액세스 정책을 선택하고 기존 정책에 다음 정책 문을 추가합니다.

```
{
    "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"
}
```

[선택 사항] SNS 주제에 서비스 측 암호화(SSE)가 활성화된 경우 "Action" 블록의 KMS 키 정책에 다음 권한을 추가해야 합니다. 자세한 내용을 알아보려면 SNS 주제에 대한 AWS KMS 권한을 참조하십시오.

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

6. 변경 사항 저장을 선택합니다.

> **Note**
>
> 기본적으로 Amazon SNS는 `AWS:SourceOwner`에 대한 조건을 적용해서 액세스 정책을 생성합니다. 자세한 내용은 SNS 액세스 정책을 참조하십시오.

> **Note**
>
> IAM은 가장 제한적인 정책 우선 규칙을 따릅니다. SNS 주제에서 문서화된 Amazon SNS 정책 블록보다 더 제한적인 정책 블록이 있는 경우 주제 정책에 대한 권한은 부여되지 않습니다. 정책을 평가하고 어떤 권한이 부여되었는지 알아보려면 정책 평가 로직을 참조하십시오.

교차 서비스 혼동된 대리자 예방

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. AWS에서는 교차 서비스 가장으로 인해 혼동된 대리자 문제가 발생할 수 있습니다. 교차 서비스 가장은 한 서비스(호출하는 서비스)가 다른 서비스(호출되는 서비스)를 직접적으로 호출할 때 발생할 수 있습니다. 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다.

Amazon Managed Service for Prometheus가 리소스에 대해 Amazon SNS에 부여하는 권한을 제한하려면 리소스 정책에서 `aws:SourceArn` 및 `aws:SourceAccount` 글로벌 조건 컨텍스트 키를 사용하는 것이 좋습니다. 두 전역 조건 컨텍스트 키를 모두 사용하는 경우 `aws:SourceAccount` 값과 `aws:SourceArn` 값의 계정은 동일한 정책 문에서 사용할 경우 동일한 계정 ID를 사용해야 합니다.

`aws:SourceArn` 값은 Amazon Managed Service for Prometheus WorkSpace의 ARN이어야 합니다.

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 리소스의 전체 ARN이 포함된 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용하는 것입니다. 리소스의 전체 ARN을 모를 경우 또는 여러 리소스를 지정하는 경우, ARN의 알 수 없는 부분에 대해 와일드카드(*)를 포함한 `aws:SourceArn` 전역 조건 컨텍스트 키를 사용합니다. 예: `arn:aws:servicename::123456789012:*`.

Amazon Managed Service for Prometheus에 Amazon SNS 주제로 메시지를 전송할 수 있는 권한 부여에 표시되는 정책은 Amazon Managed Service for Prometheus에서 `aws:SourceArn` 및

`aws:SourceAccount` 전역 조건 컨텍스트 키를 사용하여 혼동된 대리자 문제를 방지하는 방법을 보여 줍니다.

알림 관리자 구성 파일에 Amazon SNS 주제 지정

이제 Amazon SNS 수신기를 알림 관리자 구성에 추가할 수 있습니다. 이를 위해서는 Amazon SNS 주제의 Amazon 리소스 이름(ARN)을 알아야 합니다.

Amazon SNS 수신기 구성에 대한 자세한 내용은 Prometheus 구성 설명서에서 <sns_configs>를 참조하십시오.

지원되지 않는 속성

Amazon Managed Service for Prometheus는 Amazon SNS를 알림 수신기로 지원합니다. 하지만 서비스 제약으로 인해 Amazon SNS 수신기의 모든 속성이 지원되는 것은 아닙니다. 다음 속성은 Amazon Managed Service for Prometheus 알림 관리자 구성 파일에서 허용되지 않습니다.

- `api_url:` - Amazon Managed Service for Prometheus가 `api_url`을 설정하므로 이 속성은 허용되지 않습니다.
- `Http_config` - 이 속성을 사용하면 외부 프록시를 설정할 수 있습니다. Amazon Managed Service for Prometheus는 현재 이 기능을 지원하지 않습니다.

또한 리전 속성이 있으려면 SigV4 설정이 필요합니다. 리전 속성이 없으면 Amazon Managed Service for Prometheus에는 권한 부여를 요청하는 데 필요한 정보가 충분하지 않습니다.

Amazon SNS 주제를 수신기로 사용하여 알림 관리자를 구성하려면

1. 기존 알림 관리자 구성 파일을 사용하는 경우 텍스트 편집기에서 엽니다.
2. `receivers` 블록에 Amazon SNS 이외의 현재 수신기가 있는 경우 해당 수신기를 제거하십시오. 여러 Amazon SNS 주제를 `receivers` 블록 내 개별 `sns_config` 블록에 배치하여 수신기가 되도록 구성할 수 있습니다.
3. `receivers` 섹션 내에 다음 YAML 블록을 추가합니다.

```
- name: name_of_receiver
  sns_configs:
    - sigv4:
        region: region
      topic_arn: ARN_of_SNS_topic
      subject: somesubject
```

```
      attributes:
        key: somekey
        value: somevalue
```

subject를 지정하지 않으면 기본적으로 레이블 이름과 값이 포함된 기본 템플릿으로 제목이 생성되므로 SNS에 맞지 않게 값이 너무 길어질 수 있습니다. 제목에 적용되는 템플릿을 변경하려면 이 설명서의 (선택 사항) Amazon SNS에 JSON을 출력하도록 알림 관리자 구성 섹션을 참조하십시오.

이제 Amazon Managed Service for Prometheus에 알림 관리자 구성 파일을 업로드해야 합니다. 자세한 내용은 Amazon Managed Service for Prometheus에 알림 관리자 구성 파일 업로드 섹션을 참조하십시오.

(선택 사항) Amazon SNS에 JSON을 출력하도록 알림 관리자 구성

알림을 JSON 형식으로 전송하도록 알림 관리자를 구성하면 AWS Lambda 또는 웹후크 수신 엔드포인트에서 Amazon SNS의 다운스트림으로 알림을 처리할 수 있습니다. Amazon Managed Service for Prometheus 알림 관리자에 제공되는 기본 템플릿은 메시지 페이로드를 텍스트 목록 형식으로 출력하므로 구문 분석이 쉽지 않을 수 있습니다. 기본 템플릿을 사용하는 대신 메시지 내용을 JSON으로 출력하는 사용자 지정 템플릿을 정의하여 다운스트림 함수에서 더 쉽게 구문 분석하도록 할 수 있습니다.

알림 관리자의 메시지를 JSON 형식으로 Amazon SNS로 출력하려면 template_files 루트 섹션 내에 다음 코드를 포함하도록 알림 관리자 구성을 업데이트하십시오.

```
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**
>
> 이 템플릿은 영숫자 데이터로 JSON을 생성합니다. 데이터에 특수 문자가 있는 경우 이 템플릿을 사용하기 전에 특수 문자를 인코딩하십시오.

이 템플릿이 발신 알림에 사용되도록 하려면 다음과 같이 `alertmanager_config` 블록에서 템플릿을 참조하십시오.

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

> **Note**
>
> 이 템플릿은 전체 메시지 본문을 JSON으로 작성하기 위한 것입니다. 이 템플릿은 전체 메시지 본문을 덮어씁니다. 이 특정 템플릿을 사용하려는 경우 메시지 본문을 재정의할 수 없습니다. 수동으로 수행한 모든 재정의는 템플릿보다 우선합니다.

해당 내용은 다음을 참조하십시오.

- 알림 관리자 구성 파일: 알림 관리자 구성 파일 생성을 참조하십시오.
- 구성 파일 업로드: Amazon Managed Service for Prometheus에 알림 관리자 구성 파일 업로드을 참조하십시오.

(선택 사항) Amazon SNS에서 다른 대상으로 전송

현재 Amazon Managed Service for Prometheus는 알림 메시지를 Amazon SNS로만 직접 보낼 수 있습니다. Amazon SNS에서 이러한 메시지를 이메일, webhook, Slack, OpsGenie 등의 다른 대상으로 전송하도록 구성할 수 있습니다.

이메일

메시지를 이메일로 출력하도록 Amazon SNS 주제를 구성하려면 구독을 생성하십시오. Amazon SNS 콘솔에서 구독 탭을 선택하여 구독 목록 페이지를 엽니다. 구독 생성을 선택하고 이메일을 선택합니다. Amazon SNS는 나열된 이메일 주소로 확인 이메일을 보냅니다. 확인을 수락하면 구독한 주제의 Amazon SNS 알림을 이메일로 받을 수 있습니다. 자세한 내용은 Amazon SNS 주제에 구독 설정을 참조하십시오.

Webhook

메시지를 Webhook 엔드포인트로 출력하도록 Amazon SNS 주제를 구성하려면 구독을 생성하십시오. Amazon SNS 콘솔에서 구독 탭을 선택하여 구독 목록 페이지를 엽니다. 구독 생성을 선택하고 HTTP/HTTPS를 선택합니다. 구독을 생성한 후에는 확인 단계에 따라 구독을 활성화해야 합니다. 활성화되면 HTTP 엔드포인트는 Amazon SNS 알림을 수신합니다. 자세한 내용은 Amazon SNS 주제에 구독 설정을 참조하십시오. Slack webhook를 사용하여 다양한 대상으로 메시지를 게시하는 방법에 대한 자세한 내용은 Webhook를 사용하여 Amazon Chime, Slack 또는 Microsoft Teams에 Amazon SNS 메시지를 게시하려면 어떻게 해야 합니까?를 참조하십시오.

Slack

메시지를 Slack에 출력하도록 Amazon SNS 주제를 구성하는 방법에는 두 가지가 있습니다. Slack의 이메일-채널 통합 기능으로 통합하여 Slack에서 이메일 메시지를 수락하고 Slack 채널로 전달하도록 하거나 Lambda 함수를 사용하여 Amazon SNS 알림을 Slack에 다시 작성할 수 있습니다. Slack 채널로 이메일을 전달하는 방법에 대한 자세한 내용은 Slack Webhook의 AWS SNS 주제 구독 확인을 참조하십시오. Amazon SNS 메시지를 Slack으로 변환하는 Lambda 함수를 구성하는 방법에 대한 자세한 내용은 Amazon Managed Service for Prometheus를 Slack과 통합하는 방법을 참조하십시오.

OpsGenie

메시지를 OpsGenie로 출력하도록 Amazon SNS 주제를 구성하는 방법에 대한 자세한 내용은 수신 Amazon SNS와 Opsgenie 통합을 참조하십시오.

SNS 수신기 메시지 검증 및 잘림 규칙

필요한 경우 SNS 수신기는 다음 규칙에 따라 SNS 메시지를 검증하거나 자르거나 수정합니다.

- 메시지에 utf가 아닌 문자가 포함되어 있습니다.
  - 메시지가 “오류 - 유효한 UTF-8 인코딩 문자열이 아닙니다.”로 바뀝니다.
  - 키가 “잘림”이고 값이 “true”인 메시지 속성 하나가 추가됩니다.
  - “수정됨” 키와 “메시지: 오류 - 유효한 UTF-8 인코딩 문자열이 아님”이라는 값과 함께 메시지 속성 하나가 추가됩니다.
- 메시지가 비어 있습니다.
  - 메시지가 “오류 - 메시지를 비워둘 수 없음”으로 바뀝니다.
  - “수정됨” 키와 “메시지: 오류 - 메시지는 비어 있으면 안 됩니다.” 값과 함께 메시지 속성 하나가 추가됩니다.
- 메시지가 잘렸습니다.
  - 메시지에는 잘린 내용이 포함됩니다.
  - 키가 “잘림”이고 값이 “true”인 메시지 속성 하나가 추가됩니다.
  - 키가 “수정됨”이고 값이 “메시지: 오류 - 메시지가 256KB 크기 제한을 초과하여 *X*KB에서 잘렸습니다.” 값과 함께 메시지 속성 하나가 추가됩니다.
- 제목이 ASCII가 아닙니다.
  - 제목은 “오류 - 인쇄할 수 없는 ASCII 문자 포함”으로 바뀝니다.
  - “수정됨” 키와 “제목: 오류 - 인쇄할 수 없는 ASCII 문자 포함” 값과 함께 메시지 속성 하나가 추가됩니다.
- 제목이 잘렸습니다.
  - 제목에는 잘린 내용이 표시됩니다.
  - “수정됨”이라는 키와 “제목: 오류 - 제목이 100자 크기 제한을 초과하여 *X*자에서 잘렸습니다.”라는 값과 함께 메시지 속성 하나가 추가됩니다.
- 메시지 속성에 잘못된 키/값이 있습니다.
  - 잘못된 메시지 속성은 제거됩니다.
  - 키가 “수정됨”이고 값이 “MessageAttribute: 오류 - 잘못된 MessageAttributeKey 또는 MessageAttributeValue로 인해 메시지 속성 중 *X*가 제거되었습니다.”인 메시지 속성 하나가 추가됩니다.
- 메시지 속성이 잘렸습니다.
  - 추가 메시지 속성은 제거됩니다.

- 키가 “수정됨”이고 값이 “MessageAttribute: 오류 - 256KB 크기 제한을 초과하여 메시지 속성 중 *X*가 제거되었습니다.”인 메시지 속성 하나가 추가됩니다.

Amazon Managed Service for Prometheus에 알림 관리자 구성 파일 업로드

이제 Amazon Managed Service for Prometheus에 알림 관리자 구성 파일을 업로드해야 합니다. 콘솔 또는 AWS CLI를 사용하여 로드할 수 있습니다.

Amazon Managed Service for Prometheus 콘솔을 사용하여 알림 관리자 구성을 업로드하려면

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 페이지 왼쪽 상단에서 메뉴 아이콘을 선택한 다음, 모든 WorkSpace를 선택합니다.
3. WorkSpace의 WorkSpace ID를 선택한 다음, 알림 관리자 탭을 선택합니다.
4. WorkSpace에 아직 알림 관리자 정의가 없는 경우 정의 추가를 선택합니다. WorkSpace에 바꾸려는 알림 관리자 정의가 있는 경우 정의 바꾸기를 선택합니다.
5. 파일 선택을 선택하고 알림 관리자 정의 파일을 선택한 다음, 계속을 선택합니다.

AWS CLI를 사용하여 알림 관리자 구성을 WorkSpace에 처음으로 업로드하려면

1. Base64는 알림 관리자 파일의 내용을 인코딩합니다. Linux에서 다음 명령을 사용할 수 있습니다.

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

   macOS에서 다음 명령을 사용할 수 있습니다.

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

2. 파일을 업로드하려면 다음 명령 중 하나를 입력합니다.

   AWS CLI 버전 2에서는 다음을 입력합니다.

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

AWS CLI 버전 1에서는 다음을 입력합니다.

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

3. 알림 관리자 구성이 활성화되는 데 몇 초 정도 걸립니다. 상태를 확인하려면 다음 명령을 입력합니다.

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

`status`가 `ACTIVE`이면 새 알림 관리자 정의가 적용된 것입니다.

AWS CLI를 사용하여 WorkSpace의 알림 관리자 구성을 새 구성으로 바꾸려면

1. Base64는 알림 관리자 파일의 내용을 인코딩합니다. Linux에서 다음 명령을 사용할 수 있습니다.

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

macOS에서 다음 명령을 사용할 수 있습니다.

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

2. 파일을 업로드하려면 다음 명령 중 하나를 입력합니다.

AWS CLI 버전 2에서는 다음을 입력합니다.

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

AWS CLI 버전 1에서는 다음을 입력합니다.

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

3. 새 알림 관리자 구성이 활성화되는 데 몇 초 정도 걸립니다. 상태를 확인하려면 다음 명령을 입력합니다.

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

`status`가 `ACTIVE`이면 새 알림 관리자 정의가 적용된 것입니다. 그때까지는 이전 알림 관리자 구성이 계속 활성화된 상태를 유지합니다.

Amazon Managed Grafana 또는 오픈 소스 Grafana와 알림 통합

Amazon Managed Service for Prometheus 내 Alertmanager에서 생성한 알림 규칙은 Amazon Managed Grafana 및 Grafana에서 전달되고 확인되므로 단일 환경에서 알림 규칙과 알림을 통합할 수 있습니다. Amazon Managed Grafana 내에서 알림 규칙 및 생성된 알림을 볼 수 있습니다.

사전 조건

Amazon Managed Service for Prometheus를 Amazon Managed Grafana에 통합하려면 먼저 다음 사전 조건을 충족해야 합니다.

- Amazon Managed Service for Prometheus 및 IAM 역할을 프로그래밍 방식으로 생성하려면 기존 AWS 계정 및 IAM 자격 증명이 있어야 합니다.

  AWS 계정 및 IAM 자격 증명 생성에 대한 자세한 내용은 설정 섹션을 참조하세요.
- Amazon Managed Service for Prometheus 워크스페이스가 있어야 하며 여기에 데이터를 수집하고 있어야 합니다. 새 워크스페이스을 설정하려면 WorkSpace 생성 섹션을 참조하세요. Alertmanager 및 Ruler 등의 Prometheus 개념에도 익숙해야 합니다. 이러한 항목에 대한 자세한 내용은 Prometheus 설명서를 참조하세요.
- Amazon Managed Service for Prometheus에 Alertmanager 구성과 규칙 파일이 이미 구성되어 있어야 합니다. Amazon Managed Service for Prometheus의 Alertmanager에 대한 자세한 내용은 알림 관리자 섹션을 참조하세요. 규칙에 대한 자세한 설명은 기록 규칙 및 알림 규칙 섹션을 참조하십시오.
- Amazon Managed Grafana를 설정했거나 Grafana의 오픈 소스 버전을 실행 중이어야 합니다.
  - Amazon Managed Grafana를 사용하는 경우 Grafana 알림을 사용하고 있어야 합니다. 자세한 내용은 레거시 대시보드 알림을 Grafana 알림으로 마이그레이션을 참조하세요.
  - Grafana의 오픈 소스 버전을 사용하는 경우 버전 9.1 이상을 실행해야 합니다.

> **Note**
>
> 이전 버전의 Grafana를 사용할 수 있지만 통합 알림(Grafana 알림) 기능을 활성화해야 하며 Grafana에서 Amazon Managed Service for Prometheus로 호출하도록 sigv4 프록시를 설정해야 할 수도 있습니다. 자세한 내용은 Amazon Managed Service for Prometheus와 함께 사용할 Grafana 오픈 소스 또는 Grafana Enterprise를 설정하십시오. 섹션을 참조하세요.

- Amazon Managed Grafana에는 Prometheus 리소스에 대한 다음과 같은 권한이 있어야 합니다. https://docs.aws.amazon.com/grafana/latest/userguide/AMG-manage-permissions.html에 설명된 서비스 관리형 또는 고객 관리형 정책에 추가해야 합니다.
  - `aps:ListRules`
  - `aps:ListAlertManagerSilences`
  - `aps:ListAlertManagerAlerts`
  - `aps:GetAlertManagerStatus`
  - `aps:ListAlertManagerAlertGroups`
  - `aps:PutAlertManagerSilences`
  - `aps:DeleteAlertManagerSilence`

Amazon Managed Grafana 설정

Amazon Managed Service for Prometheus 인스턴스에 이미 규칙 및 알림을 설정한 경우 Amazon Managed Grafana를 해당 알림에 대한 대시보드로 사용하도록 구성하는 작업은 전적으로 Amazon Managed Grafana 내에서 수행됩니다.

Amazon Managed Grafana를 알림 대시보드로 구성하려면

1. 워크스페이스의 Grafana 콘솔을 엽니다.
2. 구성에서 데이터 소스를 선택합니다.
3. Prometheus 데이터 소스를 생성하거나 엽니다. 이전에 Prometheus 데이터 소스를 설정하지 않은 경우 Grafana에 Prometheus 데이터 소스 추가에서 자세한 내용을 참조하세요.
4. Prometheus 데이터 소스에서 Alertmanager UI를 통한 알림 관리를 선택합니다.
5. 데이터 소스 인터페이스로 돌아갑니다.
6. 새 Alertmanager 데이터 소스를 생성합니다.

7. Alertmanager 데이터 소스 구성 페이지에서 다음 설정을 추가합니다.
   - 구현을 `Prometheus`로 설정합니다.
   - URL 설정의 경우 Prometheus 워크스페이스의 URL을 사용하고 워크스페이스 ID 다음에 나오는 모든 항목을 제거한 다음, 끝에 `/alertmanager`를 추가합니다. 예: *`https://aps-workspaces.us-east1.amazonaws.com/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager`*.
   - 인증에서 SigV4Auth를 켭니다. 이렇게 하면 요청에 AWS 인증을 사용하도록 Grafana에 지시하게 됩니다.
   - SigV4Auth 세부 정보에서 기본 리전으로 Prometheus 인스턴스의 리전(예: `us-east-1`)을 입력합니다.
   - 기본 옵션을 `true`로 설정합니다.
8. 저장 및 테스트를 선택합니다.
9. 이제 Amazon Managed Service for Prometheus 알림이 Grafana 인스턴스에서 작동하도록 구성해야 합니다. Grafana 알림 페이지에서 Amazon Managed Service for Prometheus 인스턴스의 모든 알림 규칙, 알림 그룹(활성 알림 포함) 및 무음이 표시되는지 확인합니다.

알림 관리자 문제 해결

CloudWatch 로그을 사용하여 알림 관리자 및 규칙 관리자 관련 문제를 해결할 수 있습니다. 이 섹션에는 알림 관리자 관련 문제 해결 항목이 포함되어 있습니다.

주제

빈 콘텐츠 경고

로그에 다음 경고가 포함된 경우

```
{
   "workspaceId": "ws-efdc5b42-b051-11ec-b123-4567ac120002",
```

```
   "message": {
       "log": "Message has been modified because the content was empty."
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

알림 관리자 템플릿이 아웃바운드 알림을 빈 메시지로 해결했음을 의미합니다.

취할 조치

알림 관리자 템플릿을 검증하고 모든 수신기 경로에 유효한 템플릿이 있는지 확인하십시오.

비 ASCII 경고

로그에 다음 경고가 포함된 경우

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

제목에 ASCII가 아닌 문자가 포함되어 있음을 의미합니다.

취할 조치

템플릿의 제목 필드에서 ASCII가 아닌 문자를 포함할 수 있는 레이블에 대한 참조를 제거합니다.

잘못된 **key/value** 경고

로그에 다음 경고가 포함된 경우

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

```
   },
   "component": "alertmanager"
}
```

키/값이 유효하지 않아 일부 메시지 속성이 제거되었음을 의미합니다.

취할 조치

메시지 속성을 채우는 데 사용 중인 템플릿을 다시 평가하여 유효한 SNS 메시지 속성으로 확인되는지 알아봅니다. 메시지를 Amazon SNS 주제로 검증하는 방법에 대한 자세한 내용은 SNS 주제 검증을 참조하십시오.

메시지 제한 경고

로그에 다음 경고가 포함된 경우

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

일부 메시지 크기가 너무 큰 것을 의미합니다.

취할 조치

알림 수신기 메시지 템플릿을 살펴보고 크기 제한에 맞도록 재작업하십시오.

리소스 기반 정책 오류 없음

로그에 다음 오류가 포함된 경우

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

```
        "level": "ERROR"
    },
    "component": "alertmanager"
}
```

즉, Amazon Managed Service for Prometheus에 지정된 SNS 주제로 알림을 제출할 권한이 없음을 의미합니다.

취할 조치

Amazon SNS 주제의 액세스 정책이 SNS 메시지를 주제에 전송할 수 있는 권한을 Amazon Managed Service for Prometheus에 부여하는지 검증합니다. IAM 정책 시뮬레이터로 IAM 정책 시뮬레이터에 대해 주제 정책을 검증할 수 있습니다. IAM 역할에 필요한 권한과 정책이 있는지 확인해야 합니다. IAM 권한 및 정책에 대한 자세한 내용은 IAM 권한 및 정책을 참조하십시오.

로깅 및 모니터링

Amazon 로깅 및 모니터링 기능을 사용하여 Prometheus용 Amazon 관리 서비스 리소스 사용을 관리할 수 있습니다. CloudWatch

- CloudWatch 메트릭를 사용하여 Amazon Managed Service for Prometheus를 모니터링합니다.
- CloudWatch 로그를 사용하여 Amazon Managed Service for Prometheus 알림 관리자 및 규칙 관리자 이벤트를 쿼리하고 확인합니다.

CloudWatch 메트릭

Prometheus용 Amazon 매니지드 서비스는 사용량 지표를 에 판매합니다. CloudWatch 이러한 지표는 워크스페이스 사용률에 대한 가시성을 제공합니다. 벤드 메트릭은 및 네임스페이스에서 찾을 수 있습니다. `AWS/Usage` `AWS/Prometheus` CloudWatch 이러한 지표는 무료로 사용할 수 있습니다 CloudWatch. 사용량 지표에 대한 자세한 내용은 CloudWatch 사용량 지표를 참조하십시오.

CloudWatch 지표 이름	리소스 이름	CloudWatch 네임스페이스	설명
ResourceCount	IngestionRate	`AWS/Usage`	샘플 수집 속도 단위: 초당 개수 유효한 통계: Average, Minimum, Maximum, Sum
ResourceCount	ActiveSeries	`AWS/Usage`	워크스페이스당 활성 시리즈 수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
ResourceCount	ActiveAlerts	`AWS/Usage`	워크스페이스당 활성 알림 수 단위: 개수

CloudWatch 지표 이름	리소스 이름	CloudWatch 네임스페이스	설명
			유효한 통계: Average, Minimum, Maximum, Sum
ResourceCount	SizeOfAlerts	AWS/Usage	작업 공간에 있는 모든 경고의 총 크기 (바이트) 단위: 바이트 유효한 통계: Average, Minimum, Maximum, Sum
ResourceCount	SuppressedAlerts	AWS/Usage	WorkSpace당 숨김 상태 알림 수 알림은 무음 또는 금지로 억제할 수 있습니다. 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
ResourceCount	UnprocessedAlerts	AWS/Usage	WorkSpace당 처리되지 않은 상태인 알림의 수 에서 알림을 받은 후에는 처리되지 않은 상태가 AlertManager 되지만 다음 집계 그룹 평가를 기다리고 있습니다. 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum

CloudWatch 지표 이름	리소스 이름	CloudWatch 네임스페이스	설명
ResourceCount	AllAlerts	AWS/Usage	WorkSpace별 모든 상태의 경고 수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
AlertManagerAlertsReceived	-	AWS/Prometheus	알림 관리자가 수신한 총 성공 알림 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
AlertManagerNotificationsFailed	-	AWS/Prometheus	실패한 알림 전송 수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
AlertManagerNotificationsThrottled	-	AWS/Prometheus	병목 현상이 발생한 알림 수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
Discarded Samples*	-	AWS/Prometheus	이유별 폐기된 샘플 수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum

CloudWatch 지표 이름	리소스 이름	CloudWatch 네임스페이스	설명
RuleEvaluations	-	AWS/Prometheus	총 규칙 평가 수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
RuleEvaluationFailures	-	AWS/Prometheus	해당 간격 내의 규칙 평가 실패 횟수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum
RuleGroupIterationsMissed	-	AWS/Prometheus	해당 간격 동안 누락된 규칙 그룹 반복 횟수 단위: 개수 유효한 통계: Average, Minimum, Maximum, Sum

* 샘플이 폐기되는 몇 가지 이유는 다음과 같습니다.

이유	의미
greater_than_max_sample_age	현재 시간보다 오래된 로그 줄 삭제
new-value-for-timestamp	중복 샘플은 이전에 기록된 것과 다른 타임스탬프와 함께 전송됩니다.
per_metric_series_limit	지표별 활성 시리즈 제한에 도달했습니다.
per_user_series_limit	총 활성 시리즈 수 제한에 도달했습니다.

이유	의미
rate_limited	수집 속도가 제한되었습니다.
sample-out-of-order	샘플이 잘못된 순서로 전송되어 처리할 수 없습니다.
label_value_too_long	레이블 값이 허용된 문자 제한보다 깁니다.
max_label_names_per_series	지표별 레이블 이름에 도달했습니다.
missing_metric_name	지표 이름은 제공되지 않습니다.
metric_name_invalid	잘못된 지표 이름이 제공되었습니다.
label_invalid	잘못된 레이블이 제공되었습니다.
duplicate_label_names	중복된 레이블 이름이 제공되었습니다.

Note

존재하지 않거나 누락된 지표는 해당 지표의 값이 0인 것과 같습니다.

Note

`RuleGroupIterationsMissed`, `RuleEvaluations` 및 `RuleEvaluationFailures`에는 다음과 같은 구조의 `RuleGroup` 차원이 있습니다.
`RuleGroupNamespace`;*`RuleGroup`*

Prometheus 벤더 메트릭에 CloudWatch 알람 설정

경보를 사용하여 Prometheus 리소스 사용을 모니터링할 수 있습니다. CloudWatch

ActiveSeriesPrometheus의 개수에 대해 알람을 설정하려면

1. 그래프로 표시된 지표 탭을 선택하고 레이블까지 아래로 스크롤합니다. ActiveSeries

   그래프로 표시된 지표 보기에서는 현재 수집 중인 지표만 표시됩니다.

2. 작업 열에서 알림 아이콘을 선택합니다.
3. 지표 및 조건 지정에서 조건 값 필드에 임곗값 조건을 입력하고 다음을 선택합니다.
4. 작업 구성에서 기존 SNS 주제를 선택하거나 알림을 보낼 새 SNS 주제를 생성합니다.
5. 이름 및 설명 추가에서 경보 이름과 설명(선택 사항)을 추가합니다.
6. 경보 생성을 선택하세요.

CloudWatch 로그

Prometheus용 Amazon 관리형 서비스는 Amazon Logs의 로그 그룹에 알림 관리자 및 눈금자 오류 및 경고 이벤트를 기록합니다. CloudWatch 알림 관리자 및 규칙 관리자에 대한 자세한 내용은 이 안내서의 알림 관리자 주제를 참조하세요. 작업 공간 로그 데이터를 Logs의 로그 스트림에 게시할 수 있습니다. CloudWatch Amazon Managed Service for Prometheus 콘솔에서 또는 AWS CLI를 사용하여 모니터링하려는 로그를 구성할 수 있습니다. CloudWatch콘솔에서 이러한 로그를 보거나 쿼리할 수 있습니다. 콘솔에서 로그 CloudWatch 로그 스트림을 보는 방법에 대한 자세한 내용은 CloudWatch 사용 설명서의 로그 그룹 및 로그 스트림 작업을 참조하십시오. CloudWatch

CloudWatch 프리 티어를 사용하면 최대 5Gb의 로그를 Logs에 CloudWatch 게시할 수 있습니다. 프리 티어 허용량을 초과하는 로그는 CloudWatch 요금제에 따라 요금이 부과됩니다.

주제

- 로그 구성 CloudWatch

로그 구성 CloudWatch

Prometheus용 Amazon 관리형 서비스는 Amazon Logs의 로그 그룹에 알림 관리자 및 눈금자 오류 및 경고 이벤트를 기록합니다. CloudWatch

Prometheus용 Amazon 관리 서비스 콘솔에서 AWS CLI 또는 API 요청을 호출하여 CloudWatch 로그 로깅 구성을 설정할 수 있습니다. `create-logging-configuration`

사전 조건

`create-logging-configuration`을 호출하기 전에 다음 정책 또는 이에 상응하는 권한을 ID 또는 역할에 추가합니다.

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

로그를 구성하려면 CloudWatch

콘솔 또는 를 사용하여 Prometheus용 Amazon 관리 서비스에서 로깅을 구성할 수 있습니다. AWS AWS CLI

Console

Amazon Managed Service for Prometheus 콘솔에서 로깅을 구성하려면

1. 워크스페이스 세부 정보 패널의 로그 탭으로 이동합니다.
2. 로그 패널의 오른쪽 상단에서 로그 관리를 선택합니다.
3. 로그 수준 드롭다운 목록에서 모두 선택합니다.
4. 로그 그룹 드롭다운 목록에서 로그를 게시할 로그 그룹을 선택합니다.

   콘솔에서 새 로그 그룹을 생성할 수도 있습니다. CloudWatch
5. 변경 사항 저장을 선택합니다.

AWS CLI

를 사용하여 로깅 구성을 설정할 수 AWS CLI있습니다.

를 사용하여 로깅을 구성하려면 AWS CLI

- AWS CLI를 사용하여 다음 명령을 실행합니다.

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

제한 사항

- 모든 이벤트가 기록되지는 않습니다.

  Amazon Managed Service for Prometheus는 `warning` 또는 `error` 수준의 이벤트만 로깅합니다.
- 정책 크기 제한

  CloudWatch 로그 리소스 정책은 5120자로 제한됩니다. 정책이 이 크기 제한에 근접하는 것을 CloudWatch 로그에서 감지하면 로 `/aws/vendedlogs/` 시작하는 로그 그룹이 자동으로 활성화됩니다.

  로깅이 활성화된 알림 규칙을 생성하는 경우 Prometheus용 Amazon Managed Service는 지정된 로그 그룹으로 로그 리소스 정책을 CloudWatch 업데이트해야 합니다. CloudWatch 로그 리소스 정책 크기 한도에 도달하지 않으려면 로그 로그 그룹 이름에 접두사를 CloudWatch 붙이십시오. `/aws/vendedlogs/` Amazon Managed Service for Prometheus 콘솔에서 로그 그룹을 생성하면 로그 그룹 이름에 접두사 `/aws/vendedlogs/`가 붙습니다. 자세한 내용은 CloudWatch Logs User Guide의 특정 AWS 서비스에서 로깅 활성화를 참조하십시오.

비용 이해 및 최적화

다음과 같은 자주 묻는 질문과 그에 대한 답변은 Amazon Managed Service for Prometheus와 관련된 비용을 이해하고 최적화하는 데 도움이 될 수 있습니다.

비용에 영향을 미치는 요인은 무엇인가요?

대부분의 고객에서는 지표 수집이 비용 대부분을 차지합니다. 쿼리 사용량이 많은 고객에게는 처리된 쿼리 샘플에 따라 약간의 비용이 발생하며, 지표 스토리지가 전체 비용에서 차지하는 비중은 적습니다. 각 요금에 대한 자세한 내용은 Amazon Managed Service for Prometheus 제품 페이지의 요금을 참조하세요.

비용을 낮추는 가장 좋은 방법은 무엇인가요? 수집 비용을 낮추려면 어떻게 해야 하나요?

대부분의 고객에게는 지표 저장 비용이 아닌 수집 요금이 비용의 대부분을 차지합니다. 수집 빈도를 줄이거나(수집 간격을 늘림) 수집되는 활성 시리즈 수를 줄이면 수집 요금을 줄일 수 있습니다.

컬렉션 에이전트에서 컬렉션 (스크래핑) 간격을 늘릴 수 있습니다. Prometheus 서버 (에이전트 모드에서 실행) 와 AWS Distro OpenTelemetry for (ADOT) 컬렉터 모두 구성을 지원합니다. `scrape_interval` 예를 들어 수집 간격을 30초에서 60초로 늘리면 수집 사용량이 절반으로 줄어듭니다.

`<relabel_config>`를 사용하여 Amazon Managed Service for Prometheus로 전송되는 지표를 필터링할 수도 있습니다. Prometheus 에이전트 구성에서 레이블을 다시 지정하는 방법에 대한 자세한 내용은 Prometheus 설명서의 https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config를 참조하세요.

쿼리 비용을 낮추는 가장 좋은 방법은 무엇인가요?

쿼리 비용은 처리된 샘플 수를 기준으로 합니다. 쿼리 빈도를 줄여 쿼리 비용을 줄일 수 있습니다.

쿼리 비용에 가장 큰 영향을 미치는 쿼리를 더 잘 파악하려는 경우 지원 담당자에게 문의하여 티켓을 제출할 수 있습니다. Amazon Managed Service for Prometheus 팀이 비용에 가장 큰 영향을 미치는 쿼리를 이해하도록 도와드릴 수 있습니다.

지표의 보존 기간을 줄이면 총 청구액을 줄이는 데 도움이 되나요?

보존 기간을 줄일 수는 있지만 이렇게 해도 비용이 크게 줄어들 가능성은 낮습니다.

보존 기간을 줄이거나 늘리려면 서비스 제한 요청을 제출하여 `Retention time for ingested data` 할당량을 변경할 수 있습니다.

알림 쿼리 비용을 낮게 유지하려면 어떻게 해야 하나요?

알림을 사용하면 데이터에 대해 쿼리가 생성되므로 쿼리 비용이 늘어납니다. 알림 쿼리를 최적화하고 비용을 낮추는 데 사용할 수 있는 몇 가지 전략은 다음과 같습니다.

- Prometheus용 Amazon Managed Service 알림 사용 — Prometheus용 Amazon Managed Service 외부의 경고 시스템은 외부 서비스가 여러 가용 영역 또는 지역에서 지표를 쿼리하므로 복원력 또는 고가용성을 추가하기 위해 추가 쿼리가 필요할 수 있습니다. 여기에는 고가용성에 대한 Grafana의 경고가 포함됩니다. 이로 인해 비용이 3배 이상 증가할 수 있습니다. Prometheus용 Amazon Managed Service의 알림은 최적화되어 있으며 가장 적은 수의 쿼리로 높은 가용성과 탄력성을 제공합니다.

  외부 알림 시스템 대신 Prometheus용 Amazon Managed Service의 네이티브 알림을 사용하는 것이 좋습니다.
- 알림 간격 최적화 - 알림 쿼리를 최적화하는 한 가지 빠른 방법은 자동 새로 고침 간격을 늘리는 것입니다. 1분마다 쿼리하지만 5분 간격으로만 필요한 알림이 있는 경우 자동 새로 고침 간격을 늘리면 해당 알림에 대한 쿼리 비용을 5배 절약할 수 있습니다.
- 최적의 룩백 사용 — 쿼리의 룩백 윈도우가 커지면 더 많은 데이터를 가져오므로 쿼리 비용이 증가합니다. PromQL 쿼리의 룩백 윈도우 크기가 알림이 필요한 데이터에 맞는지 확인하세요. 예를 들어, 다음 규칙에서 표현식에는 10분 룩백 윈도우가 포함됩니다.

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

  `expr`로 변경하면 쿼리 비용을 줄이는 데 도움이 될 `avg(rate(container_cpu_usage_seconds_total[5m])) > 0` 수 있습니다.

일반적으로 알림 규칙을 살펴보고 서비스에 가장 적합한 지표에 대해 알림을 보내고 있는지 확인하세요. 특히 시간이 지남에 따라 알림을 추가하는 경우 동일한 메트릭에 대해 중복되는 알림을 만들거나

동일한 정보를 제공하는 여러 알림을 쉽게 만들 수 있습니다. 여러 알림 그룹이 동시에 발생하는 경우가 많다면 알림을 모두 포함시키지 않고 최적화할 수 있습니다.

이러한 제안은 비용 절감에 도움이 될 수 있습니다. 궁극적으로 시스템 상태를 파악하기 위한 적절한 알림 세트를 생성하여 비용 균형을 맞춰야 합니다.

Prometheus용 Amazon 매니지드 서비스의 알림에 대한 자세한 내용은 을 참조하십시오. 알림 관리자

비용을 모니터링하기 위해 어떤 지표를 사용할 수 있나요?

`IngestionRate`Amazon에서 CloudWatch 모니터링하여 수집 비용을 추적하십시오. 에서 Prometheus용 Amazon 관리형 서비스 지표를 모니터링하는 방법에 대한 자세한 내용은 을 참조하십시오. CloudWatch CloudWatch 메트릭

언제든지 청구 내역을 확인할 수 있나요?

AWS 사용량을 AWS Cost and Usage Report 추적하고 청구 기간 내 계정과 관련된 예상 요금을 제공합니다. 자세한 내용은 AWS 비용 및 사용 보고서란 무엇입니까? 를 참조하십시오. AWS 비용 및 사용 보고서 사용 설명서에서

월초의 청구액이 월말보다 높은 이유는 무엇인가요?

Amazon Managed Service for Prometheus에는 수집에 대해 계층화된 요금 모델이 있으므로 초기 사용 비용이 더 많이 듭니다. 사용량이 더 높은 수집 티어에 도달하면 비용이 낮아지며 사용자에게 부과되는 비용도 낮아집니다. 수집 티어를 포함한 요금에 대한 자세한 내용은 Amazon Managed Service for Prometheus 제품 페이지의 요금을 참조하세요.

> **Note**
>
> - 등급은 지역 간 사용이 아닌 지역 내 사용을 위한 것입니다. 지역 내 사용량이 다음 등급에 도달해야 더 낮은 요금을 사용할 수 있습니다.
> - 의 AWS Organizations조직에서는 등급 사용량이 계정별이 아니라 지불자 계정별로 집계됩니다 (지급자 계정은 항상 조직 관리 계정임). 조직의 모든 계정에 대해 수집된 총 지표 (지역 내) 가 다음 등급에 도달하면 모든 계정에 더 낮은 요금이 부과됩니다.

Amazon Managed Service for Prometheus 작업 영역을 모두 삭제했지만 요금이 계속 청구되는 것 같습니다. 무슨 일이 벌어지고 있는 걸까요?

이 경우 한 가지 가능성은 삭제된 작업 영역에 지표를 전송하도록 설정된 스크레이퍼를 AWS 관리해 두고 있을 수 있다는 것입니다. 지침을 따르세요. 스크레이퍼 찾기 및 삭제

다른 AWS 서비스와 통합

Amazon Managed Service for Prometheus는 다른 AWS 서비스와 통합됩니다. 이 섹션에서는 Amazon Elastic Kubernetes Service(Amazon EKS) 비용 모니터링(Kubecost 사용)과의 통합과 Terraform 모듈에서 AWS Observability Accelerator를 사용한 완전한 관찰성 솔루션 생성에 대해 설명합니다.

주제

Amazon EKS 비용 모니터링과 통합

Amazon Managed Service for Prometheus는 Kubecost를 통한 Amazon Elastic Kubernetes Service(Amazon EKS) 비용 모니터링과 통합되어 비용 할당 계산을 수행하고 Kubernetes 클러스터 최적화에 대한 인사이트를 제공합니다. Kubecost를 통한 Amazon Managed Service for Prometheus를 사용하면 비용 모니터링을 안정적으로 확장하여 대규모 클러스터를 지원할 수 있습니다.

Kubecost와 통합하면 Amazon EKS 클러스터 비용을 보다 세밀하게 파악할 수 있습니다. 컨테이너 수준에서 클러스터 수준, 심지어 다중 클러스터 수준까지 대부분의 Kubernetes 컨텍스트별로 비용을 집계할 수 있습니다. 컨테이너 또는 클러스터 전반에서 보고서를 생성하여 다시 표시 또는 차지백 목적으로 비용을 추적할 수 있습니다.

다음은 단일 또는 다중 클러스터 시나리오에서 Kubecost와 통합하기 위한 지침을 제공합니다.

- 단일 클러스터 통합—Amazon EKS 비용 모니터링을 단일 클러스터와 통합하는 방법을 알아보려면 AWS 블로그 게시물 Kubecost와 Amazon Managed Service for Prometheus 통합을 참조하세요.
- 다중 클러스터 통합—Amazon EKS 비용 모니터링을 여러 클러스터와 통합하는 방법을 알아보려면 AWS 블로그 게시물 Kubecost 및 Amazon Managed Service for Prometheus를 사용한 Amazon EKS의 다중 클러스터 비용 모니터링을 참조하세요.

> **Note**
>
> Kubecost 사용에 대한 자세한 내용은 Amazon EKS 사용 설명서의 비용 모니터링을 참조하세요.

AWS Observability Accelerator 사용

AWS에서는 Amazon Elastic Kubernetes Service(Amazon EKS) 프로젝트를 위한 모니터링, 로깅, 알림 및 대시보드를 비롯한 관찰성 도구를 제공합니다. 여기에는 Amazon Managed Service for Prometheus, Amazon Managed Grafana, AWS Distro for OpenTelemetry 및 기타 도구가 포함됩니다. 이러한 도구를 함께 사용할 수 있도록 AWS에서는 이와 같은 서비스에서 관찰성을 구성하는 AWS Observability Accelerator라고 하는 Terraform 모듈을 제공합니다.

AWS Observability Accelerator는 인프라 모니터링, NGINX 배포 및 기타 시나리오에 대한 예제를 제공합니다. 이 섹션에서는 Amazon EKS 클러스터 내 인프라 모니터링의 예제를 제공합니다.

Terraform 템플릿과 자세한 지침은 Terraform용 AWS Observability Accelerator GitHub 페이지에서 확인할 수 있습니다. AWS Observability Accelerator를 발표하는 블로그 게시물을 읽어볼 수도 있습니다.

사전 조건

AWS Observability Accelerator를 사용하려면 기존 Amazon EKS 클러스터가 있어야 하고 다음과 같은 사전 요구 사항을 충족해야 합니다.

- AWS CLI—명령줄에서 AWS 기능을 호출하는 데 사용됩니다.
- kubectl—명령줄에서 EKS 클러스터를 제어하는 데 사용됩니다.
- Terraform—이 솔루션의 리소스 생성을 자동화하는 데 사용됩니다. AWS 계정 내에서 Amazon Managed Service for Prometheus, Amazon Managed Grafana, IAM을 생성하고 관리할 수 있는 액세스 권한이 있는 IAM 역할로 AWS 공급자를 설정해야 합니다. Terraform용 AWS 공급자를 구성하는 방법에 대한 자세한 내용은 Terraform 설명서의 AWS 공급자를 참조하세요.

인프라 모니터링 사용 예제

AWS Observability Accelerator는 포함된 Terraform 모듈을 사용하여 Amazon EKS 클러스터의 관찰성을 설정 및 구성하는 예제 템플릿을 제공합니다. 이 예제에서는 AWS Observability Accelerator를 사용하여 인프라 모니터링을 설정하는 방법을 보여 줍니다. 이 템플릿과 템플릿에 포함된 추가 기능의 사용

에 대한 자세한 내용은 GitHub의 기존 클러스터와 AWS Observability Accelerator 기반 및 인프라 모니터링 페이지를 참조하세요.

인프라 모니터링 Terraform 모듈을 사용하려면

1. 프로젝트를 생성하려는 폴더에서 다음 명령을 사용하여 리포지토리를 복제합니다.

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

2. 다음 명령을 사용하여 Terraform을 초기화합니다.

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

terraform init
```

3. 다음 예제와 같이 새 `terraform.tfvars` 파일을 생성합니다. Amazon EKS 클러스터에 대한 AWS 리전 및 클러스터 ID를 사용합니다.

```
# (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. 사용하려는 Amazon Managed Grafana 워크스페이스가 아직 없는 경우 생성합니다. 새 워크스페이스를 생성하는 방법에 대한 자세한 내용은 Amazon Managed Grafana 사용 설명서의 첫 번째 워크스페이스 생성을 참조하세요.
5. 명령줄에서 다음 명령을 실행하여 Terraform에서 Grafana 워크스페이스를 사용하기 위한 두 개의 변수를 생성합니다. *grafana-workspace-id*를 Grafana 워크스페이스의 ID로 대체해야 합니다.

```
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. [선택 사항] 기존의 Amazon Managed Service for Prometheus 워크스페이스를 사용하려면 다음 예와 같이 *prometheus-workspace-id*를 Prometheus 워크스페이스 ID로 대체하여 `terraform.tfvars` 파일에 ID를 추가합니다. 기존 워크스페이스를 지정하지 않으면 새 Prometheus 워크스페이스가 자동으로 생성됩니다.

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

7. 다음 명령을 사용하여 솔루션을 배포합니다.

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

이렇게 하면 AWS 계정에 다음을 포함한 리소스가 생성됩니다.

- 새 Amazon Managed Service for Prometheus 워크스페이스(기존 워크스페이스를 사용하기로 선택한 경우 제외)
- Prometheus 워크스페이스의 알림 관리자 구성, 알림 및 규칙
- 현재 워크스페이스의 새로운 Amazon Managed Grafana 데이터 소스 및 대시보드입니다. 데이터 소스는 `aws-observability-accelerator`로 지칭됩니다. 대시보드는 Observability Accelerator 대시보드 아래에 나열됩니다.
- 제공된 Amazon EKS 클러스터에서 Amazon Managed Service for Prometheus 워크스페이스로 지표를 전송하도록 AWS Distro for OpenTelemetry 운영자가 설정됩니다.

새 대시보드를 보려면 Amazon Managed Grafana 워크스페이스에서 특정 대시보드를 엽니다. Amazon Managed Grafana 사용에 대한 자세한 내용은 Amazon Managed Grafana 사용 설명서의 Grafana 워크스페이스에서 작업을 참조하세요.

쿠버네티스용 AWS 컨트롤러와 통합

Amazon Managed Service for Prometheus는 Kubernetes용 AWS 컨트롤러(ACK)와 통합되어 Amazon EKS의 워크스페이스, 알림 관리자 및 규칙 관리자 리소스의 관리를 지원합니다. 클러스터 외부의 리소스를 정의할 필요 없이 Kubernetes용 AWS 컨트롤러와 CRD (사용자 지정 리소스 정의) 및 네이티브 Kubernetes 객체를 사용할 수 있습니다.

이 섹션에서는 기존 Amazon EKS 클러스터에서 쿠버네티스용 AWS 컨트롤러와 Prometheus용 Amazon Managed Service를 설정하는 방법을 설명합니다.

쿠버네티스용 AWS 컨트롤러를 소개하고 Prometheus용 Amazon Managed Service용 ACK 컨트롤러를 소개하는 블로그 게시물도 읽을 수 있습니다.

필수 조건

쿠버네티스용 AWS 컨트롤러와 Prometheus용 Amazon Managed Service를 Amazon EKS 클러스터와 통합하기 전에 다음과 같은 사전 요구 사항이 있어야 합니다.

- Prometheus용 Amazon 관리 서비스 AWS 계정 및 IAM 역할을 프로그래밍 방식으로 생성하려면 기존 및 권한이 있어야 합니다.
- OpenID Connect(OIDC)가 활성화된 기존 Amazon EKS 클러스터가 있어야 합니다.

  OIDC가 활성화되지 않았다면 다음 명령을 사용하여 활성화할 수 있습니다. *YOUR_CLUSTER_NAME* 및 *AWS_REGION*을 계정에 맞는 올바른 값으로 바꿉니다.

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

  Amazon EKS에서 OIDC를 사용하는 방법에 대한 자세한 내용은 Amazon EKS 사용 설명서의 OIDC 자격 증명 공급자 인증 및 IAM OIDC 공급자 생성을 참조하세요.
- Amazon EKS 클러스터에 Amazon EBS CSI 드라이버가 설치되어 있어야 합니다.
- AWS CLI가 설치되어 있어야 합니다. 명령줄에서 AWS 기능을 AWS CLI 호출하는 데 사용됩니다.
- Kubernetes의 패키지 관리자인 Helm을 설치해야 합니다.
- Prometheus를 사용한 컨트롤 플레인 지표를 Amazon EKS 클러스터에서 설정해야 합니다.
- 새 워크스페이스에서 알림을 전송하려는 Amazon Simple Notification Service(Amazon SNS) 주제가 있어야 합니다. 주제에 메시지를 보낼 수 있는 권한을 Amazon Managed Service for Prometheus에 부여했는지 확인합니다.

Amazon EKS 클러스터가 적절하게 구성되면 `kubectl get --raw /metrics` 호출을 통해 Prometheus에 맞게 형식이 지정된 지표가 표시됩니다. 이제 Kubernetes용 AWS 컨트롤러 서비스 컨트롤러를 설치하고 이를 사용하여 Prometheus용 Amazon Managed Service 리소스를 배포할 준비가 되었습니다.

Kubernetes용 컨트롤러가 포함된 워크스페이스 배포 AWS

Prometheus용 Amazon Managed Service 작업 공간을 새로 배포하려면 Kubernetes용 컨트롤러 컨트롤러를 AWS 설치한 다음 이를 사용하여 작업 공간을 생성합니다.

쿠버네티스용 컨트롤러를 사용하여 Prometheus용 Amazon 관리 서비스 작업 공간을 새로 배포하려면 AWS

1. 다음 명령을 사용하여 Helm에서 Amazon Managed Service for Prometheus 서비스 컨트롤러를 설치합니다. 자세한 내용은 의 Kubernetes용 컨트롤러 설명서에 ACK 컨트롤러 설치를 참조하십시오. AWS GitHub 시스템에 맞는 *##*(예: `us-east-1`)을 사용합니다.

```
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
```

   몇 분 후 다음과 유사한 응답이 나타나는 것을 볼 수 있습니다.

```
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"
```

   다음 명령을 사용하여 Kubernetes용 AWS 컨트롤러가 성공적으로 설치되었는지 선택적으로 확인할 수 있습니다.

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

   그러면 `status: deployed`를 비롯한 컨트롤러 `ack-prometheusservice-controller`에 대한 정보가 반환됩니다.

2. 다음 텍스트를 사용하여 `workspace.yaml`이라는 파일을 생성합니다. 이 파일은 생성 중인 워크스페이스의 구성으로 사용됩니다.

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: Workspace
metadata:
  name: my-amp-workspace
```

```
spec:
  alias: my-amp-workspace
  tags:
    ClusterName: EKS-demo
```

3. 다음 명령을 실행하여 워크스페이스를 생성합니다(이 명령은 1단계에서 설정한 시스템 변수에 따라 달라짐).

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

잠시 후 계정에서 `my-amp-workspace`라는 새 워크스페이스를 볼 수 있을 것입니다.

다음 명령을 실행하면 워크스페이스 ID를 포함한 워크스페이스의 세부 정보 및 상태를 볼 수 있습니다. 또는 Amazon Managed Service for Prometheus 콘솔에서 새 워크스페이스를 볼 수도 있습니다.

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

> **Note**
>
> 워크스페이스를 생성하지 않고 기존 워크스페이스를 사용할 수도 있습니다.

4. Rulegroups의 구성으로 두 개의 새 yaml 파일을 생성하고 다음 구성을 사용하여 다음 구성을 사용하여 생성할 수 있습니다. AlertManager

이 구성을 `rulegroup.yaml`로 저장합니다. *`WORKSPACE-ID`*를 이전 단계의 워크스페이스 ID로 바꿉니다.

```
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 }}"
```

다음 구성을 `alertmanager.yaml`로 대체합니다. *`WORKSPACE-ID`*를 이전 단계의 워크스페이스 ID로 바꿉니다. *`TOPIC-ARN`*# 알림을 전송할 Amazon SNS 주제에 대한 ARN으로 *`###`* 사용 중인 지역으로 바꾸십시오. AWS 리전 Amazon Managed Service for Prometheus에는 Amazon SNS 주제에 대한 권한이 있어야 합니다.

```
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**
>
> 이러한 구성 파일의 형식에 대한 자세한 내용은 RuleGroupsNamespaceData 및 AlertManagerDefinitionData 섹션을 참조하세요.

5. 다음 명령을 실행하여 규칙 그룹 및 알림 관리자 구성을 생성합니다(이 명령은 1단계에서 설정한 시스템 변수에 따라 달라짐).

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

몇 분 이내에 변경 사항을 확인할 수 있습니다.

> **Note**
>
> 리소스를 생성하지 않고 업데이트하려면 yaml 파일을 업데이트하고 `kubectl apply` 명령을 다시 실행하면 됩니다.
> 리소스를 삭제하려면 다음 명령을 실행합니다. 삭제하려는 리소스 유형, 또는 *ResourceType*으로 바꾸십시오. `Workspace` `AlertManagerDefinition` `RuleGroupNamespace` 삭제할 리소스의 *ResourceName*이름으로 바꿉니다.
>
> ```
> kubectl delete ResourceType ResourceName -n $ACK_SYSTEM_NAMESPACE
> ```

이것으로 새 워크스페이스 배포가 완료됩니다. 다음 섹션에서는 해당 워크스페이스에 지표를 전송하도록 클러스터를 구성하는 방법을 설명합니다.

Amazon Managed Service for Prometheus 워크스페이스에 쓰도록 Amazon EKS 클러스터 구성

이 섹션에서는 Helm을 사용하여 Amazon EKS 클러스터에서 실행되는 Prometheus가 이전 섹션에서 생성한 Amazon Managed Service for Prometheus 워크스페이스에 지표를 원격으로 쓰도록 구성하는 방법을 설명합니다.

이 절차를 수행하려면 지표 수집에 사용하기 위해 생성한 IAM 역할의 이름이 필요합니다. 이 작업을 아직 수행하지 않은 경우 Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정에서 자세한 내용 및 지침을 참조하세요. 이러한 지침을 따르면 IAM 역할이 `amp-iamproxy-ingest-role`로 지칭됩니다.

Amazon EKS 클러스터에 대해 원격 쓰기를 구성하려면

1. 다음 명령을 사용하여 워크스페이스의 `prometheusEndpoint`를 가져옵니다. *`WORKSPACE-ID`*를 이전 섹션의 워크스페이스 ID로 대체합니다.

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

   prometheusEndpoint는 반환 결과에 표시되며 형식은 다음과 같습니다.

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

   다음 몇 단계에서 사용할 수 있도록 이 URL을 저장합니다.

2. 다음 텍스트로 새 파일을 생성하고 이름을 `prometheus-config.yaml`로 지정합니다. *`account`*를 계정 ID로, *`workspaceURL/`*을 방금 찾은 URL로, *`region`*을 시스템에 적합한 AWS 리전 으로 대체합니다.

   ```
   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. 다음 Helm 명령을 사용하여 Prometheus 차트 및 네임스페이스 이름과 차트 버전을 찾습니다.

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

지금까지 진행한 단계에 따라 Prometheus 차트와 네임스페이스의 이름을 모두 `prometheus`로 지정해야 하며 차트 버전은 `15.2.0`일 수 있습니다.

4. 이전 단계에서 *`PrometheusChartVersion`*찾은 *`PrometheusChartNamePrometheusNamespace`*, 및 를 사용하여 다음 명령을 실행합니다.

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

몇 분 후 업그레이드가 성공했다는 메시지가 표시됩니다.

5. 선택적으로 `awscurl`을 통해 Amazon Managed Service for Prometheus 엔드포인트를 쿼리하여 지표가 성공적으로 전송되고 있는지 확인할 수 있습니다. *`###`* 사용 중인 URL로 바꾸고 *`WorkspaceURL/# 1####`* 찾은 URL로 바꾸십시오. AWS 리전

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

이제 Amazon Managed Service for Prometheus 워크스페이스를 생성하고, YAML 파일을 구성으로 사용하여 Amazon EKS 클러스터에서 해당 워크스페이스에 연결했습니다. 사용자 지정 리소스 정의(CRD)라고 하는 이러한 파일은 Amazon EKS 클러스터 내에 있습니다. Kubernetes용 AWS 컨트롤러 컨트롤러를 사용하여 클러스터에서 직접 Prometheus용 Amazon 관리 서비스 리소스를 모두 관리할 수 있습니다.

CloudWatch 측정항목을 Firehose와 통합하기

이 섹션에서는 Amazon CloudWatch 메트릭 스트림을 계측하고 Amazon Data Firehose를 사용하는 방법과 Prometheus용 Amazon Managed Service에 지표를 수집하는 방법을 설명합니다. AWS Lambda

AWS Cloud Development Kit (CDK) 를 사용하여 스택을 설정하여 Firehose 전송 스트림, Lambda 및 Amazon S3 버킷을 생성하여 전체 시나리오를 시연해 보겠습니다.

인프라

가장 먼저 해야 할 일은 이 레시피의 인프라를 설정하는 것입니다.

CloudWatch 메트릭 스트림을 사용하면 스트리밍 지표 데이터를 HTTP 엔드포인트 또는 Amazon S3 버킷으로 전달할 수 있습니다.

인프라 설정은 다음 4단계로 구성됩니다.

- 사전 조건 구성
- Amazon Managed Service for Prometheus 워크스페이스 생성
- 종속성 설치
- 스택 배포

사전 조건

- 사용자 환경에 설치 및 구성됩니다. AWS CLI
- AWS CDK Typescript가 사용자 환경에 설치되어 있어야 합니다.
- Node.js 및 Go가 사용자 환경에 설치되어 있어야 합니다.
- AWS 옵저버빌리티 CloudWatch 메트릭 익스포터인 github 리포지토리 (`CWMetricsStreamExporter`) 가 로컬 시스템에 복제되었습니다.

Amazon Managed Service for Prometheus 워크스페이스를 생성하려면

1. 이 레시피의 데모 애플리케이션은 Amazon Managed Service for Prometheus에서 실행됩니다. 다음 명령을 사용하여 Amazon Managed Service for Prometheus 워크스페이스를 생성합니다.

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

2. 다음 명령으로 워크스페이스가 생성되었는지 확인하세요.

   ```
   aws amp list-workspaces
   ```

   Amazon Managed Service for Prometheus에 대한 자세한 내용은 Amazon Managed Service for Prometheus 사용 설명서를 참조하세요.

종속성을 설치하려면

1. 종속성 설치

aws-o11y-recipes 리포지토리의 루트에서 다음 명령을 사용하여 디렉터리를 CWMetricStreamExporter로 변경합니다.

```
cd sandbox/CWMetricStreamExporter
```

앞으로는 이 위치가 리포지토리의 루트로 간주될 것입니다.

2. 다음 명령을 사용하여 디렉터리를 /cdk로 변경합니다.

```
cd cdk
```

3. 다음 명령을 실행하여 CDK 종속성을 설치합니다.

```
npm install
```

4. 디렉터리를 리포지토리의 루트로 다시 변경한 후 다음 명령을 사용하여 디렉터리를 다시 /lambda로 변경합니다.

```
cd lambda
```

5. /lambda 폴더에 들어가면 다음을 사용하여 Go 종속성을 설치합니다.

```
go get
```

이제 모든 종속성이 설치되었습니다.

스택을 배포하려면

1. 리포지토리의 루트에서 config.yaml을 열고 {workspace}를 새로 생성한 워크스페이스 ID와 Amazon Managed Service for Prometheus 워크스페이스가 있는 리전으로 대체하여 Amazon Managed Service for Prometheus 워크스페이스 URL을 수정합니다.

   예를 들어 다음을 수정합니다.

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

Firehose 전송 스트림과 Amazon S3 버킷의 이름을 원하는 대로 변경합니다.

2. AWS CDK 및 Lambda 코드를 빌드하려면 리포지토리의 루트에서 다음 명령을 실행합니다.

```
npm run build
```

이 빌드 단계는 Go Lambda 바이너리가 빌드되고 CDK를 배포하도록 합니다. CloudFormation

3. 배포를 완료하려면 스택에 필요한 IAM 변경 사항을 검토하고 수락해야 합니다.
4. (선택 사항) 다음 명령을 실행하여 스택이 생성되었는지 확인할 수 있습니다.

```
aws cloudformation list-stacks
```

이름이 CDK Stack인 스택이 목록에 표시됩니다.

아마존 CloudWatch 스트림 생성

이제 지표를 처리하는 Lambda 함수가 생겼으므로 Amazon에서 지표 스트림을 생성할 수 있습니다. CloudWatch

지표 스트림을 CloudWatch 생성하려면

1. https://console.aws.amazon.com/cloudwatch/home#metric-streams:streamsList CloudWatch 콘솔로 이동하여 메트릭 스트림 생성을 선택합니다.
2. 필요한 지표(모든 지표 또는 선택한 네임스페이스의 지표)를 선택합니다.
3. Configuration에서 계정에서 소유한 기존 Firehose 선택을 선택합니다.
4. CDK에서 이전에 만든 Firehose를 사용하게 됩니다. Kinesis Data Firehose 전송 스트림 선택 드롭다운에서 이전에 만든 스트림을 선택합니다. 이름은 CdkStack-KinesisFirehoseStream123456AB-sample1234와 같습니다.
5. 출력 형식을 JSON으로 변경합니다.
6. 지표 스트림에 의미 있는 이름을 지정합니다.
7. 지표 스트림 생성을 선택합니다.
8. (선택 사항) Lambda 함수 호출을 확인하려면 Lambda 콘솔로 이동하고 함수 KinesisMessageHandler를 선택합니다. 모니터링 탭과 로그 하위 탭을 선택하면 최근 호출 아래에 트리거되는 Lambda 함수 항목이 표시됩니다.

> **Note**
>
> 모니터링 탭에 호출이 표시되기 시작하는 데 최대 5분이 걸릴 수 있습니다.

이제 지표가 Amazon에서 Prometheus용 아마존 CloudWatch 매니지드 서비스로 스트리밍되고 있습니다.

정리

이 예제에서 사용된 리소스를 정리할 수 있습니다. 다음 절차에서는 정리하는 방법을 설명합니다. 이렇게 하면 생성한 지표 스트림이 중지됩니다.

리소스를 정리하려면

1. 먼저 다음 명령을 사용하여 CloudFormation 스택을 삭제하십시오.

```
cd cdk
cdk destroy
```

2. Amazon Managed Service for Prometheus 워크스페이스를 제거합니다.

```
aws amp delete-workspace --workspace-id \
    `aws amp list-workspaces --alias prometheus-sample-app --query
 'workspaces[0].workspaceId' --output text`
```

3. 마지막으로 Amazon CloudWatch 콘솔을 사용하여 Amazon CloudWatch 메트릭 스트림을 제거합니다.

Amazon Managed Service for Prometheus의 보안

AWS에서 클라우드 보안은 가장 중요합니다. AWS 고객은 보안에 가장 보안에 민감한 조직의 요구 사항에 부합하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 AWS와 사용자의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드의 보안: AWS는 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS는 안전하게 사용할 수 있는 서비스 또한 제공합니다. 서드 파티 감사자는 AWS 규정 준수 프로그램의 일환으로 보안 효과를 정기적으로 테스트하고 검증합니다. Amazon Managed Service for Prometheus에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램을 통한 범위 내 AWS 서비스를 참조하세요.
- 클라우드 내 보안: 귀하의 책임은 귀하가 사용하는 AWS 서비스에 의해 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Amazon Managed Service for Prometheus를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목적에 맞게 Amazon Managed Service for Prometheus를 구성하는 방법을 보여줍니다. 또한 Amazon Managed Service for Prometheus 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법을 알아봅니다.

주제

- Amazon Managed Service for Prometheus의 데이터 보호
- Amazon Managed Service for Prometheus용 Identity and Access Management
- IAM 권한 및 정책
- Amazon Managed Service for Prometheus에 대한 규정 준수 확인
- Amazon Managed Service for Prometheus의 복원력
- Amazon Managed Service for Prometheus의 인프라 보안
- Amazon Managed Service for Prometheus에 대한 서비스 연결 역할
- AWS CloudTrail을 사용하여 Amazon Managed Service for Prometheus API 호출 로깅
- 서비스 계정에 대한 IAM 역할 설정
- 인터페이스 VPC 엔드포인트에서 Amazon Managed Service for Prometheus 사용

Amazon Managed Service for Prometheus의 데이터 보호

AWS 공동 책임 모델은 Amazon Managed Service for Prometheus의 데이터 보호에 적용됩니다. 이 모델이 설명하는 것처럼 AWS는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQ를 참조하십시오. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터를 보호하려면 AWS 계정 보안 인증 정보를 보호하고 AWS IAM Identity Center 또는 AWS Identity and Access Management(IAM)을 통해 개별 사용자 계정을 설정하는 것이 좋습니다. 이 방식을 사용하면 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다중 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2가 필수이며 TLS 1.3을 권장합니다.
- AWS CloudTrail로 API 및 사용자 활동 로깅을 설정합니다.
- AWS 암호화 솔루션을 AWS 서비스 내의 모든 기본 보안 컨트롤과 함께 사용합니다.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 API를 통해 AWS에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 Federal Information Processing Standard(FIPS) 140-2 섹션을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API, AWS CLI 또는 AWS SDK를 사용하여 Amazon Managed Service for Prometheus 또는 기타 AWS 서비스에서 작업하는 경우가 포함됩니다. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 자격 보안 인증를 URL에 포함시켜서는 안 됩니다.

주제

- Amazon Managed Service for Prometheus에서 수집된 데이터
- 저장 시 암호화

Amazon Managed Service for Prometheus에서 수집된 데이터

Amazon Managed Service for Prometheus는 사용자 계정에서 실행되는 Prometheus 서버에서 Amazon Managed Service for Prometheus로 전송하도록 구성한 운영 지표를 수집하여 저장합니다. 이 데이터에는 다음이 포함됩니다.

- 지표 값
- 데이터를 식별하고 분류하는 데 도움이 되는 지표 레이블(또는 임의의 키-값 쌍)
- 데이터 샘플의 타임스탬프

고유한 테넌트 ID는 서로 다른 고객으로부터 데이터를 격리합니다. 이러한 ID는 액세스할 수 있는 고객 데이터를 제한합니다. 고객은 테넌트 ID를 변경할 수 없습니다.

Amazon Managed Service for Prometheus는 AWS Key Management Service(AWS KMS) 키로 저장하는 데이터를 암호화합니다. Amazon Managed Service for Prometheus는 이러한 키를 관리합니다.

> **Note**
>
> Amazon Managed Service for Prometheus는 고객 관리형 키 생성을 지원하지 않습니다. Amazon Managed Service for Prometheus는 매우 민감한 데이터를 저장하기 위한 것이 아닙니다. 서버 측 데이터는 사용자를 대신하여 AWS 관리형 키를 사용하여 암호화됩니다. 이러한 키에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 관리형 키를 참조하십시오.

전송 중 데이터는 HTTPS를 사용하여 자동으로 암호화됩니다. Amazon Managed Service for Prometheus는 내부적으로 HTTPS를 사용하여 AWS 리전 내 가용 영역 간 연결을 보호합니다.

저장 시 암호화

기본적으로 Amazon Managed Service for Prometheus는 저장 데이터 암호화를 자동으로 제공하며, 이 작업은 AWS 소유 암호화 키를 사용하여 수행합니다.

- AWS 소유 키 - Amazon Managed Service for Prometheus는 이러한 키를 사용하여 WorkSpace에 업로드된 데이터를 자동으로 암호화합니다. 사용자는 AWS 소유 키를 보거나 관리하거나 사용할 수 없으며 해당 키의 사용을 감사할 수 없습니다. 하지만 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 소유 키를 참조하십시오.

저장 데이터 암호화는 개인 식별 정보와 같은 민감한 고객 데이터를 보호하는 데 필요한 운영 오버헤드와 복잡성을 줄이는 데 도움이 됩니다. 또한 이 기능을 통해 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다.

또는 WorkSpace를 생성할 때 고객 관리형 키를 사용하도록 선택할 수도 있습니다.

- 고객 관리형 키 - Amazon Managed Service for Prometheus는 사용자가 생성하고 소유하고 관리하는 대칭형 고객 관리형 키를 사용하여 WorkSpace의 데이터를 암호화할 수 있도록 지원합니다. 이 암호화를 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.
  - 키 정책 수립 및 유지
  - IAM 정책 및 권한 부여 수립 및 유지
  - 키 정책 활성화 및 비활성화
  - 키 암호화 자료 교체
  - 태그 추가
  - 키 별칭 생성
  - 삭제를 위한 키 예약

  자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키를 참조하십시오.

고객 관리형 키를 사용할지 아니면 AWS 소유 키를 사용할지 신중하게 선택합니다. 고객 관리형 키로 생성한 WorkSpace는 나중에 AWS 소유 키를 사용하도록 전환할 수 없으며, 그 반대의 경우도 마찬가지입니다.

> **Note**
>
> Amazon Managed Service for Prometheus에서는 AWS 소유 키를 사용하여 저장 데이터 암호화를 활성화하여 추가 비용 없이 데이터를 보호합니다.
> 그러나 고객 관리형 키 사용에는 AWS KMS 비용이 부과됩니다. 요금에 대한 자세한 내용은 AWS Key Management Service 요금을 참조하십시오.

AWS KMS에 대한 자세한 내용은 AWS Key Management Service 소개를 참조하십시오.

> **Note**
>
> 고객 관리형 키로 생성된 WorkSpace에서는 수집에 AWS 관리형 수집기를 사용할 수 없습니다.

Amazon Managed Service for Prometheus가 AWS KMS에서 권한 부여를 사용하는 방법

Amazon Managed Service for Prometheus에서 고객 관리형 키를 사용하려면 세 가지 권한 부여가 필요합니다.

고객 관리 키로 암호화된 Prometheus용 Amazon 관리 서비스 작업 공간을 생성하면 Prometheus용 Amazon Managed Service에서 요청을 보내 사용자를 대신하여 세 가지 권한 부여를 생성합니다. CreateGrantAWS KMS AWS KMS의 권한 부여는 Amazon Managed Service for Prometheus가 사용자 계정의 KMS 키에 대한 액세스 권한을 부여하는 데 사용되며, 이는 사용자를 대신하여 직접 호출하지 않는 경우에도 마찬가지입니다(예: Amazon EKS 클러스터에서 스크래핑한 지표 데이터를 저장하는 경우).

Amazon Managed Service for Prometheus는 다음 내부 작업에 대해 고객 관리형 키를 사용하기 위한 권한 부여가 필요합니다.

- DescribeKey요청을 보내 작업 공간을 AWS KMS 생성할 때 제공한 대칭 고객 관리형 KMS 키가 유효한지 확인하십시오.
- 고객 관리 키로 암호화된 데이터 키를 생성해 달라는 GenerateDataKey 요청을 AWS KMS로 보내세요.
- 데이터를 암호화하는 데 사용할 수 있도록 암호화된 데이터 키를 해독하려면 해독 요청을 AWS KMS에 전송합니다.

Amazon Managed Service for Prometheus는 Amazon Managed Service for Prometheus가 사용자를 대신하여 키를 사용할 수 있도록 AWS KMS 키에 세 가지 권한 부여를 생성합니다. 키 정책을 변경하거나, 키를 비활성화하거나, 권한 부여를 취소하여 키에 대한 액세스 권한을 제거할 수 있습니다. 이러한 작업을 수행하기 전에 이러한 작업의 결과를 이해해야 합니다. 이로 인해 WorkSpace의 데이터가 손실될 수 있습니다.

어떤 방식으로든 권한 부여에 대한 액세스 권한을 제거하면 Amazon Managed Service for Prometheus는 고객 관리형 키로 암호화된 데이터에 액세스할 수 없고 WorkSpace로 전송된 새 데이터

를 저장할 수도 없으며, 이는 해당 데이터에 의존하는 작업에 영향을 미칩니다. WorkSpace로 전송된 새 데이터는 액세스할 수 없으며 영구적으로 손실될 수 있습니다.

> ⚠ Warning
>
> - 키 정책에서 키를 비활성화하거나 Amazon Managed Service for Prometheus 액세스를 제거하면 WorkSpace 데이터에 더 이상 액세스할 수 없습니다. WorkSpace로 전송되는 새 데이터는 액세스할 수 없으며 영구적으로 손실될 수 있습니다.
>
>   Amazon Managed Service for Prometheus의 키 액세스를 복원하여 WorkSpace 데이터에 액세스하고 새 데이터를 다시 수신할 수 있습니다.
> - 권한을 취소하면 다시 생성할 수 없으며 WorkSpace의 데이터가 영구적으로 손실됩니다.

1단계: 고객 관리형 키 생성

AWS Management Console 또는 AWS KMS API를 사용하여 대칭형 고객 관리형 키를 생성할 수 있습니다. 아래 설명과 같이 정책을 통해 올바른 액세스 권한을 제공하기만 하면 키는 Amazon Managed Service for Prometheus WorkSpace와 동일한 계정에 있지 않아도 됩니다.

대칭형 고객 관리형 키를 생성하려면

AWS Key Management Service 개발자 안내서의 대칭형 고객 관리형 키 생성 단계를 따르십시오.

키 정책

키 정책은 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 생성할 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키에 대한 액세스 관리를 참조하십시오.

Amazon Managed Service for Prometheus WorkSpace에서 고객 관리형 키를 사용하려면 키 정책에서 다음 API 작업을 허용해야 합니다.

- `kms:CreateGrant` — 고객 관리형 키에 권한 부여를 추가합니다. 지정된 KMS 키에 대한 제어 액세스 권한을 부여합니다. 이를 통해 Amazon Managed Service for Prometheus에 필요한 권한 부여 작업에 대한 액세스가 허용됩니다. 자세한 내용은 AWS Key Management Service 개발자 안내서에서 권한 부여 사용을 참조하십시오.

  Amazon Managed Service for Prometheus는 이를 통해 다음을 수행할 수 있습니다.

  - 데이터 키가 암호화에 즉시 사용되지 않으므로 암호화된 데이터 키를 생성하여 저장하려면 `GenerateDataKey`를 직접적으로 호출합니다.
  - 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하려면 `Decrypt`를 직접적으로 호출합니다.
- `kms:DescribeKey` - Amazon Managed Service for Prometheus에서 키를 검증할 수 있도록 고객 관리형 키 세부 정보를 제공합니다.

다음은 Amazon Managed Service for Prometheus에 추가할 수 있는 정책 설명 예시입니다.

```
 "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>
 ]
```

- 정책에서 사용 권한을 지정하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하십시오.
- 키 액세스 문제 해결에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하십시오.

2단계: Prometheus용 Amazon 관리 서비스의 고객 관리 키 지정

WorkSpace를 생성할 때 KMS Key ARN을 입력하여 고객 관리형 키를 지정할 수 있습니다. 이는 Amazon Managed Service for Prometheus에서 WorkSpace에 저장된 데이터를 암호화하는 데 사용합니다.

3단계: Amazon Managed Grafana와 같은 다른 서비스에서 데이터에 액세스

이 단계는 선택 사항이며, 다른 서비스에서 Prometheus용 Amazon 관리 서비스 데이터에 액세스해야 하는 경우에만 필요합니다.

다른 서비스에서 키를 사용할 수 있는 액세스 권한도 없는 한 암호화된 데이터에는 액세스할 수 없습니다. AWS KMS 예를 들어, Amazon Managed Grafana를 사용하여 데이터에 대한 대시보드 또는 알림을 생성하려면 Amazon Managed Grafana에 해당 키에 대한 액세스 권한을 부여해야 합니다.

Amazon Managed Grafana에 고객 관리 키에 대한 액세스 권한을 부여하려면

1. 아마존 관리형 Grafana 워크스페이스 목록에서 Prometheus용 아마존 매니지드 서비스에 액세스하려는 워크스페이스의 이름을 선택합니다. Amazon 관리형 Grafana 워크스페이스에 대한 요약 정보를 보여줍니다.
2. 워크스페이스에서 사용하는 IAM 역할의 이름을 기록해 두십시오. 이름은 다음과 같은 `AmazonGrafanaServiceRole-<unique-id>` 형식입니다. 콘솔에는 해당 역할의 전체 ARN이 표시됩니다. 이후 단계에서 AWS KMS 콘솔에서 이 이름을 지정하게 됩니다.
3. AWS KMS고객 관리 키 목록에서 Prometheus용 Amazon 관리 서비스 작업 공간을 생성할 때 사용한 고객 관리 키를 선택합니다. 그러면 키 구성 세부 정보 페이지가 열립니다.
4. 주요 사용자 옆의 추가 버튼을 선택합니다.
5. 이름 목록에서 위에서 언급한 Amazon 관리형 Grafana IAM 역할을 선택합니다. 이름을 기준으로 검색할 수도 있습니다. 더 쉽게 찾을 수 있습니다.

6. 추가를 선택하여 IAM 역할을 주요 사용자 목록에 추가합니다.

이제 아마존 매니지드 Grafana 워크스페이스에서 Prometheus용 아마존 매니지드 서비스 워크스페이스의 데이터에 액세스할 수 있습니다. 주요 사용자에게 다른 사용자 또는 역할을 추가하여 다른 서비스가 작업 공간에 액세스할 수 있도록 할 수 있습니다.

Amazon Managed Service for Prometheus 암호화 컨텍스트

암호화 컨텍스트는 데이터에 대한 추가 컨텍스트 정보를 포함하는 선택적 키-값 페어 세트입니다.

AWS KMS는 암호화 컨텍스트를 추가 인증 데이터로 사용하여 인증된 암호화를 지원합니다. 데이터 암호화 요청에 암호화 컨텍스트를 포함하는 경우, AWS KMS는 암호화된 데이터에 암호화 컨텍스트를 바인딩합니다. 데이터 복호화를 위해, 이 요청에 동일한 암호화 컨텍스트를 포함합니다.

Amazon Managed Service for Prometheus 암호화 컨텍스트

Amazon Managed Service for Prometheus는 모든 AWS KMS 암호화 작업에서 동일한 암호화 컨텍스트를 사용합니다. 여기서 키는 `aws:amp:arn`이고 값은 WorkSpace의 Amazon 리소스 이름(ARN)입니다.

Example

```
"encryptionContext": {
    "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-sample-1234-
abcd-56ef-7890abcd12ef"
}
```

모니터링을 위한 암호화 컨텍스트 사용

대칭형 고객 관리형 키를 사용하여 WorkSpace 데이터를 암호화하는 경우 감사 레코드 및 로그의 암호화 컨텍스트를 사용하여 고객 관리형 키가 사용되는 방식을 식별할 수도 있습니다. 암호화 컨텍스트는 AWS CloudTrail또는 Amazon Logs에서 생성한 CloudWatch 로그에도 나타납니다.

암호화 컨텍스트를 사용하여 고객 관리형 키에 대한 액세스 제어

그러나 키 정책 및 IAM 정책의 암호화 컨텍스트를 `conditions`로 사용하여 대칭형 고객 관리형 키에 대한 액세스를 제어할 수도 있습니다. 또한 권한 부여에서 암호화 컨텍스트 제약 조건을 사용할 수 있습니다.

Amazon Managed Service for Prometheus는 권한 부여에서 암호화 컨텍스트 제약 조건을 사용하여 계정 및 리전에서 고객 관리형 키에 대한 액세스를 제어합니다. 권한 부여 제약 조건에 따라 권한 부여가 허용하는 작업은 지정된 암호화 컨텍스트를 사용해야 합니다.

Example

다음은 특정 암호화 컨텍스트에서 고객 관리형 키에 대한 액세스 권한을 부여하는 키 정책 설명의 예시입니다. 이 정책 설명의 조건에 따라 권한 부여에는 암호화 컨텍스트를 지정하는 암호화 컨텍스트 제약 조건이 있어야 합니다.

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

Amazon Managed Service for Prometheus에 사용되는 암호화 키 모니터링

Prometheus용 Amazon 관리 서비스 작업 영역에서 AWS KMS 고객 관리 키를 사용하는 경우 Amazon CloudWatch Logs를 사용하여 AWS CloudTrailPrometheus용 Amazon 관리 서비스가 보내는 요청을 추적할 수 있습니다. AWS KMS

다음 예는 고객 관리형 키로 암호화된 데이터에 액세스하기 위해 Amazon Managed Service for Prometheus에서 호출한 KMS 작업을 모니터링하기 위한 `CreateGrant`, `GenerateDataKey`, `Decrypt`, `DescribeKey`에 대한 AWS CloudTrail 이벤트입니다.

CreateGrant

AWS KMS 고객 관리형 키를 사용하여 WorkSpace을 암호화하면 Amazon Managed Service for Prometheus에서 사용자를 대신하여 사용자가 지정한 KMS 키에 액세스하기 위한 세 가지 `CreateGrant` 요청을 보냅니다. Amazon Managed Service for Prometheus에서 생성하는 권한 부여는 AWS KMS 고객 관리형 키와 연결된 리소스에만 적용됩니다.

다음 예제 이벤트는 `CreateGrant` 작업을 기록합니다.


```
{
    "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

WorkSpace에 AWS KMS 고객 관리형 키를 활성화하면 Amazon Managed Service for Prometheus에서 고유한 키를 생성합니다. 리소스에 대한 AWS KMS 고객 관리형 키를 지정하는 `GenerateDataKey` 요청을 AWS KMS에 보냅니다.

다음 예제 이벤트는 `GenerateDataKey` 작업을 기록합니다.

```
{
    "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

암호화된 WorkSpace에서 쿼리가 생성되면 Amazon Managed Service for Prometheus는 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하는 `Decrypt` 작업을 호출합니다.

다음 예제 이벤트는 `Decrypt` 작업을 기록합니다.

```
{
    "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

Amazon Managed Service for Prometheus는 `DescribeKey` 작업을 사용하여 WorkSpace와 관련된 AWS KMS 고객 관리형 키가 계정 및 지역에 존재하는지 확인합니다.

다음 예제 이벤트는 `DescribeKey` 작업을 기록합니다.


```
{
    "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"
}
```

자세히 알아보기

다음 리소스에서 저장 데이터 암호화에 대한 추가 정보를 확인할 수 있습니다.

- AWS Key Management Service기본 개념에 대한 자세한 내용은 AWS Key Management Service개발자 안내서를 참조하십시오.
- AWS Key Management Service의 보안 모범 사례에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서를 참조하십시오.

Amazon Managed Service for Prometheus용 Identity and Access Management

AWS Identity and Access Management(IAM)은 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어할 수 있도록 지원하는 AWS 서비스입니다. IAM 관리자는 어떤 사용자가 Amazon Managed Service for Prometheus 리소스를 사용할 수 있도록 인증(로그인)되고 권한이 부여(권한 있음)될 수 있는지 제어합니다. IAM은 추가 비용 없이 사용할 수 있는 AWS 서비스입니다.

주제

- 고객
- ID를 통한 인증
- 정책을 사용한 액세스 관리
- Amazon Managed Service for Prometheus가 IAM에서 작동하는 방식

- Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제
- Amazon Managed Service for Prometheus에 사용되는 AWS 관리형 정책
- Amazon Managed Service for Prometheus ID 및 액세스 문제 해결

고객

AWS Identity and Access Management(IAM)를 사용하는 방법은 Amazon Managed Service for Prometheus에서 수행하는 작업에 따라 달라집니다.

서비스 사용자 – Amazon Managed Service for Prometheus 서비스를 사용하여 작업을 수행하는 경우 필요한 자격 증명과 권한을 관리자가 제공합니다. 더 많은 Amazon Managed Service for Prometheus 기능을 사용하여 작업을 수행한다면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. Amazon Managed Service for Prometheus의 기능에 액세스할 수 없는 경우 Amazon Managed Service for Prometheus ID 및 액세스 문제 해결 섹션을 참조하십시오.

서비스 관리자 – 회사에서 Amazon Managed Service for Prometheus 리소스를 담당하고 있다면 Amazon Managed Service for Prometheus에 대한 완전한 액세스 권한이 있을 것입니다. 서비스 관리자는 서비스 사용자가 액세스해야 하는 Amazon Managed Service for Prometheus 기능과 리소스를 결정합니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해합니다. 회사가 Amazon Managed Service for Prometheus에서 IAM을 사용하는 방법에 대해 자세히 알아보려면 Amazon Managed Service for Prometheus가 IAM에서 작동하는 방식 섹션을 참조하십시오.

IAM 관리자 - IAM 관리자라면 Amazon Managed Service for Prometheus에 대한 액세스 관리 정책 작성 방법을 자세히 알아야 할 것입니다. IAM에서 사용할 수 있는 Amazon Managed Service for Prometheus ID 기반 정책 예제를 보려면 Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제 섹션을 참조하십시오.

ID를 통한 인증

인증은 ID 보안 인증을 사용하여 AWS에 로그인하는 방식입니다. AWS 계정 루트 사용자 또는 IAM 사용자로 또는 IAM 역할을 수임하여 인증(AWS에 로그인)되어야 합니다.

자격 증명 소스를 통해 제공된 보안 인증 정보를 사용하여 페더레이션 ID로 AWS에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증, Google 또는 Facebook 보안 인증이 페더레이션형 ID의 예입니다. 페더레이션형 ID로 로그인할 때 관리자가 이전

에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 연동을 사용하여 AWS에 액세스하면 간접적으로 역할을 수임합니다.

사용자 유형에 따라 AWS Management Console 또는AWS 액세스 포털에 로그인할 수 있습니다. AWS에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 계정에 로그인하는 방법을 참조하십시오.

AWS에 프로그래밍 방식으로 액세스하는 경우, AWS는 보안 인증 정보를 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK) 및 명령줄 인터페이스(CLI)를 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 요청에 직접 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS는 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하십시오.

AWS 계정 루트 사용자

AWS 계정을 생성할 때는 해당 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 단일 로그인 자격 증명으로 시작합니다. 이 자격 증명은 AWS 계정루트 사용자라고 하며, 계정을 생성할 때 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 태스크에는 루트 사용자를 가급적 사용하지 않는 것이 좋습니다. 루트 사용자 자격 증명을 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 태스크의 전체 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 태스크 섹션을 참조하십시오.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 사용자가 자격 증명 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스에 액세스하도록 요구합니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, AWS Directory Service, Identity Center 디렉터리의 사용자 또는 자격 증명 소스를 통해 제공된 보안 인증을 사용하여 AWS 서비스에 액세스하는 모든 사용자입니다. 페더레이션 자격 증명은 AWS 계정에 액세스할 때 역할을 수임하고 역할은 임시 보안 인증을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을(를) 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 AWS 계정및 애플리케이션에서 사용하기 위해 고유한 자격 증명 소스의 사용자 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한

자세한 내용은 AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 개인 또는 애플리케이션에 대한 특정 권한을 가지고 있는 AWS 계정에 속하는 ID입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체 섹션을 참조하십시오.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증을 가지고 있지만 역할은 임시 보안 인증만 제공합니다. 자세한 정보는 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님) 섹션을 참조하십시오.

IAM 역할

IAM 역할은 특정 권한을 가지고 있는 AWS 계정 내의 ID입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할 전환하여 AWS Management Console에서 IAM 역할을 임시로 수임할 수 있습니다. AWS CLI 또는 AWS API 태스크를 직접적으로 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM 역할 사용을 참조하십시오.

보안 인증 정보가 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 자격 증명에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 자격 증명이 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 타사 자격 증명 공급자의 역할 생성을 참조하십시오. IAM Identity Center를 사용하는 경우 권한 집합을 구성합니다. 인증 후 아이덴티티가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관 짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트 섹션을 참조하십시오.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.

- 크로스 계정 액세스: IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스를 사용하면 정책을 리소스에 직접 연결할 수 있습니다(역할을 프록시로 사용하는 대신). 교차 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이 섹션을 참조하십시오.
- 교차 서비스 액세스 - 일부 AWS 서비스는 다른 AWS 서비스의 기능을 사용합니다. 예를 들어 서비스에서 직접적으로 호출하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 직접 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 태스크를 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하십시오.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.
  - 서비스 연결 역할 - 서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 유형입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon EC2에서 실행 중인 애플리케이션 – IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 보안 인증을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS역할을 할당하고 해당 역할을 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여 섹션을 참조하십시오.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우 섹션을 참조하십시오.

정책을 사용한 액세스 관리

정책을 생성하고 AWS 자격 증명 또는 리소스에 연결하여 AWS 내 액세스를 제어합니다. 정책은 ID 또는 리소스와 연결될 때 해당 권한을 정의하는 AWS의 개체입니다. AWS는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청을 보낼 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되는지 또는 거부되는지를 결정합니다. 대부분의 정책은 AWS에 JSON 설명서로서 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요 섹션을 참조하십시오.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI또는 AWSAPI에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 AWS 계정에 속한 다수의 사용자, 그룹 및 역할에 독립적으로 추가할 수 있는 정책입니다. 관리형 정책에는 AWS관리형 정책과 고객 관리형 정책이 포함되어 있습니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하십시오.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니

다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스이(가) 포함될 수 있습니다.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책에서는 IAM의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACLs)

액세스 제어 목록(ACLs)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는 지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3, AWS WAF 및 Amazon VPC는 ACL을 지원하는 대표적인 서비스입니다. ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 ACL(액세스 제어 목록) 개요 섹션을 참조하십시오.

기타 정책 유형

AWS은(는) 비교적 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 유형은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 ID 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 개체의 ID 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계 섹션을 참조하십시오.
- 서비스 제어 정책(SCP) - SCP는 AWS Organizations에서 조직 또는 조직 단위(OU)에 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations는 기업이 소유하는 여러 개의 AWS 계정을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 각 AWS 계정 루트 사용자을(를) 비롯하여 멤버 계정의 엔터티에 대한 권한을 제한합니다. 조직 및 SCP에 대한 자세한 정보는 AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하십시오.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 ID 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하십시오.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련될 때 AWS가 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

Amazon Managed Service for Prometheus가 IAM에서 작동하는 방식

IAM을 사용하여 Amazon Managed Service for Prometheus에 대한 액세스를 관리하기 전에 Amazon Managed Service for Prometheus에서 사용할 수 있는 IAM 기능에 대해 알아봅니다.

Amazon Managed Service for Prometheus에서 사용할 수 있는 IAM 기능

IAM 특성	Amazon Managed Service for Prometheus 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예
정책 조건 키	아니요
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
전달 액세스 세션(FAS)	아니요
서비스 역할	아니요
서비스 연결 역할	예

Amazon Managed Service for Prometheus 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하십시오.

Amazon Managed Service for Prometheus에 대한 ID 기반 정책

ID 기반 정책 지원	예

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 ID에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 작업와 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조 섹션을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제

Amazon Managed Service for Prometheus ID 기반 정책의 예제를 보려면 Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제 섹션을 참조하십시오.

Amazon Managed Service for Prometheus 내의 리소스 기반 정책

리소스 기반 정책 지원	아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스이(가) 포함될 수 있습니다.

교차 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념합니다. 보안 주체와 리소스가 서로 다른 AWS 계정에 있는 경우 신뢰할 수 있는 계정의 IAM 관리자는 보안 주체 개체(사용자 또는 역할)에도 리소스 액세스 권한을 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기

반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우 추가 ID 기반 정책이 필요하지 않습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이 섹션을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 정책 작업

정책 작업 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 일반적으로 정책 작업의 이름은 연결된 AWS API 작업의 이름과 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함합니다.

Amazon Managed Service for Prometheus 작업 목록을 보려면 서비스 승인 참조의 Amazon Managed Service for Prometheus에서 정의한 작업을 참조하십시오.

Amazon Managed Service for Prometheus의 정책 작업은 작업 앞에 다음 접두사를 사용합니다.

```
aps
```

단일 명령문에서 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "aps:action1",
      "aps:action2"
         ]
```

Amazon Managed Service for Prometheus ID 기반 정책의 예제를 보려면 Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제 섹션을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 정책 리소스

정책 리소스 지원	예

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 명령문에는 `Resource` 또는 `NotResource` 요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

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

Amazon Managed Service for Prometheus 리소스 유형 및 해당 ARN 목록을 보려면 서비스 승인 참조의 Amazon Managed Service for Prometheus에서 정의한 리소스를 참조하십시오. 각 리소스의 ARN을 지정할 수 있는 작업을 알아보려면 Amazon Managed Service for Prometheus에서 정의한 작업을 참조하십시오.

Amazon Managed Service for Prometheus ID 기반 정책의 예제를 보려면 Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제 섹션을 참조하십시오.

Amazon Managed Service for Prometheus에 사용되는 정책 조건 키

서비스별 정책 조건 키 지원	아니요

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우 AWS는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키의 여러 값을 지정하는 경우 AWS는 논리적 `OR` 작업을 사용하여 조건을 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그 섹션을 참조하십시오.

AWS는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키 섹션을 참조하십시오.

Amazon Managed Service for Prometheus 조건 키 목록을 보려면 서비스 승인 참조의 Amazon Managed Service for Prometheus에 대한 조건 키를 참조하십시오. 조건 키를 사용할 수 있는 작업과 리소스를 알아보려면 Amazon Managed Service for Prometheus에서 정의한 작업을 참조하십시오.

Amazon Managed Service for Prometheus ID 기반 정책의 예제를 보려면 Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제 섹션을 참조하십시오.

Amazon Managed Service for Prometheus의 액세스 제어 목록(ACL)

ACL 지원	아니요

액세스 제어 목록(ACLs)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는 지를 제어합니다. ACL은 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon Managed Service for Prometheus의 속성 기반 액세스 제어(ABAC)

ABAC 지원(정책의 태그)	예

ABAC(속성 기반 액세스 제어)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. AWS에서는 이러한 속성을 태그라고 합니다. IAM 개체(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. ABAC의 첫 번째 단계로 개체 및 리소스에 태그를 지정합니다. 그런 다음 보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하도록 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로운 상황에 도움이 됩니다.

태그를 기반으로 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우 값은 부분입니다.

ABAC에 대한 자세한 정보는 IAM 사용 설명서의 ABAC란 무엇입니까? 섹션을 참조하십시오. ABAC 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어(ABAC) 사용을 참조하십시오.

Amazon Managed Service for Prometheus에서 임시 자격 증명 사용

임시 보안안 인증 정보 지원	예

일부 AWS 서비스는 임시 보안 인증을 사용하여 로그인할 때 작동하지 않습니다. 임시 자격 증명으로 작동하는 AWS 서비스를 비롯한 추가 정보는 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스 섹션을 참조하십시오.

사용자 이름과 암호를 제외한 다른 방법을 사용하여 AWS Management Console에 로그인하면 임시 보안 인증을 사용하는 것입니다. 예를 들어 회사의 Single Sign-On(SSO) 링크를 사용하여 AWS에 액세스하면 해당 프로세스에서 자동으로 임시 보안 인증을 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 보안 인증을 자동으로 생성합니다. 역할 전환에 대한 자세한 정보는 IAM 사용 설명서의 역할로 전환(콘솔)을 참조하십시오.

AWS CLI 또는 AWS API를 사용하여 임시 보안 인증을 수동으로 만들 수 있습니다 그런 다음 이러한 임시 보안 인증을 사용하여 AWS에 액세스할 수 있습니다. AWS에서는 장기 액세스 키를 사용하는 대신 임시 보안 인증을 동적으로 생성할 것을 권장합니다. 자세한 정보는 IAM의 임시 보안 자격 증명을 참조하십시오.

Amazon Managed Service for Prometheus에 사용되는 전달 액세스 세션

전달 액세스 세션(FAS) 지원	아니요

IAM 사용자 또는 역할을 사용하여 AWS에서 작업을 수행하는 사람은 보안 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 AWS 서비스를 직접 호출하는 보안 주체의 권한과 요청하는 AWS 서비스를 함께 사용하여 다운스트림 서비스에 대한 요청을 수행합니다. FAS 요청은 서비스에서 완료를 위해 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 받은 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 서비스 역할

서비스 역할 지원	아니요

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM role(IAM 역할)입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하십시오.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 Amazon Managed Service for Prometheus 기능이 중단될 수 있습니다. Amazon Managed Service for Prometheus에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할

서비스 연결 역할 지원	예

서비스 연결 역할은 AWS 서비스에 연결된 서비스 역할의 한 타입입니다. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 AWS 계정에 나타나고, 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집할 수는 없습니다.

Amazon Managed Service for Prometheus 생성 또는 관리에 대한 자세한 정보는 Amazon Managed Service for Prometheus에 대한 서비스 연결 역할 섹션을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 ID 기반 정책 예제

기본적으로 사용자 및 역할은 Amazon Managed Service for Prometheus 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface(AWS CLI) 또는 AWS API를 사용해 작업을 수행할 수 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 맡을 수 있습니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하십시오.

각 리소스 유형에 대한 ARN 형식을 비롯하여 Amazon Managed Service for Prometheus에서 정의되는 작업 및 리소스 유형에 대한 자세한 내용은 서비스 승인 참조의 Amazon Managed Service for Prometheus에 사용되는 작업, 리소스 및 조건 키를 참조하십시오.

주제

- 정책 모범 사례
- Amazon Managed Service for Prometheus 콘솔 사용
- 사용자가 자신의 고유한 권한을 볼 수 있도록 허용

정책 모범 사례

ID 기반 정책에 따라 계정에서 사용자가 Amazon Managed Service for Prometheus 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부가 결정됩니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책으로 시작하고 최소 권한을 향해 나아가기: 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 관리형 정책은 AWS 계정에서 사용할 수 있습니다. 사용 사례에 고유한 AWS고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 직무에 대한 AWS 관리형 정책 섹션을 참조하십시오.
- 최소 권한 적용: IAM 정책을 사용하여 권한을 설정하는 경우 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM의 정책 및 권한을 참조하십시오.
- IAM 정책의 조건을 사용하여 액세스 추가 제한: 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. 특정 AWS 서비스(예: AWS CloudFormation)을(를) 통해 사용되는 경우에만 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하십시오.
- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 IAM 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 권장 사항을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하십시오.
- 다중 인증(MFA) 필요: AWS 계정 계정에 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 설정합니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에

MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 MFA 보호 API 액세스 구성 섹션을 참조하십시오.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례 섹션을 참조하십시오.

Amazon Managed Service for Prometheus 콘솔 사용

Amazon Managed Service for Prometheus 콘솔에 액세스하려면 최소한의 권한 집합이 있어야 합니다. 이러한 권한은 AWS 계정에서 Amazon Managed Service for Prometheus 리소스에 대한 세부 정보를 나열하고 볼 수 있도록 허용해야 합니다. 최소 필수 권한보다 더 제한적인 자격 증명 기반 정책을 만들면 콘솔이 해당 정책에 연결된 엔터티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요가 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자와 역할이 Amazon Managed Service for Prometheus 콘솔을 계속해서 사용할 수 있도록 하려면 Amazon Managed Service for Prometheus `ConsoleAccess` 또는 `ReadOnly` AWS 관리형 정책을 엔터티에 추가합니다. 자세한 내용은 IAM 사용 설명서의 사용자에게 권한 추가를 참조하십시오.

사용자가 자신의 고유한 권한을 볼 수 있도록 허용

이 예시는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 또는 AWS CLI나 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

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

Amazon Managed Service for Prometheus에 사용되는 AWS 관리형 정책

AWS 관리형 정책은 AWS에서 생성되고 관리되는 독립 실행형 정책입니다. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있기 때문에 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에서 정의한 권한은 변경할 수 없습니다. AWS에서 AWS 관리형 정책에 정의된 권한을 업데이트할 경우 정책이 연결되어 있는 모든 보안 주체 엔터티(사용자, 그룹 및 역할)에도 업데이트가 적용됩니다. 새로운 AWS 서비스를 시작하거나 새로운 API 작업을 기존 서비스에 이용하는 경우 AWS가 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책을 참조하십시오.

AmazonPrometheusFullAccess

`AmazonPrometheusFullAccess` 정책을 IAM ID에 연결할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- aps - Amazon Managed Service for Prometheus에 대한 전체 액세스 권한 허용
- eks - Amazon Managed Service for Prometheus 서비스가 Amazon EKS 클러스터에 대한 정보를 읽을 수 있도록 허용합니다. 이는 클러스터에서 관리형 스크레이퍼를 생성하고 지표를 검색할 수 있도록 하는 데 필요합니다.
- ec2 - Amazon Managed Service for Prometheus 서비스가 Amazon EC2 네트워크에 대한 정보를 읽을 수 있도록 허용합니다. 이는 Amazon EKS 지표에 액세스할 수 있는 관리형 스크레이퍼를 생성할 수 있도록 하는 데 필요합니다.
- iam – 보안 주체가 관리형 지표 스크레이퍼에 대한 서비스 연결 역할을 생성할 수 있도록 허용합니다.

AmazonPrometheusFullAccess의 내용은 다음과 같습니다.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllPrometheusActions",
   "Effect": "Allow",
   "Action": [
    "aps:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DescribeCluster",
   "Effect": "Allow",
   "Action": [
    "eks:DescribeCluster",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "aps.amazonaws.com"
     ]
    }
```

```
    },
    "Resource": "*"
   },
   {
    "Sid": "CreateServiceLinkedRole",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper*",
    "Condition": {
     "StringEquals": {
      "iam:AWSServiceName": "scraper.aps.amazonaws.com"
     }
    }
   }
  ]
}
```

AmazonPrometheusConsoleFullAccess

`AmazonPrometheusConsoleFullAccess` 정책을 IAM ID에 연결할 수 있습니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `aps` - Amazon Managed Service for Prometheus에 대한 전체 액세스 권한 허용
- `tag` - 보안 주체가 Amazon Managed Service for Prometheus 콘솔에서 태그 제안을 볼 수 있도록 허용

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "TagSuggestions",
   "Effect": "Allow",
   "Action": [
    "tag:GetTagValues",
    "tag:GetTagKeys"
   ],
   "Resource": "*"
```

```
  },
  {
   "Sid": "PrometheusConsoleActions",
   "Effect": "Allow",
   "Action": [
    "aps:CreateWorkspace",
    "aps:DescribeWorkspace",
    "aps:UpdateWorkspaceAlias",
    "aps:DeleteWorkspace",
    "aps:ListWorkspaces",
    "aps:DescribeAlertManagerDefinition",
    "aps:DescribeRuleGroupsNamespace",
    "aps:CreateAlertManagerDefinition",
    "aps:CreateRuleGroupsNamespace",
    "aps:DeleteAlertManagerDefinition",
    "aps:DeleteRuleGroupsNamespace",
    "aps:ListRuleGroupsNamespaces",
    "aps:PutAlertManagerDefinition",
    "aps:PutRuleGroupsNamespace",
    "aps:TagResource",
    "aps:UntagResource",
    "aps:CreateLoggingConfiguration",
    "aps:UpdateLoggingConfiguration",
    "aps:DeleteLoggingConfiguration",
    "aps:DescribeLoggingConfiguration"
   ],
   "Resource": "*"
  }
 ]
}
```

AmazonPrometheusRemoteWriteAccess

AmazonPrometheusRemoteWriteAccess의 내용은 다음과 같습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aps:RemoteWrite"
            ],
            "Effect": "Allow",
```

```
            "Resource": "*"
        }
    ]
}
```

AmazonPrometheusQueryAccess

AmazonPrometheusQueryAccess의 내용은 다음과 같습니다.

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

AWS 관리형 정책: AmazonPrometheusScraperServiceLinkedRolePolicy

AmazonPrometheusScraperServiceLinkedRolePolicy를 IAM 엔터티에 연결할 수 없습니다. 이 정책은 Amazon Managed Service for Prometheus에서 사용자를 대신하여 작업을 수행할 수 있도록 서비스 연결 역할에 연결됩니다. 자세한 내용은 역할을 사용하여 EKS의 지표를 스크래핑합니다. 섹션을 참조하십시오.

이 정책은 기여자에게 Amazon EKS 클러스터에서 읽고 Amazon Managed Service for Prometheus WorkSpace에 쓸 수 있는 권한을 부여합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- aps - 서비스 주체가 Amazon Managed Service for Prometheus WorkSpace에 지표를 작성할 수 있도록 허용합니다.

- ec2 - 서비스 주체가 네트워크 구성을 읽고 수정하여 Amazon EKS 클러스터를 포함하는 네트워크에 연결하도록 허용합니다.
- eks - 서비스 주체가 Amazon EKS 클러스터에 액세스할 수 있도록 허용합니다. 이는 지표를 자동으로 스크래핑할 수 있도록 하기 위해 필요합니다.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DeleteSLR",
   "Effect": "Allow",
   "Action": [
    "iam:DeleteRole"
   ],
   "Resource": "arn:aws:iam::*:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper*"
  },
  {
   "Sid": "NetworkDiscovery",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ENIManagement",
   "Effect": "Allow",
   "Action": "ec2:CreateNetworkInterface",
   "Resource": "*",
   "Condition": {
    "ForAllValues:StringEquals": {
     "aws:TagKeys": [
      "AMPAgentlessScraper"
     ]
    }
   }
  },
  {
   "Sid": "TagManagement",
```

```
    "Effect": "Allow",
    "Action": "ec2:CreateTags",
    "Resource": "arn:*:ec2:*:*:network-interface/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
     },
     "Null": {
      "aws:RequestTag/AMPAgentlessScraper": "false"
     }
    }
   },
   {
    "Sid": "ENIUpdating",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteNetworkInterface",
     "ec2:ModifyNetworkInterfaceAttribute"
    ],
    "Resource": "*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AMPAgentlessScraper": "false"
     }
    }
   },
   {
    "Sid": "EKSAccess",
    "Effect": "Allow",
    "Action": "eks:DescribeCluster",
    "Resource": "arn:*:eks:*:*:cluster/*"
   },
   {
    "Sid": "APSWriting",
    "Effect": "Allow",
    "Action": "aps:RemoteWrite",
    "Resource": "arn:*:aps:*:*:workspace/*",
    "Condition": {
     "StringEquals": {
      "aws:PrincipalAccount": "${aws:ResourceAccount}"
     }
    }
   }
  ]
```

```
}
```

AWS 관리형 정책에 대한 Amazon Managed Service for Prometheus 업데이트

이 서비스가 이러한 변경 내용을 추적하기 시작한 이후부터 Amazon Managed Service for Prometheus의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 Amazon Managed Service for Prometheus 문서 기록 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
AmazonPrometheusFullAccess - 기존 정책에 대한 업데이트	Amazon Managed Service for Prometheus는 Amazon EKS 클러스터에서 지표를 위한 관리형 스크레이퍼 생성을 지원하는 AmazonPrometheusFullAccess에 대한 새로운 권한을 추가했습니다. Amazon EKS 클러스터에 연결하고, Amazon EC2 네트워크를 읽고, 스크레이퍼를 위한 서비스 연결 역할을 생성할 수 있는 권한을 포함합니다.	2023년 11월 26일
AmazonPrometheusScraperServiceLinkedRolePolicy - 새 정책	Amazon Managed Service for Prometheus에서는 Amazon EKS 컨테이너에서 읽을 수 있는 새로운 서비스 연결 역할 정책을 추가하여 지표를 자동으로 스크래핑할 수 있도록 했습니다. Amazon EKS 클러스터에 연결하고, Amazon EC2 네트워크를 읽고, `AMPAgentlessScraper` 로 태그가 지정된 네트워크를 생성 및 삭	2023년 11월 26일

변경 사항	설명	날짜
	제하며, Amazon Managed Service for Prometheus WorkSpace에 쓸 수 있는 권한을 포함합니다.	
AmazonPrometheusConsoleFullAccess - 기존 정책 업데이트	Amazon Managed Service for Prometheus는 CloudWatch Logs에서 알림 관리자 및 규칙 관리자 이벤트 기록을 지원하기 위해 AmazonPrometheusConsoleFullAccess에 새로운 권한을 추가했습니다. `aps:CreateLoggingConfiguration`, `aps:UpdateLoggingConfiguration`, `aps:DeleteLoggingConfiguration`, `aps:DescribeLoggingConfiguration` 권한이 추가되었습니다.	2022년 10월 24일

변경 사항	설명	날짜
AmazonPrometheusConsoleFullAccess - 기존 정책 업데이트	Amazon Managed Service for Prometheus는 새로운 Amazon Managed Service for Prometheus 기능을 지원하기 위해 AmazonPrometheusConsoleFullAccess에 새로운 권한을 추가했으므로 이 정책을 사용하면 Amazon Managed Service for Prometheus 리소스에 태그를 적용할 때 태그 제안 사항 목록을 볼 수 있습니다. `tag:GetTagKeys` , `tag:GetTagValues` , `aps:CreateAlertManagerDefinition` , `aps:CreateRuleGroupsNamespace` , `aps:DeleteAlertManagerDefinition` , `aps:DeleteRuleGroupsNamespace` , `aps:DescribeAlertManagerDefinition` , `aps:DescribeRuleGroupsNamespace` , `aps:ListRuleGroupsNamespaces` , `aps:PutAlertManagerDefinition` , `aps:PutRuleGroupsNamespace` , `aps:TagResource` 및 `aps:UntagResource` 권한이 추가되었습니다.	2021년 9월 29일

변경 사항	설명	날짜
Amazon Managed Service for Prometheus가 변경 사항 추적을 시작함	Amazon Managed Service for Prometheus가 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2021년 9월 15일

Amazon Managed Service for Prometheus ID 및 액세스 문제 해결

다음 정보를 사용하여 Amazon Managed Service for Prometheus 및 IAM으로 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다.

주제

- Amazon Managed Service for Prometheus에서 작업을 수행할 수 있는 권한이 없음
- iam:PassRole을 수행할 권한이 없음
- 내 AWS 계정 외부의 사용자가 내 Amazon Managed Service for Prometheus 리소스에 액세스할 수 있도록 허용하려고 함

Amazon Managed Service for Prometheus에서 작업을 수행할 수 있는 권한이 없음

작업을 수행할 권한이 없다는 오류가 수신되면, 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 `aps:`*`GetWidget`* 권한이 없을 때 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aps:GetWidget on resource: my-example-widget
```

이 경우 `aps:`*`GetWidget`* 작업을 사용하여 *`my-example-widget`* 리소스에 액세스할 수 있도록 `mateojackson` 사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하십시오. 관리자는 로그인 보안 인증을 제공한 사람입니다.

iam:PassRole을 수행할 권한이 없음

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 Amazon Managed Service for Prometheus에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부 AWS 서비스에서는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 해당 서비스에 기존 역할을 전달할 수 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 Amazon Managed Service for Prometheus에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우 Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의합니다. 관리자는 로그인 보안 인증을 제공한 사람입니다.

내 AWS 계정 외부의 사용자가 내 Amazon Managed Service for Prometheus 리소스에 액세스할 수 있도록 허용하려고 함

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하십시오.

- Amazon Managed Service for Prometheus가 이러한 기능을 지원하는지 여부를 알아보려면 Amazon Managed Service for Prometheus가 IAM에서 작동하는 방식 섹션을 참조하십시오.
- 소유하고 있는 AWS 계정의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 자신이 소유한 다른 AWS 계정의 IAM 사용자에 대한 액세스 권한 제공을 참조하십시오.
- 리소스에 대한 액세스 권한을 타사 AWS 계정에게 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사가 소유한 AWS 계정에 대한 액세스 제공을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하십시오.
- 교차 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하십시오.

IAM 권한 및 정책

Amazon Managed Service for Prometheus 작업 및 데이터에 액세스하려면 자격 증명이 필요합니다. 이 자격 증명에는 클라우드 리소스에 대한 Amazon Managed Service for Prometheus 데이터 검색과 같은 작업을 수행하고 AWS 리소스에 액세스할 수 있는 권한이 포함되어야 합니다. 다음 섹션에서는 리소스에 액세스할 수 있는지 대상을 제어하여 리소스를 보호할 수 있도록 AWS Identity and Access Management(IAM) 및 Amazon Managed Service for Prometheus를 사용하는 방법에 대한 세부 정보를 제공합니다. 자세한 내용은 IAM의 정책 및 권한을 참조하세요.

Amazon Managed Service for Prometheus 권한

다음 표에는 Amazon Managed Service for Prometheus에서 수행할 수 있는 작업 및 필요한 권한이 나와 있습니다. 이 작업을 수행하려면 여기에 자세히 설명하지 않은 다른 서비스의 권한이 필요할 수도 있습니다.

작업	필수 권한
알림을 생성합니다.	`aps:CreateAlertManagerAlerts`
워크스페이스에서 알림 관리자 정의를 생성합니다. 자세한 내용은 알림 관리자 섹션을 참조하세요.	`aps:CreateAlertManagerDefinition`
워크스페이스에 규칙 그룹 네임스페이스를 생성합니다. 자세한 내용은 기록 규칙 및 알림 규칙 섹션을 참조하세요.	`aps:CreateRuleGroupsNamespace`
Amazon Managed Service for Prometheus 워크스페이스를 생성합니다. 워크스페이스는 Prometheus 지표 보관 및 쿼리를 위한 전용 논리 공간입니다.	`aps:CreateWorkspace`
워크스페이스에서 알림 관리자 정의를 삭제합니다.	`aps:DeleteAlertManagerDefinition`
무음 알림을 삭제합니다.	`aps:DeleteAlertManagerSilence`
Amazon Managed Service for Prometheus 워크스페이스를 삭제합니다.	`aps:DeleteWorkspace`

작업	필수 권한
알림 관리자 정의에 대한 세부 정보를 검색합니다.	aps:DescribeAlertManagerDefinition
규칙 그룹 네임스페이스에 대한 세부 정보를 검색합니다.	aps:DescribeRuleGroupsNamespace
Amazon Managed Service for Prometheus 워크스페이스에 대한 세부 정보를 검색합니다.	aps:DescribeWorkspace
무음 알림에 대한 세부 정보를 검색합니다.	aps:GetAlertManagerSilence
워크스페이스에서 알림 관리자의 상태를 검색합니다.	aps:GetAlertManagerStatus
레이블을 검색합니다.	aps:GetLabels
Amazon Managed Service for Prometheus 지표에 대한 메타데이터를 검색합니다.	aps:GetMetricMetadata
시계열 데이터를 검색합니다.	aps:GetSeries
알림 관리자 정의에 정의된 알림 그룹 목록을 검색합니다.	aps:ListAlertManagerAlertGroups
알림 관리자에 정의된 알림 목록을 검색합니다.	aps:ListAlertManagerAlerts
알림 관리자 정의에 정의된 수신기 목록을 검색합니다.	aps:ListAlertManagerReceivers
정의된 무음 알림 목록을 검색합니다.	aps:ListAlertManagerSilences
활성 알림 목록을 검색합니다.	aps:ListAlerts
워크스페이스의 규칙 그룹 네임스페이스에서 규칙 목록을 검색합니다.	aps:ListRules

작업	필수 권한
워크스페이스에 있는 규칙 그룹 네임스페이스 목록을 검색합니다.	`aps:ListRuleGroupsNamespaces`
Amazon Managed Service for Prometheus 리소스와 연결된 태그를 검색합니다.	`aps:ListTagsForResource`
계정에 있는 Amazon Managed Service for Prometheus 워크스페이스 목록을 검색합니다.	`aps:ListWorkspaces`
워크스페이스의 기존 알림 관리자 정의를 업데이트합니다.	`aps:PutAlertManagerDefinition`
무음 알림을 생성합니다.	`aps:PutAlertManagerSilences`
기존 규칙 그룹 네임스페이스를 업데이트합니다.	`aps:PutRuleGroupsNamespace`
Amazon Managed Service for Prometheus 지표에서 쿼리를 실행합니다.	`aps:QueryMetrics`
원격 쓰기 작업을 수행하여 Prometheus 서버에서 Amazon Managed Service for Prometheus로의 지표 스트리밍을 시작합니다.	`aps:RemoteWrite`
Amazon Managed Service for Prometheus 리소스에 태그를 할당합니다.	`aps:TagResource`
Amazon Managed Service for Prometheus 리소스에서 태그를 제거합니다.	`aps:UntagResource`
기존 워크스페이스의 별칭을 수정합니다.	`aps:UpdateWorkspaceAlias`
로깅 구성을 생성합니다.	`aps:CreateLoggingConfiguration`
쿼리 로깅 구성을 삭제합니다.	`aps:DeleteLoggingConfiguration`

작업	필수 권한
워크스페이스 로깅 구성을 설명합니다.	`aps:DescribeLoggingConfiguration`
로깅 구성을 업데이트합니다.	`aps:UpdateLoggingConfiguration`

샘플 IAM 정책

이 섹션에서는 생성할 수 있는 다른 자체 관리형 정책의 예를 제공합니다.

다음 IAM 정책은 Amazon Managed Service for Prometheus에 대한 전체 액세스 권한을 부여하고 사용자가 Amazon EKS 클러스터를 검색하고 이에 대한 세부 정보를 볼 수 있도록 합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:*",
                "eks:DescribeCluster",
                "eks:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Managed Service for Prometheus에 대한 규정 준수 확인

AWS 서비스이(가) 특정 규정 준수 프로그램의 범위에 포함되는지 알아보려면 규정 준수 프로그램 제공 범위 내 AWS 서비스을(를) 참조하고 관심 있는 규정 준수 프로그램을 선택하십시오. 일반적인 정보는 AWS규정 준수 프로그램을 참조하세요.

AWS Artifact을(를) 사용하여 제3자 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 AWS Artifact에서 보고서 다운로드 섹션을 참조하세요.

AWS 서비스 사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 결정됩니다.AWS에서는 규정 준수를 지원할 다음과 같은 리소스를 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 – 이 배포 안내서에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 기본 AWS환경을 배포하기 위한 단계를 제공합니다.
- Architecting for HIPAA Security and Compliance on Amazon Web Services(Amazon Web Services에서 HIPAA 보안 및 규정 준수 기술 백서 설계) - 이 백서는 기업에서 AWS를 사용하여 HIPAA를 준수하는 애플리케이션을 만드는 방법을 설명합니다.

> **Note**
>
> 모든 AWS 서비스에 HIPAA 자격이 있는 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조 섹션을 참조하십시오.

- AWS 규정 준수 리소스 - 고객 조직이 속한 산업 및 위치에 적용될 수 있는 워크북 및 가이드 컬렉션입니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드에서는 AWS 서비스를 보호하기 위한 모범 사례를 요약하고 여러 프레임워크(미국 표준 기술 연구소(NIST), 결제 카드 산업 보안 표준 위원회(PCI), 국제 표준화기구(ISO) 등)에서 보안 제어에 대한 지침을 매핑합니다.
- AWS Config 개발자 가이드의 규칙을 사용하여 리소스 평가 – AWS Config 서비스는 내부 사례, 산업 지침 및 규제에 대한 리소스 구성의 준수 상태를 평가합니다.
- AWS Security Hub – 이 AWS 서비스는 AWS내의 보안 상태에 대한 포괄적인 보기를 제공합니다. Security Hub는 보안 제어를 사용하여 AWS리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조 섹션을 참조하십시오.
- AWS Audit Manager - 이 AWS 서비스는 AWS 사용을 지속적으로 감사하여 리스크를 관리하고 규정 및 업계 표준을 준수하는 방법을 간소화할 수 있도록 지원합니다.

Amazon Managed Service for Prometheus의 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 다수의 가용 영역을 제공하며 이러한 가용 영역은 짧은 지연 시간, 높은 처리량 및 높은 중복성을 갖춘 네트워크에 연결되어 있습니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라 섹션을 참조하세요.

AWS 글로벌 인프라 외에도 Amazon Managed Service for Prometheus는 고가용성 데이터 지원을 비롯하여 데이터 복원력 및 백업 요구를 지원하는 데 도움이 되는 몇 가지 기능을 제공합니다.

Amazon Managed Service for Prometheus의 인프라 보안

관리형 서비스인 Amazon Managed Service for Prometheus는 AWS 글로벌 네트워크 보안으로 보호됩니다. AWS 보안 서비스와 AWS의 인프라 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하세요. 인프라 보안에 대한 모범 사례를 사용하여 AWS 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS에서 게시한 API 호출을 사용하여 네트워크를 통해 Amazon Managed Service for Prometheus에 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.
- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)을 사용하여 임시 보안 자격 증명을 생성하여 요청에 서명할 수 있습니다.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할

Amazon Managed Service for Prometheus는 AWS Identity and Access Management(IAM) 서비스 연결 역할을 사용합니다. 서비스 연결 역할은 Amazon Managed Service for Prometheus에 직접 연결된 고유한 유형의 IAM 역할입니다. 서비스 연결 역할은 Amazon Managed Service for Prometheus에서 사전 정의하며, 서비스에서 다른 AWS 서비스를 대신 호출하기 위해 필요한 모든 권한을 포함합니다.

필요한 권한을 수동으로 추가할 필요가 없으므로 서비스 연결 역할로 Amazon Managed Service for Prometheus를 더 쉽게 설정할 수 있습니다. Amazon Managed Service for Prometheus에서 서비스 연결 역할의 권한을 정의하므로 다르게 정의되지 않은 한, Amazon Managed Service for Prometheus만 해당 역할을 맡을 수 있습니다. 정의된 권한에는 신뢰 정책과 권한 정책이 포함되며 이 권한 정책은 다른 IAM 엔터티에 연결할 수 없습니다.

역할을 사용하여 EKS의 지표를 스크래핑합니다.

Amazon Managed Service for Prometheus를 사용하여 자동으로 지표를 스크래핑하는 경우 AWSServiceRoleForAmazonPrometheusScraper 서비스 연결 역할은 필요한 권한을 수동으로 추가할 필요가 없기 때문에 관리형 수집기의 설정이 쉬워집니다. Amazon Managed Service for Prometheus에서 권한을 정의하며 Amazon Managed Service for Prometheus만 역할을 맡을 수 있습니다.

서비스 연결 역할을 지원하는 기타 서비스에 대한 자세한 내용은 IAM으로 작업하는 AWS 서비스를 참조하고 서비스 연결 역할 열에 예가 있는 서비스를 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할 권한

Amazon Managed Service for Prometheus는 AWSServiceRoleForAmazonPrometheusScraper라는 접두사가 붙은 서비스 연결 역할을 사용하여 Amazon Managed Service for Prometheus가 Amazon EKS 클러스터의 지표를 자동으로 스크래핑할 수 있도록 합니다.

AWSServiceRoleForAmazonPrometheusScraper 서비스 연결 역할은 다음 서비스를 신뢰하여 역할을 맡습니다.

- `scraper.aps.amazonaws.com`

AmazonPrometheusScraperServiceLinkedRolePolicy라는 역할 권한 정책을 사용하면 Amazon Managed Service for Prometheus가 지정된 리소스에 대해 다음 작업을 완료할 수 있습니다.

- Amazon EKS 클러스터를 포함하는 네트워크에 연결할 수 있도록 네트워크 구성을 준비하고 수정하십시오.
- Amazon EKS 클러스터에서 지표를 읽고 Amazon Managed Service for Prometheus WorkSpace에 지표를 작성합니다.

사용자, 그룹 또는 역할이 서비스 연결 역할을 생성할 수 있도록 사용 권한을 구성해야 합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console, AWS CLI, 또는 AWS API에서 Amazon EKS 또는 Amazon Managed Service for Prometheus를 사용하여 관리형 수집기 인스턴스를 생성하는 경우 Amazon Managed Service for Prometheus에서 서비스 연결 역할을 생성합니다.

> **Important**
>
> 이러한 서비스 연결 역할은 해당 역할이 지원하는 기능을 사용하는 다른 서비스에서 작업을 완료했을 경우 계정에 나타날 수 있습니다. 자세한 내용은 내 AWS 계정에 표시되는 새 역할을 참조하십시오.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. Amazon EKS 또는 Amazon Managed Service for Prometheus를 사용하여 관리형 수집기 인스턴스를 생성하는 경우 Amazon Managed Service for Prometheus에서 서비스 연결 역할을 다시 생성합니다.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할 편집

Amazon Managed Service for Prometheus는 AWSServiceRoleForAmazonPrometheusScraper 서비스 연결 역할을 편집하도록 허용하지 않습니다. 서비스 연결 역할을 생성한 후에는 다양한 객체가 역할을 참조할 수 있기 때문에 역할 이름을 변경할 수 없습니다. 하지만 IAM을 사용하여 역할의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 편집을 참조하십시오.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할 삭제

AWSServiceRoleForAmazonPrometheusScraper 역할을 수동으로 삭제하지 않아도 됩니다. AWS Management Console, AWS CLI, 또는 AWS API의 역할과 관련된 관리형 수집기 인스턴스를 모두 삭제하면 Amazon Managed Service for Prometheus가 리소스를 정리하고 서비스 연결 역할을 자동으로 삭제합니다.

Amazon Managed Service for Prometheus에 대한 서비스 연결 역할에 대해 지원되는 리전

Amazon Managed Service for Prometheus는 서비스가 제공되는 모든 리전에서 서비스 연결 역할 사용을 지원합니다. 자세한 내용은 지원되는 리전 섹션을 참조하십시오.

AWS CloudTrail을 사용하여 Amazon Managed Service for Prometheus API 호출 로깅

Prometheus용 Amazon Managed Service는 Prometheus용 Amazon Managed Service에서 사용자, 역할 또는 AWS 서비스가 수행한 작업에 대한 기록을 제공하는 서비스와 통합되어 AWS CloudTrail 있습니다. CloudTrail Prometheus용 Amazon 관리 서비스에 대한 모든 API 호출을 이벤트로 캡처합니다.

캡처되는 호출에는 Amazon Managed Service for Prometheus에서 수행한 호출과 Amazon Managed Service for Prometheus API 작업에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 Amazon S3 버킷으로 CloudTrail 이벤트를 지속적으로 전송할 수 있습니다. 여기에는 Prometheus용 Amazon 매니지드 서비스에 대한 이벤트가 포함됩니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다. 에서 수집한 정보를 사용하여 Prometheus용 Amazon Managed Service에 이루어진 요청 CloudTrail, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

자세한 내용은 사용 CloudTrail 설명서를 참조하십시오. AWS CloudTrail

Prometheus용 Amazon 매니지드 서비스 정보 CloudTrail

CloudTrail 계정을 만들면 AWS 계정에서 활성화됩니다. Amazon Managed Service for Prometheus에서 활동이 발생하면 해당 활동이 이벤트 기록의 CloudTrail AWS 다른 서비스 이벤트와 함께 이벤트에 기록됩니다. 계정에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다. AWS 자세한 내용은 이벤트 기록으로 CloudTrail 이벤트 보기를 참조하십시오.

Amazon Managed Service for Prometheus의 이벤트를 포함하여 AWS 계정의 진행 중인 이벤트 기록을 보려면 트레일을 생성하십시오. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 기본적으로 콘솔에서 트레일을 생성하면 트레일이 모든 AWS 지역에 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- CloudTrail에 대한 Amazon SNS 알림 구성
- 여러 지역에서 CloudTrail 로그 파일 받기 및 여러 계정에서 CloudTrail 로그 파일 받기

Amazon Managed Service for Prometheus는 다음 작업의 로깅을 지원합니다.

- CreateAlertManagerAlerts
- CreateAlertManagerDefinition
- CreateRuleGroupsNamespace
- CreateWorkspace
- DeleteAlertManagerDefinition
- DeleteAlertManagerSilence

- DeleteWorkspace
- DeleteRuleGroupsNamespace
- DescribeAlertManagerDefinition
- DescribeRulesGroupsNamespace
- DescribeWorkspace
- ListRuleGroupsNamespaces
- ListWorkspaces
- PutAlertManagerDefinition
- PutAlertManagerSilences
- PutRuleGroupsNamespace
- UpdateWorkspaceAlias

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하십시오.

Amazon Managed Service for Prometheus 로그 파일 항목의 이해

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

예: CreateWorkspace

다음 예제는 CreateWorkspace 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
    "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"
}
```

예: CreateAlertManagerDefinition

다음 예제는 CreateAlertManagerDefinition 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
    "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"
}
```


예: CreateRuleGroupsNamespace

다음 예제는 CreateRuleGroupsNamespace 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
    "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"
}
```

서비스 계정에 대한 IAM 역할 설정

서비스 계정에 대한 IAM 역할을 사용할 경우 IAM 역할을 Kubernetes 서비스 계정에 연결할 수 있습니다. 이렇게 하면 이 서비스 계정에서는 이 서비스 계정을 사용하는 모든 포드에 있는 컨테이너에 AWS 권한을 제공할 수 있습니다. 자세한 내용은 서비스 계정에 대한 IAM 역할을 참조하십시오.

서비스 계정의 IAM 역할을 서비스 역할이라고도 합니다.

Amazon Managed Service for Prometheus에서 서비스 역할을 사용하면 Amazon Managed Service for Prometheus, Prometheus 서버 및 Grafana 서버 간에 권한을 부여하고 인증하는 데 필요한 역할을 얻을 수 있습니다.

사전 조건

이 페이지의 절차를 수행하려면 AWS CLI 및 EKSCTL 명령줄 인터페이스가 설치되어 있어야 합니다.

Amazon EKS 클러스터의 지표 수집을 위한 서비스 역할 설정

Amazon Managed Service for Prometheus가 Amazon EKS 클러스터의 Prometheus 서버에서 지표를 수집할 수 있도록 서비스 역할을 설정하려면 다음 권한을 가진 계정으로 로그온해야 합니다.

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

Amazon Managed Service for Prometheus에 수집하기 위한 서비스 역할을 설정하려면

1. 다음 콘텐츠가 포함된 `createIRSA-AMPIngest.sh`이라는 파일을 생성합니다. `<my_amazon_eks_clustername>`을 클러스터 이름으로 바꾸고 `<my_prometheus_namespace>`를 Prometheus 네임스페이스로 바꿉니다.

```
#!/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. 다음 명령을 입력하여 스크립트에 필요한 권한을 부여합니다.

```
chmod +x createIRSA-AMPIngest.sh
```

3. 스크립트를 실행합니다.

지표 쿼리를 위해 서비스 계정에 대한 IAM 역할 설정

Amazon Managed Service for Prometheus에서 지표를 쿼리할 수 있도록 서비스 계정(서비스 역할)에 대한 IAM 역할을 설정하려면 다음 권한을 가진 계정으로 로그온해야 합니다.

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

Amazon Managed Service for Prometheus 지표의 쿼리를 위한 서비스 역할을 설정하려면

1. 다음 콘텐츠가 포함된 `createIRSA-AMPQuery.sh`이라는 파일을 생성합니다. <my_amazon_eks_clustername>을 클러스터 이름으로 바꾸고 <my_prometheus_namespace>를 Prometheus 네임스페이스로 바꿉니다.

```
#!/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. 다음 명령을 입력하여 스크립트에 필요한 권한을 부여합니다.

```
chmod +x createIRSA-AMPQuery.sh
```

3. 스크립트를 실행합니다.

인터페이스 VPC 엔드포인트에서 Amazon Managed Service for Prometheus 사용

Amazon Virtual Private Cloud(Amazon VPC)를 사용하여 AWS 리소스를 호스트하는 경우, VPC와 Amazon Managed Service for Prometheus 간에 프라이빗 연결을 설정할 수 있습니다. 이러한 연결을 사용하면 Amazon Managed Service for Prometheus가 퍼블릭 인터넷을 통하지 않고 VPC의 리소스와 통신할 수 있습니다.

Amazon VPC란 사용자가 정의한 가상 네트워크에서 AWS 리소스를 시작할 때 사용할 수 있는 AWS 서비스입니다. VPC가 있으면 IP 주소 범위, 서브넷, 라우팅 테이블, 네트워크 게이트웨이 등 네트워크 설정을 제어할 수 있습니다. VPC를 Amazon Managed Service for Prometheus에 연결하려면 인터페이스 VPC 엔드포인트를 정의하여 VPC를 AWS 서비스에 연결합니다. 이 엔드포인트를 사용하면 인터넷 게이트웨이나 NAT(네트워크 주소 변환) 인스턴스 또는 VPN 연결 없이도 Amazon Managed Service for Prometheus에 안정적이고 확장 가능하게 연결됩니다. 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC란 무엇입니까를 참조하세요.

인터페이스 VPC 엔드포인트는 프라이빗 IP 주소와 함께 탄력적 네트워크 인터페이스를 사용하여 AWS 서비스 간 프라이빗 통신을 사용할 수 있는 AWS 기술인 AWS PrivateLink에 의해 구동됩니다. 자세한 내용은 새 기능 – AWS 서비스를 위한 AWS PrivateLink 블로그 게시물을 참조하세요.

다음은 Amazon VPC 사용자를 위한 정보입니다. Amazon VPC를 시작하는 방법에 대한 내용은 Amazon VPC 사용 설명서에서 시작하기를 참조하세요.

Amazon Managed Service for Prometheus에 대한 인터페이스 VPC 엔드포인트 생성

인터페이스 VPC 엔드포인트를 생성하여 Amazon Managed Service for Prometheus 사용을 시작합니다. 다음 서비스 이름 엔드포인트 중에서 선택합니다.

- `com.amazonaws.region.aps-workspaces`

  Prometheus 호환 API를 사용하려면 이 서비스 이름을 선택합니다. 자세한 내용은 Amazon Managed Service for Prometheus 사용 설명서에서 Prometheus 호환 API를 참조하세요.

- `com.amazonaws.region.aps`

  워크스페이스 관리 태스크를 수행하려면 이 서비스 이름을 선택합니다. 자세한 내용은 Amazon Managed Service for Prometheus 사용 설명서에서 Amazon Managed Service for Prometheus API를 참조합니다.

> **Note**
>
> 인터넷에 직접 액세스할 수 없는 VPC에서 remote_write를 사용하는 경우 AWS Security Token Service에 대한 인터페이스 VPC 엔드포인트도 생성하여 sigv4가 엔드포인트를 통해 작동할 수 있도록 해야 합니다. AWS STS에 대한 VPC 엔드포인트 생성에 대한 내용은 AWS Identity and Access Management 사용 설명서의 AWS STS 인터페이스 VPC 엔드포인트 사용을 참조하세요. 리전화된 엔드포인트를 사용하도록 AWS STS를 설정해야 합니다.

인터페이스 VPC 엔드포인트를 생성하는 단계별 지침을 비롯한 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하세요.

> **Note**
>
> VPC 엔드포인트 정책을 사용하여 Amazon Managed Service for Prometheus 인터페이스 VPC 엔드포인트에 대한 액세스를 제어할 수 있습니다. 자세한 내용은 다음 섹션을 참조하세요.

Amazon Managed Service for Prometheus에 대한 인터페이스 VPC 엔드포인트를 생성했고 VPC에 있는 워크스페이스로 흐르는 데이터가 이미 있는 경우 지표는 기본적으로 인터페이스 VPC 엔드포인트를 통해 흐릅니다. Amazon Managed Service for Prometheus는 퍼블릭 엔드포인트 또는 프라이빗 인터페이스 엔드포인트(사용 중인 것 중 하나)를 사용하여 이 태스크를 수행합니다.

Amazon Managed Service for Prometheus VPC 엔드포인트에 대한 액세스 제어

VPC 엔드포인트 정책을 사용하여 Amazon Managed Service for Prometheus 인터페이스 VPC 엔드포인트에 대한 액세스를 제어할 수 있습니다. VPC 엔드포인트 정책은 엔드포인트를 만들거나 수정 시 엔드포인트에 연결하는 IAM 리소스 정책입니다. 엔드포인트를 생성할 때 정책을 연결하지 않으면 Amazon VPC는 서비스에 대한 전체 액세스를 허용하는 기본 정책을 자동으로 연결합니다. 엔드포인트 정책은 IAM ID 기반 정책 또는 서비스별 정책을 재정의하거나 대체하지 않습니다. 이는 엔드포인트에서 지정된 서비스로의 액세스를 제어하기 위한 별도의 정책입니다.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 통해 서비스에 대한 액세스 제어를 참조하십시오.

다음은 Amazon Managed Service for Prometheus에 대한 엔드포인트 정책의 예입니다. 이 정책은 VPC를 통해 Amazon Managed Service for Prometheus에 연결하는 `PromUser` 역할이 있는 사용자가 워크스페이스 및 규칙 그룹을 볼 수 있도록 허용하지만, 워크스페이스를 생성하거나 삭제하는 등은 허용하지 않습니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonManagedPrometheusPermissions",
            "Effect": "Allow",
            "Action": [
                "aps:DescribeWorkspace",
                "aps:DescribeRuleGroupsNamespace",
                "aps:ListRuleGroupsNamespace",
                "aps:ListWorkspaces"
```

```
            ],
            "Resource": "arn:aws:aps:*:*:/workspaces*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/PromUser"
                ]
            }
        }
    ]
}
```

다음 예제는 지정된 VPC의 지정된 IP 주소에서 들어오는 요청만 성공하도록 허용하는 정책을 보여 줍니다. 다른 IP 주소에서 들어오는 요청은 실패합니다.

```
{
    "Statement": [
        {
            "Action": "aps:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
}
```

문제 해결

다음 섹션을 사용하여 Amazon Managed Service for Prometheus와 관련된 문제를 해결할 수 있습니다.

주제

429 오류

다음 예와 비슷한 429 오류가 표시되면 요청이 Amazon Managed Service for Prometheus 수집 할당량을 초과한 것입니다.

```
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
```

다음 예와 비슷한 429 오류가 표시되면 요청이 WorkSpace의 활성 지표 수에 대한 Amazon Managed Service for Prometheus 할당량을 초과한 것입니다.

```
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
```

Amazon Managed Service for Prometheus 서비스 할당량 및 증가 요청 방법에 대한 자세한 내용은 Amazon Managed Service for Prometheus 서비스 할당량 섹션을 참조하십시오.

중복된 샘플이 보임

고가용성 Prometheus 그룹을 사용하는 경우 Prometheus 인스턴스에서 외부 레이블을 사용하여 중복 제거를 설정해야 합니다. 자세한 설명은 Amazon Managed Service for Prometheus로 전송된 고가용성 지표 중복 제거 섹션을 참조하세요.

중복 데이터와 관련된 기타 문제는 다음 섹션에서 설명합니다.

샘플 타임스탬프에 대한 오류가 표시됩니다.

Prometheus용 Amazon Managed Service는 데이터를 순서대로 수집하며, 각 샘플에 이전 샘플보다 늦은 타임스탬프가 있을 것으로 예상합니다.

데이터가 순서대로 도착하지 않는 경우, 또는 에 대한 오류가 발생할 수 있습니다. `out-of-order samples` `duplicate sample for timestamp` `samples with different value but same timestamp` 이러한 문제는 일반적으로 Prometheus용 Amazon Managed Service로 데이터를 보내는 클라이언트의 잘못된 설정으로 인해 발생합니다. 에이전트 모드에서 실행되는 Prometheus 클라이언트를 사용하는 경우 구성에서 시리즈 이름이 중복되거나 대상이 중복된 규칙이 있는지 확인하십시오. 메트릭이 타임스탬프를 직접 제공하는 경우 타임스탬프가 잘못된 것이 아닌지 확인하십시오.

작동 방식이나 설정을 확인하는 방법에 대한 자세한 내용은 Prom Labs의 Prometheus의 중복 샘플 및 O ut-of-order 타임스탬프 오류에 대한 이해 블로그 게시물을 참조하십시오.

제한과 관련된 오류 메시지가 표시됨

> **Note**
>
> Prometheus용 Amazon 관리형 서비스는 Prometheus 리소스 CloudWatch 사용을 모니터링하기 위한 사용량 지표를 제공합니다. CloudWatch사용량 지표 알람 기능을 사용하면 Prometheus 리소스 및 사용량을 모니터링하여 한도 오류를 방지할 수 있습니다.

다음 오류 메시지 중 하나가 표시되면 Amazon Managed Service for Prometheus 할당량 중 하나의 증가를 요청하여 문제를 해결할 수 있습니다. 자세한 설명은 Amazon Managed Service for Prometheus 서비스 할당량 섹션을 참조하세요.

- 사용자당 시리즈 제한인 *<value>*개를 초과했습니다. 관리자에게 문의하여 상향 조정하십시오.
- 지표당 시리즈 제한인 *<value>*개를 초과했습니다. 관리자에게 문의하여 상향 조정하십시오.
- 수집 속도 제한(...)을 초과했습니다.
- 시리즈에 너무 많은 레이블(...) 시리즈가 있습니다. '%s'
- 쿼리 시간 범위가 제한(쿼리 길이: xxx, 제한: yyy)을 초과했습니다.
- 수집기에서 청크를 가져오는 동안 쿼리가 최대 청크 수 제한에 도달했습니다.
- 제한을 초과했습니다. 계정당 최대 WorkSpace 수입니다.

로컬 Prometheus 서버 출력이 제한을 초과했습니다.

Amazon Managed Service for Prometheus에는 WorkSpace가 Prometheus 서버에서 수신할 수 있는 데이터 양에 대한 서비스 할당량이 있습니다. Prometheus 서버가 Amazon Managed Service for Prometheus로 보내는 데이터의 양을 확인하려면 Prometheus 서버에서 다음 쿼리를 실행하면 됩니다. Prometheus 출력이 Amazon Managed Service for Prometheus 제한을 초과하는 경우 해당 서비스 할당량의 증가를 요청할 수 있습니다. 자세한 내용은 Amazon Managed Service for Prometheus 서비스 할당량 섹션을 참조하십시오.

로컬 자체 실행 Prometheus 서버를 대상으로 쿼리하여 출력 제한을 확인합니다.

데이터 유형	사용할 쿼리
현재 활성 시리즈	`prometheus_tsdb_head_series`
현재 수집 속도	`rate(prometheus_tsdb_head_samples_appended_total[5m])`
메트릭 ost-to-least 이름별 활성 시리즈 목록	`sort_desc(count by(__name__)`

데이터 유형	사용할 쿼리	
	`({__name__!=""})`	
지표 시리즈별 레이블 수	`group by(mylabelname) ({__name__!=""})`	

일부 데이터가 표시되지 않아요.

Prometheus용 Amazon 관리 서비스로 전송된 데이터는 여러 가지 이유로 삭제될 수 있습니다. 다음 표는 데이터가 수집되지 않고 폐기될 수 있는 이유를 보여줍니다.

Amazon을 사용하여 데이터가 삭제되는 양과 이유를 추적할 수 있습니다. CloudWatch 자세한 설명은 CloudWatch 메트릭 섹션을 참조하세요.

이유	의미
greater_than_max_sample_age	현재 시간보다 오래된 로그 라인 삭제
new-value-for-timestamp	중복 샘플은 이전에 기록된 것과 다른 타임스탬프와 함께 전송됩니다.
per_metric_series_limit	지표별 활성 시리즈 제한에 도달했습니다.
per_user_series_limit	총 활성 시리즈 수 제한에 도달했습니다.
rate_limited	수집 속도가 제한되었습니다.
sample-out-of-order	샘플이 잘못된 순서로 전송되어 처리할 수 없습니다.
label_value_too_long	레이블 값이 허용된 문자 제한보다 깁니다.
max_label_names_per_series	지표별 레이블 이름에 도달했습니다.
missing_metric_name	지표 이름은 제공되지 않습니다.

이유	의미
metric_name_invalid	잘못된 지표 이름이 제공되었습니다.
label_invalid	잘못된 레이블이 제공되었습니다.
duplicate_label_names	중복된 레이블 이름이 제공되었습니다.

태그 지정

태그는 사용자 또는 AWS에서 AWS 리소스에 할당하는 사용자 지정 속성 레이블입니다. 각 AWS 태그는 두 부분으로 구성됩니다.

- 태그 키(예: `CostCenter`, `Environment`, `Project` 또는 `Secret`). 태그 키는 대/소문자를 구별합니다.
- 태그 값(예: `111122223333`, `Production` 또는 팀 이름)으로 알려진 선택적 필드. 태그 값을 생략하는 것은 빈 문자열을 사용하는 것과 같습니다. 태그 키처럼 태그 값은 대/소문자를 구별합니다.

태그 키와 태그 값을 합해서 키 값 페어라고 합니다. 각 워크스페이스에 최대 50개의 태그를 할당할 수 있습니다.

태그를 사용하면 AWS 리소스를 식별하고 구성하는 데 도움이 됩니다. 많은 AWS 서비스가 태그 지정을 지원하므로 다른 서비스의 리소스에 동일한 태그를 할당하여 해당 리소스의 관련 여부를 나타낼 수 있습니다. 예를 들어 Amazon S3 버킷에 할당한 것과 동일한 태그를 Amazon Managed Service for Prometheus 워크스페이스에 할당할 수 있습니다. 태깅 전략에 대한 자세한 내용은 AWS 리소스 태그 지정을 참조하세요.

Amazon Managed Service for Prometheus에서는 워크스페이스와 규칙 그룹 네임스페이스 모두에 태그를 지정할 수 있습니다. 콘솔, AWS CLI, API 또는 SDK를 사용하여 이러한 리소스의 태그를 추가, 관리 및 제거할 수 있습니다. 태그로 워크스페이스 및 규칙 그룹 네임스페이스를 식별, 구성 및 추적하는 것 외에도 IAM 정책의 태그를 사용하여 Amazon Managed Service for Prometheus 리소스를 보고 상호 작용할 수 있는 사용자를 제어할 수 있습니다.

태그 제한

태그에 적용되는 기본 제한은 다음과 같습니다.

- 각 리소스는 최대 50개의 태그를 보유할 수 있습니다.
- 각 리소스에 대해 각 태그 키는 고유하며 하나의 값만 가질 수 있습니다.
- 태그 키의 최대 길이는 UTF-8 형식의 유니코드 문자 128자입니다.
- 태그 값의 최대 길이는 UTF-8 형식의 유니코드 문자 256자입니다.
- 태그 지정 스키마를 여러 AWS 서비스와 리소스에서 사용하는 경우 다른 서비스에서 허용되는 문자에 제한이 있을 수 있음에 유의하십시오. 일반적으로 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 숫자, 공백 및 . : + = @ _ / -(하이픈) 문자도 있습니다.

- 태그 키와 값은 대/소문자를 구분합니다. 모범 사례는 태그를 대문자로 사용할 것을 전략으로 결정하고 모든 리소스 유형에 대해 일관되게 해당 전략을 구현하는 것입니다. 예를 들어, `Costcenter`, `costcenter` 또는 `CostCenter`를 사용할지 결정하고 모든 태그에 대해 동일한 규칙을 사용합니다. 대/소문자가 일치하지 않는 유사한 태그를 사용하지 마세요.
- 키 또는 값에 `aws:`, `AWS:` 또는 이러한 접두사의 대문자 또는 소문자 조합을 사용하지 않습니다. 이러한 이름은 AWS 전용으로 예약되어 있습니다. 이 접두사가 지정된 태그 키나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 포함된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.

주제

워크스페이스 태그 지정

Amazon Managed Service for Prometheus 워크스페이스에 태그를 지정하려면 이 섹션의 절차를 수행하세요.

주제

워크스페이스에 태그 추가

Amazon Managed Service for Prometheus 워크스페이스에 태그를 추가하면 AWS 리소스를 식별 및 구성하고 해당 리소스에 대한 액세스를 관리할 수 있습니다. 먼저 워크스페이스에 하나 이상의 태그(키-값 페어)를 추가합니다. 태그가 생성된 후 해당 태그를 기준으로 워크스페이스에 대한 액세스를 관리하는 IAM 정책을 생성할 수 있습니다. 콘솔 또는 AWS CLI를 사용하여 Amazon Managed Service for Prometheus 워크스페이스에 태그를 추가할 수 있습니다.

> **Important**
>
> 워크스페이스에 태그를 추가하면 해당 워크스페이스에 대한 액세스에 영향을 미칠 수 있습니다. 워크스페이스에 태그를 추가하기 전에 태그를 사용하여 리소스에 대한 액세스를 제어할 수도 있는 모든 IAM 정책을 검토하세요.

워크스페이스를 생성할 때 Amazon Managed Service for Prometheus 워크스페이스에 태그를 추가하는 방법에 대한 자세한 내용은 WorkSpace 생성 섹션을 참조하세요.

주제

- 워크스페이스에 태그 추가(콘솔)
- 워크스페이스에 태그 추가(AWS CLI)

워크스페이스에 태그 추가(콘솔)

콘솔을 사용하여 Amazon Managed Service for Prometheus 워크스페이스에 1개 이상의 태그를 추가할 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. [Tags] 탭을 선택합니다.
6. Amazon Managed Service for Prometheus 워크스페이스에 태그가 추가되지 않은 경우 태그 생성을 선택합니다. 그렇지 않으면 태그 관리를 선택합니다.
7. 키에 태그 이름을 입력합니다. 값(Value)에 태그의 선택적 값을 추가할 수 있습니다.
8. (선택 사항) 다른 태그를 추가하려면 다시 태그 추가를 선택합니다.
9. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.

워크스페이스에 태그 추가(AWS CLI)

다음 단계에 따라 AWS CLI를 사용하여 Amazon Managed Service for Prometheus 워크스페이스에 태그를 추가하세요. 워크스페이스를 생성할 때 워크스페이스에 태그를 추가하려면 WorkSpace 생성 섹션을 참조하세요.

이 단계에서는 사용자가 이미 최신 버전의 AWS CLI를 설치했거나 현재 버전으로 업데이트했다고 가정합니다. 자세한 정보는 AWS Command Line Interface 설치 섹션을 참조하세요.

터미널이나 명령줄에서 tag-resource 명령을 실행하여, 태그를 추가할 워크스페이스의 Amazon 리소스 이름(ARN)과 추가할 태그의 키와 값을 지정합니다. 하나의 워크스페이스에 2개 이상의 태그를 추가할 수 있습니다. 예를 들어 My-Workspace라는 Amazon Managed Service for Prometheus 워크스페이스에 태그 키가 *Status*이고 태그 값이 *Secret*인 태그와 태그 키가 *Team*이고 태그 값이 *My-Team*인 2개의 태그를 지정하려면 다음과 같이 하세요.

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
--tags Status=Secret,Team=My-Team
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다.

워크스페이스의 태그 보기

태그를 사용하면 AWS 리소스를 식별 및 구성하고 해당 리소스에 대한 액세스를 관리할 수 있습니다. 태깅 전략에 대한 자세한 내용은 AWS 리소스 태그 지정을 참조하세요.

Amazon Managed Service for Prometheus 워크스페이스의 태그 보기(콘솔)

콘솔을 사용하여 Amazon Managed Service for Prometheus 워크스페이스와 연결된 태그를 볼 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. [Tags] 탭을 선택합니다.

Amazon Managed Service for Prometheus 워크스페이스의 태그 보기(AWS CLI)

AWS CLI를 사용하여 워크스페이스의 AWS 태그를 보려면 다음 단계를 수행하세요. 태그가 추가되지 않은 경우 반환된 목록은 비어 있습니다.

터미널 또는 명령줄에서 list-tags-for-resource 명령을 실행합니다. 예를 들어, 워크스페이스의 태그 키 및 태그 값 목록을 보려면 다음을 수행하세요.

```
aws amp list-tags-for-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
```

이 명령이 제대로 실행되면 다음과 비슷한 정보를 반환합니다.

```
{
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

워크스페이스의 태그 편집

워크스페이스와 연결된 태그에 대한 값을 변경할 수 있습니다. 또한 키 이름을 변경할 수 있습니다. 이는 현재 태그를 제거하고 새 이름 및 다른 키와 동일한 값을 가진 다른 태그를 추가하는 것과 동일합니다.

> **Important**
>
> Amazon Managed Service for Prometheus 워크스페이스의 태그를 편집하면 해당 워크스페이스에 대한 액세스에 영향을 미칠 수 있습니다. 워크스페이스의 태그 이름(키) 또는 값을 편집하기 전에 리포지토리와 같은 리소스에 대한 액세스를 제어하는 태그의 키 또는 값을 사용할 수도 있는 모든 IAM 정책을 검토하세요.

Amazon Managed Service for Prometheus 워크스페이스의 태그 편집(콘솔)

콘솔을 사용하여 Amazon Managed Service for Prometheus 워크스페이스와 연결된 태그를 편집할 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. [Tags] 탭을 선택합니다.
6. 워크스페이스에 추가된 태그가 없는 경우 태그 생성을 선택합니다. 그렇지 않으면 태그 관리를 선택합니다.
7. 키에 태그 이름을 입력합니다. 값(Value)에 태그의 선택적 값을 추가할 수 있습니다.
8. (선택 사항) 다른 태그를 추가하려면 다시 태그 추가를 선택합니다.
9. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.

Amazon Managed Service for Prometheus 워크스페이스의 태그 편집(AWS CLI)

AWS CLI를 사용하여 워크스페이스의 태그를 업데이트하려면 다음 단계를 수행하세요. 기존 키의 값을 변경하거나 다른 키를 추가할 수 있습니다.

터미널이나 명령줄에서 tag-resource 명령을 실행하여, 태그를 업데이트하고 태그 키 및 태그 값을 지정할 Amazon Managed Service for Prometheus 워크스페이스의 Amazon 리소스 이름(ARN)을 지정합니다.

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tags Team=New-Team
```

워크스페이스에서 태그 제거

워크스페이스와 연결된 하나 이상의 태그를 제거할 수 있습니다. 태그를 제거할 때 해당 태그와 연결된 다른 AWS 리소스에서 해당 태그가 삭제되지는 않습니다.

> **Important**
>
> Amazon Managed Service for Prometheus 워크스페이스의 태그를 제거하면 해당 워크스페이스에 대한 액세스에 영향을 미칠 수 있습니다. 워크스페이스에서 태그를 제거하기 전에 리포지토리와 같은 리소스에 대한 액세스를 제어하는 태그의 키 또는 값을 사용할 수도 있는 모든 IAM 정책을 검토하세요.

Amazon Managed Service for Prometheus 워크스페이스에서 태그 제거(콘솔)

콘솔을 사용하면 태그와 워크스페이스 간의 연결을 제거할 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. [Tags] 탭을 선택합니다.
6. 태그 관리를 선택합니다.
7. 삭제할 태그를 찾은 후 제거를 선택합니다.

Amazon Managed Service for Prometheus 워크스페이스에서 태그 제거(AWS CLI)

AWS CLI를 사용하여 워크스페이스의 태그를 제거하려면 다음 단계를 수행하세요. 태그를 제거하면 태그는 삭제되지 않고 태그와 워크스페이스 간의 연결만 제거됩니다.

> **Note**
>
> Amazon Managed Service for Prometheus 워크스페이스를 삭제하면 삭제된 워크스페이스에서 모든 태그 연결이 제거됩니다. 워크스페이스를 삭제하기 전에 태그를 제거할 필요가 없습니다.

터미널이나 명령줄에서 untag-resource 명령을 실행하여, 태그를 제거할 워크스페이스의 Amazon 리소스 이름(ARN)과 제거할 태그의 태그 키를 지정합니다. 예를 들어 My-Workspace라는 워크스페이스에서 태그 키 *Status*인 태그를 제거하려면 다음을 수행하세요.

```
aws amp untag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tag-keys Status
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다. 워크스페이스와 연결된 태그를 확인하려면 list-tags-for-resource 명령을 실행합니다.

규칙 그룹 네임스페이스 태그 지정

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에 태그를 지정하려면 이 섹션의 절차를 수행하세요.

주제

규칙 그룹 네임스페이스에 태그 추가

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에 태그를 추가하면 AWS 리소스를 식별 및 구성하고 해당 리소스에 대한 액세스를 관리할 수 있습니다. 먼저 규칙 그룹 네임스페이스에 하나 이상의 태그(키-값 페어)를 추가합니다. 태그가 생성된 후 해당 태그를 기준으로 네임스페이스에 대한 액세스를 관리하는 IAM 정책을 생성할 수 있습니다. 콘솔 또는 AWS CLI를 사용하여 Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에 태그를 추가할 수 있습니다.

> **Important**
>
> 규칙 그룹 네임스페이스에 태그를 추가하면 해당 규칙 그룹 네임스페이스에 대한 액세스에 영향을 미칠 수 있습니다. 태그를 추가하기 전에 태그를 사용하여 리소스에 대한 액세스를 제어할 수도 있는 모든 IAM 정책을 검토하세요.

규칙 그룹 네임스페이스를 생성할 때 규칙 그룹 네임스페이스에 태그를 추가하는 방법에 대한 자세한 내용은 규칙 파일 생성 섹션을 참조하세요.

주제

규칙 그룹 네임스페이스에 태그 추가(콘솔)

콘솔을 사용하여 Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에 하나 이상의 태그를 추가할 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. 규칙 관리 탭을 선택합니다.
6. 네임스페이스 이름 옆에 있는 버튼을 선택하고 편집을 선택합니다.
7. 태그 생성, 새 태그 추가를 선택합니다.
8. 키에 태그 이름을 입력합니다. 값(Value)에 태그의 선택적 값을 추가할 수 있습니다.
9. (선택 사항) 다른 태그를 추가하려면 다시 새 태그 추가를 선택합니다.
10. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.

규칙 그룹 네임스페이스에 태그 추가(AWS CLI)

다음 단계에 따라 AWS CLI를 사용하여 Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에 태그를 추가하세요. 규칙 그룹 네임스페이스를 생성할 때 태그를 추가하려면 Amazon Managed Service for Prometheus에 규칙 구성 파일 업로드 섹션을 참조하세요.

이 단계에서는 사용자가 이미 최신 버전의 AWS CLI를 설치했거나 현재 버전으로 업데이트했다고 가정합니다. 자세한 정보는 AWS Command Line Interface 설치 섹션을 참조하세요.

터미널이나 명령줄에서 tag-resource 명령을 실행하여, 태그를 추가할 규칙 그룹 네임스페이스의 Amazon 리소스 이름(ARN)과 추가할 태그의 키와 값을 지정합니다. 규칙 그룹 네임스페이스에 2개 이상의 태그를 추가할 수 있습니다. 예를 들어 My-Workspace라는 Amazon Managed Service for Prometheus 네임스페이스에 태그 키가 *Status*이고 태그 값이 *Secret*인 태그와 태그 키가 *Team*이고 태그 값이 *My-Team*인 2개의 태그를 지정하려면 다음과 같이 하세요.

```
aws amp tag-resource \
    --resource-arn arn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name \
```

```
    --tags Status=Secret,Team=My-Team
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다.

규칙 그룹 네임스페이스의 태그 보기

태그를 사용하면 AWS 리소스를 식별 및 구성하고 해당 리소스에 대한 액세스를 관리할 수 있습니다. 태깅 전략에 대한 자세한 내용은 AWS 리소스 태그 지정을 참조하세요.

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스의 태그 보기(콘솔)

콘솔을 사용하여 Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스와 연결된 태그를 볼 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. 규칙 관리 탭을 선택합니다.
6. 네임스페이스 이름을 선택합니다.

Amazon Managed Service for Prometheus 워크스페이스의 태그 보기(AWS CLI)

AWS CLI를 사용하여 규칙 그룹 네임스페이스의 AWS 태그를 보려면 다음 단계를 수행하세요. 태그가 추가되지 않은 경우 반환된 목록은 비어 있습니다.

터미널 또는 명령줄에서 list-tags-for-resource 명령을 실행합니다. 예를 들어 규칙 그룹 네임스페이스의 태그 키 및 태그 값 목록을 보려면 다음을 수행하세요.

```
aws amp list-tags-for-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name
```

이 명령이 제대로 실행되면 다음과 비슷한 정보를 반환합니다.

```
{
```

```
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

규칙 그룹 네임스페이스의 태그 편집

규칙 그룹 네임스페이스와 연결된 태그에 대한 값을 변경할 수 있습니다. 또한 키 이름을 변경할 수 있습니다. 이는 현재 태그를 제거하고 새 이름 및 다른 키와 동일한 값을 가진 다른 태그를 추가하는 것과 동일합니다.

> **Important**
>
> 규칙 그룹 네임스페이스의 태그를 편집하면 액세스에 영향을 미칠 수 있습니다. 리소스의 태그 이름(키) 또는 값을 편집하기 전에 태그의 키 또는 값을 사용하여 리소스에 대한 액세스를 제어할 수 있는 모든 IAM 정책을 검토해야 합니다.

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스의 태그 편집(콘솔)

콘솔을 사용하여 Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스와 연결된 태그를 편집할 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. 규칙 관리 탭을 선택합니다.
6. 네임스페이스의 이름을 선택합니다.
7. 태그 관리, 새 태그 추가를 선택합니다.
8. 기존 태그의 값을 변경하려면 값에 새 값을 입력합니다.
9. 태그를 더 추가하려면 새 태그 추가를 선택합니다.
10. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스의 태그 편집(AWS CLI)

AWS CLI를 사용하여 규칙 그룹 네임스페이스의 태그를 업데이트하려면 다음 단계를 수행하세요. 기존 키의 값을 변경하거나 다른 키를 추가할 수 있습니다.

터미널이나 명령줄에서 tag-resource 명령을 실행하여, 태그를 업데이트하고 태그 키 및 태그 값을 지정할 리소스의 Amazon 리소스 이름(ARN)을 지정합니다.

```
aws amp tag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tags Team=New-Team
```

규칙 그룹 네임스페이스에서 태그 제거

규칙 그룹 네임스페이스와 연결된 태그를 하나 이상 제거할 수 있습니다. 태그를 제거할 때 해당 태그와 연결된 다른 AWS 리소스에서 해당 태그가 삭제되지는 않습니다.

> **Important**
>
> 리소스의 태그를 제거하면 해당 리소스에 대한 액세스에 영향을 미칠 수 있습니다. 리소스에서 태그를 제거하기 전에 리포지토리와 같은 리소스에 대한 액세스를 제어하는 태그의 키 또는 값을 사용할 수도 있는 모든 IAM 정책을 검토하세요.

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에서 태그 제거(콘솔)

콘솔을 사용하면 태그와 규칙 그룹 네임스페이스 간의 연결을 제거할 수 있습니다.

1. https://console.aws.amazon.com/prometheus/에서 Amazon Managed Service for Prometheus 콘솔을 엽니다.
2. 탐색 창에서 메뉴 아이콘을 선택합니다.
3. 모든 워크스페이스를 선택합니다.
4. 관리해야 하는 워크스페이스의 워크스페이스 ID를 선택합니다.
5. 규칙 관리 탭을 선택합니다.
6. 네임스페이스의 이름을 선택합니다.
7. 태그 관리를 선택합니다.

8. 삭제할 태그 옆의 제거를 선택합니다.
9. 작업을 마쳤으면 변경 사항 저장을 선택합니다.

Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스에서 태그 제거 (AWS CLI)

AWS CLI를 사용하여 규칙 그룹 네임스페이스에서 태그를 제거하려면 다음 단계를 수행하세요. 태그를 제거하면 태그는 삭제되지 않고 태그와 규칙 그룹 네임스페이스 간의 연결만 제거됩니다.

> **Note**
>
> Amazon Managed Service for Prometheus 규칙 그룹 네임스페이스를 삭제하면 삭제된 네임스페이스에서 모든 태그 연결이 제거됩니다. 네임스페이스를 삭제하기 전에 태그를 제거할 필요가 없습니다.

터미널이나 명령줄에서 untag-resource 명령을 실행하여, 태그를 제거할 규칙 그룹 네임스페이스의 Amazon 리소스 이름(ARN)과 제거할 태그의 태그 키를 지정합니다. 예를 들어 My-Workspace라는 워크스페이스에서 태그 키 *Status*인 태그를 제거하려면 다음을 수행하세요.

```
aws amp untag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tag-keys Status
```

성공한 경우 이 명령은 아무 것도 반환하지 않습니다. 리소스와 연결된 태그를 확인하려면 list-tags-for-resource 명령을 실행합니다.

Amazon Managed Service for Prometheus 서비스 할당량

다음 두 섹션에서는 Amazon Managed Service for Prometheus와 관련된 할당량 및 제한에 대해 설명합니다.

Service quotas

Amazon Managed Service for Prometheus의 할당량은 다음과 같습니다. Prometheus용 Amazon 관리형 서비스는 Prometheus 리소스 CloudWatch 사용량을 모니터링하기 위해 사용량 지표를 제공합니다. CloudWatch 사용량 지표 알람 기능을 사용하면 Prometheus 리소스 및 사용량을 모니터링하여 한도 오류를 방지할 수 있습니다.

프로젝트와 WorkSpace가 확장되면서 모니터링하거나 증가를 요청해야 할 수 있는 가장 일반적인 할당량은 WorkSpace당 활성 시리즈, WorkSpace당 수집 속도, WorkSpace당 수집 버스트 크기입니다.

조정 가능한 모든 할당량의 경우 조정 가능 열의 링크를 선택하거나 할당량 증가를 요청하여 할당량 증가를 요청할 수 있습니다.

WorkSpace당 활성 시리즈 제한은 동적으로 적용됩니다. 자세한 설명은 활성 시리즈 기본값 섹션을 참조하세요. 작업 영역당 처리 속도와 작업 영역당 통합 버스트 크기를 함께 사용하면 작업 공간에 데이터를 수집하는 속도가 결정됩니다. 자세한 내용은 인제스트 스로틀링 섹션을 참조하세요.

> **Note**
>
> 달리 명시되지 않는 한, 이러한 할당량은 WorkSpace를 기준으로 합니다.

명칭	기본값	조정 가능	설명
WorkSpace별 메타데이터가 포함된 활성 지표	지원되는 각 리전: 20,000	아니요	WorkSpace당 메타데이터가 포함된 고유한 활성 지표의 수입니다.

명칭	기본값	조정 가능	설명
WorkSpace별 활성 시리즈	지원되는 각 리전: 2시간당 10,000,000	예	WorkSpace당 고유한 활성 시리즈 수입니다. 지난 2시간 동안 샘플이 보고된 경우 시리즈는 활성 상태입니다. 2M에서 10M까지의 용량은 지난 30분 사용량을 기준으로 자동으로 조정됩니다.
알림 관리자 정의 파일의 알림 집계 그룹 크기	지원되는 각 리전: 1,000개	예	알림 관리자 정의 파일에 있는 알림 집계 그룹의 최대 크기입니다. group_by의 각 레이블 값 조합은 집계 그룹을 생성합니다.
알림 관리자 정의 파일 크기	지원되는 각 리전: 1메가바이트	아니요	알림 관리자 정의 파일의 최대 크기.
알림 관리자의 알림 페이로드 크기	지원되는 각 리전: 20MB	아니요	작업 영역당 모든 Alert Manager 알림의 최대 경고 페이로드 크기입니다. 알림 크기는 레이블과 주석에 따라 달라집니다.
알림 관리자의 알림	지원되는 각 리전: 1,000개	예	작업 영역당 동시 경보 관리자 경고의 최대 수입니다.
HA 트래커 클러스터	지원되는 각 리전: 500	아니요	WorkSpace별로 수집된 샘플에 대해 HA 추적기가 추적하는 최대 클러스터 수입니다.

명칭	기본값	조정 가능	설명
WorkSpace별 수집 버스트 크기	지원되는 각 리전: 1,000,000	예	WorkSpace별로 수집할 수 있는 최대 샘플 수(초당 버스트 단위)입니다.
WorkSpace별 수집 속도	지원되는 각 리전: 170,000	예	초당 WorkSpace별 지표 샘플 수집 속도입니다.
알림 관리자 정의 파일의 금지 규칙	지원되는 각 리전: 100	예	알림 관리자 정의 파일의 최대 금지 규칙 수입니다.
라벨 크기	지원되는 각 리전: 7KB	아니요	시리즈에 허용되는 모든 라벨과 라벨 값을 합친 최대 크기입니다.
메트릭 시리즈별 레이블	지원되는 각 리전: 70	예	메트릭 시리즈당 라벨 수.
메타데이터 길이	지원되는 각 리전: 1KB	아니요	지표 메타데이터에 허용되는 최대 길이. 메타데이터는 지표 이름, HELP 및 UNIT을 나타냅니다.
지표별 메타데이터	지원되는 각 리전: 10	아니요	지표당 메타데이터 수.
알림 관리자 라우팅 트리의 노드	지원되는 각 리전: 100	예	알림 관리자 라우팅 트리의 최대 노드 수입니다.

명칭	기본값	조정 가능	설명
초당 트랜잭션의 API 작업 수	지원되는 각 리전: 10개	예	초당 수행할 수 있는 최대 API 작업 수입니다. 여기에는 WorkSpace CRUD API, 태그 지정 API, 규칙 그룹 네임스페이스 CRUD API 및 알림 관리자 정의 CRUD API가 포함됩니다.
인스턴트 쿼리를 위한 쿼리 바이트	지원되는 각 리전: 5GB	아니요	단일 인스턴트 쿼리로 스캔할 수 있는 최대 바이트 수.
범위 쿼리의 쿼리 바이트	지원되는 각 리전: 5GB	아니요	단일 범위 쿼리에서 24시간 간격으로 스캔할 수 있는 최대 바이트 수입니다.
쿼리 청크를 가져왔습니다.	지원되는 각 리전: 20,000,000	아니요	단일 쿼리 중에 스캔할 수 있는 최대 청크 수입니다.
샘플 쿼리	지원되는 각 리전: 50,000,000	아니요	단일 쿼리 중에 스캔할 수 있는 최대 샘플 수입니다.
쿼리 시리즈를 가져왔습니다	지원되는 각 리전: 12,000,000	아니요	단일 쿼리 중에 스캔할 수 있는 최대 시리즈 수입니다.
쿼리 시간 범위(일)	지원되는 각 리전: 32	아니요	모든 PromQL 쿼리의 최대 시간 범위입니다.

명칭	기본값	조정 가능	설명
요청 크기	지원되는 각 리전: 1메가바이트	아니요	수집 또는 쿼리의 최대 요청 크기입니다.
수집된 데이터의 보존 시간(일)	지원되는 각 리전: 150	예	WorkSpace의 데이터가 보존되는 일수입니다. 이보다 오래된 데이터는 삭제됩니다. 할당량 변경을 요청하여 이 값을 늘리거나 줄일 수 있습니다.
규칙 평가 간격	지원되는 각 리전: 30초	예	WorkSpace별 규칙 그룹의 최소 규칙 평가 간격입니다.
규칙 그룹 네임스페이스 정의 파일 크기	지원되는 각 리전: 1메가바이트	아니요	규칙 그룹 네임스페이스 정의 파일의 최대 크기.
WorkSpace별 규칙	지원되는 각 리전: 2,000	예	WorkSpace별 최대 규칙 수입니다.
알림 관리자 정의 파일의 템플릿	지원되는 각 리전: 100	예	알림 관리자 정의 파일의 최대 템플릿 수.
계정당 리전별 WorkSpace	지원되는 각 지역: 25	예	리전별 최대 WorkSpace 수입니다.

활성 시리즈 기본값

Amazon Managed Service for Prometheus에서는 기본적으로 활성 시계열 할당량까지 사용할 수 있습니다.

Amazon Managed Service for Prometheus WorkSpace는 수집 볼륨에 맞게 자동으로 조정됩니다. 사용량이 증가하면 Amazon Managed Service for Prometheus에서 자동으로 시계열 용량을 늘려 기본 할당량까지 기준 사용량을 두 배로 늘립니다. 예를 들어 최근 30분 동안의 평균 활성 시계열이 350만 개인 경우 조절 없이 최대 700만 개 시계열을 사용할 수 있습니다.

이전 기준의 두 배가 넘는 용량이 필요한 경우 Amazon Managed Service for Prometheus는 수집 볼륨이 증가함에 따라 더 많은 용량을 자동으로 할당하여 워크로드에 지속적인 조절이 발생하지 않도록 할당량까지 보장합니다. 하지만 지난 30분 동안 계산된 이전 기준의 두 배를 초과하는 경우 조절이 발생할 수 있습니다. 조절을 방지하기 위해 Amazon Managed Service for Prometheus에서는 이전 활성 시계열의 두 배를 넘도록 수집량을 늘리는 것이 좋습니다.

> **Note**
>
> 활성 시계열의 최소 용량은 2백만 개이며, 시계열 수가 2백만 개 미만인 경우 조절이 발생하지 않습니다.
> 기본 할당량을 초과하려면 할당량 증가를 요청할 수 있습니다.

인제스트 스로틀링

Prometheus용 Amazon 관리형 서비스는 현재 한도를 기준으로 각 작업 영역에 대한 데이터 수집을 제한합니다. 이는 워크스페이스의 성능을 유지하는 데 도움이 됩니다. 한도를 초과하면 `DiscardedSamples` CloudWatch 지표에 (`rate_limited`이유 포함) 이 표시됩니다. CloudWatch Amazon을 사용하여 섭취량을 모니터링하고 스로틀링 한도에 가까워지면 경고하는 경보를 생성할 수 있습니다. 자세한 설명은 CloudWatch 메트릭 섹션을 참조하세요.

Prometheus용 Amazon 관리형 서비스는 토큰 버킷 알고리즘을 사용하여 수집 제한을 구현합니다. 이 알고리즘을 사용하면 계정에 특정 수의 토큰을 보관하는 버킷이 있습니다. 버킷의 토큰 수는 특정 초당 수집 한도를 나타냅니다.

수집된 각 데이터 샘플은 버킷에서 토큰 하나를 제거합니다. 버킷 크기 (작업 영역당 통합 버스트 크기)가 1,000,000인 경우 작업 공간은 1초에 100만 개의 데이터 샘플을 수집할 수 있습니다. 수집할 샘플이 100만 개를 초과하는 경우 샘플이 병목 현상을 일으키고 더 이상 레코드를 수집하지 않습니다. 추가 데이터 샘플은 삭제됩니다.

버킷은 설정된 속도로 자동으로 리필됩니다. 버킷이 최대 용량 이하인 경우 최대 용량에 도달할 때까지 1초마다 정해진 수의 토큰이 버킷에 다시 추가됩니다. 리필 토큰이 도착했을 때 버킷이 가득 차면 토큰은 폐기됩니다. 버킷에는 최대 토큰 수보다 많은 토큰을 담을 수 없습니다. 샘플 수집의 리필 비율은 작

업 공간당 처리 속도 한도에 따라 설정됩니다. 작업 영역당 처리 속도가 170,000으로 설정된 경우 버킷의 리필 속도는 초당 170,000 토큰입니다.

작업 공간이 1초에 1,000,000개의 데이터 샘플을 수집하는 경우 버킷은 즉시 토큰 0으로 줄어듭니다. 그러면 최대 토큰 1,000,000개에 도달할 때까지 매초 170,000개의 토큰이 버킷에 다시 채워집니다. 더 이상 수집이 없을 경우 이전에 비어 있던 버킷은 6초 후에 최대 용량으로 돌아갑니다.

> **Note**
>
> 인제스트는 일괄 요청에서 이루어집니다. 사용 가능한 토큰이 100개이고 샘플 101개가 포함된 요청을 보내면 전체 요청이 거부됩니다. Prometheus용 Amazon 관리형 서비스는 요청을 부분적으로 수락하지 않습니다. 컬렉터를 작성하는 경우 재시도 (배치 수를 줄이거나 일정 시간이 지난 후) 를 관리할 수 있습니다.

작업 영역에서 더 많은 데이터 샘플을 수집하기 전에 버킷이 가득 찰 때까지 기다릴 필요가 없습니다. 버킷에 추가된 토큰은 그대로 사용할 수 있습니다. 리필 토큰을 즉시 사용하면 버킷이 최대 용량에 도달하지 못합니다. 예를 들어 버킷을 고갈시키더라도 초당 170,000개의 데이터 샘플을 계속 수집할 수 있습니다. 초당 170,000개 미만의 데이터 샘플을 수집하는 경우에만 버킷을 최대 용량까지 채울 수 있습니다.

수집된 데이터에 대한 추가 제한

Amazon Managed Service for Prometheus에서는 WorkSpace로 수집된 데이터에 대해 다음과 같은 추가 요구 사항이 적용됩니다. 이 설정은 조정할 수 없습니다.

- 1시간 이상 경과된 측정 샘플은 섭취가 거부됩니다.
- 모든 샘플과 메타데이터에는 지표 이름이 있어야 합니다.

API 참조

이 섹션에는 Amazon Managed for Amazon Managed for Prometheus에서 지원하는 API 작업 및 데이터 구조가 나열되어 있습니다.

이러한 API 작업과 시리즈, 레이블 및 API 요청의 할당량에 대한 자세한 내용은 Amazon Managed Service for Prometheus 사용 설명서에서 Amazon Managed Service for Prometheus 서비스 할당량을 참조하세요.

주제

- Amazon Managed Service for Prometheus API
- Prometheus 호환 API

Amazon Managed Service for Prometheus API

프로메테우스용 아마존 매니지드 서비스는 프로메테우스용 아마존 매니지드 서비스 워크스페이스를 생성하고 유지 관리하는 API 작업을 제공합니다. 여기에는 워크스페이스, 스크레이퍼, 알림 관리자 정의, 규칙 그룹 네임스페이스 및 로깅을 위한 API가 포함됩니다.

Prometheus API용 Amazon 관리 서비스에 대한 자세한 내용은 Prometheus API용 Amazon 관리 서비스 참조를 참조하십시오.

SDK와 함께 Prometheus용 아마존 매니지드 서비스 사용 AWS

AWS 소프트웨어 개발 키트 (SDK) 는 널리 사용되는 여러 프로그래밍 언어로 제공됩니다. 각 SDK는 개발자가 선호하는 언어로 AWS 애플리케이션을 쉽게 빌드할 수 있도록 API, 코드 예제 및 설명서를 제공합니다. 언어별 SDK 및 도구 목록은 AWS 개발자 센터에서 빌드할 도구를 참조하십시오. AWS

> **SDK 버전**
>
> 프로젝트에서 사용하는 가장 최신 AWS SDK 빌드와 기타 SDK를 사용하고 SDK를 최신 상태로 유지하는 것이 좋습니다. AWS SDK는 최신 특징과 기능과 보안 업데이트도 제공합니다.

Prometheus 호환 API

Amazon Managed Service for Prometheus에서는 다음 Prometheus 호환 API를 지원합니다.

Prometheus 호환 API 사용에 대한 자세한 내용은 을 참조하십시오. Prometheus 호환 API를 사용한 쿼리

주제

- CreateAlertManagerAlerts
- DeleteAlertManagerSilence
- GetAlertManagerStatus
- GetAlertManagerSilence
- GetLabels
- GetMetricMetadata
- GetSeries
- ListAlerts
- ListAlertManagerAlerts
- ListAlertManagerAlertGroups
- ListAlertManagerReceivers
- ListAlertManagerSilences
- ListRules
- PutAlertManagerSilences
- QueryMetrics
- RemoteWrite

CreateAlertManagerAlerts

`CreateAlertManagerAlerts` 작업은 워크스페이스에 알림을 생성합니다.

유효한 HTTP 동사:

`POST`

유효한 URI:

`/workspaces/workspaceId/alertmanager/api/v2/alerts`

URL 쿼리 파라미터:

`alerts` 각 객체가 하나의 알림을 나타내는 객체 배열입니다. 다음은 알림 객체의 예제입니다.

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

샘플 요청

```
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/"
  }
]
```

샘플 응답

```
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

DeleteSilence는 무음 알림 하나를 삭제합니다.

유효한 HTTP 동사:

DELETE

유효한 URI:

/workspaces/*workspaceId*/alertmanager/api/v2/silence/*silenceID*

URL 쿼리 파라미터: 없음

샘플 요청

```
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
```

샘플 응답

```
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

GetAlertManagerStatus는 알림 관리자의 상태에 대한 정보를 검색합니다.

유효한 HTTP 동사:

GET

유효한 URI:

/workspaces/*workspaceId*/alertmanager/api/v2/status

URL 쿼리 파라미터: 없음

샘플 요청

```
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
```

샘플 응답

```
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

GetAlertManagerSilence는 무음 알림 하나에 대한 정보를 검색합니다.

유효한 HTTP 동사:

GET

유효한 URI:

/workspaces/*workspaceId*/alertmanager/api/v2/silence/*silenceID*

URL 쿼리 파라미터: 없음

샘플 요청

```
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
```

샘플 응답

```
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 작업은 시계열과 관련된 레이블을 검색합니다.

유효한 HTTP 동사:

GET, POST

유효한 URI:

/workspaces/*workspaceId*/api/v1/labels

/workspaces/*workspaceId*/api/v1/label/*label-name*/values 이 URI는 GET 요청만 지원합니다.

URL 쿼리 파라미터:

match[]=<series_selector> 레이블 이름을 읽을 시리즈를 선택하는 반복 시리즈 선택기 인수입니다. 선택 사항입니다.

start=<rfc3339 | unix_timestamp> 시작 타임스탬프입니다. 선택 사항입니다.

end=<rfc3339 | unix_timestamp> 종료 타임스탬프입니다. 선택 사항입니다.

/workspaces/workspaceId/api/v1/labels에 대한 샘플 요청

```
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
```

/workspaces/workspaceId/api/v1/labels에 대한 샘플 응답

```
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",
```

```
        ...
    ]
}
```

/workspaces/workspaceId/api/v1/label/label-name/values에 대한 샘플 요청

```
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
```

/workspaces/workspaceId/api/v1/label/label-name/values에 대한 샘플 응답

```
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 작업은 대상에서 현재 스크래핑 중인 지표에 대한 메타데이터를 검색합니다. 대상 정보는 제공하지 않습니다.

쿼리 결과의 데이터 섹션은 각 키가 지표 이름이고 각 값이 모든 대상에서 해당 지표 이름에 대해 노출되는 고유한 메타데이터 객체 목록인 객체로 구성됩니다.

유효한 HTTP 동사:

- GET

유효한 URI:

`/workspaces/workspaceId/api/v1/metadata`

URL 쿼리 파라미터:

`limit=<number>` 반환할 최대 지표 수입니다.

`metric=<string>` 메타데이터를 필터링할 때 지표 이름입니다. 이 파라미터를 비워 두면 모든 지표 메타데이터가 검색됩니다.

샘플 요청

```
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
```

샘플 응답

```
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 작업은 특정 레이블 세트와 일치하는 시계열 목록을 검색합니다.

유효한 HTTP 동사:

GET, POST

유효한 URI:

/workspaces/*workspaceId*/api/v1/series

URL 쿼리 파라미터:

match[]=<series_selector> 반환할 시리즈를 선택하는 반복 시리즈 선택기 인수입니다. 1개 이상의 match[] 인수를 제공해야 합니다.

start=<rfc3339 | unix_timestamp> 시작 타임스탬프입니다. 선택 사항

end=<rfc3339 | unix_timestamp> 종료 타임스탬프입니다. 선택 사항

샘플 요청

```
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
```

샘플 응답

```
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` 작업은 워크스페이스에서 현재 활성 상태인 알림을 검색합니다.

유효한 HTTP 동사:

```
GET
```

유효한 URI:

```
/workspaces/workspaceId/api/v1/alerts
```

샘플 요청

```
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
```

샘플 응답

```
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

ListAlertManagerAlerts는 워크스페이스의 알림 관리자에서 현재 발생하고 있는 알림에 대한 정보를 검색합니다.

유효한 HTTP 동사:

GET

유효한 URI:

/workspaces/*workspaceId*/alertmanager/api/v2/alerts

샘플 요청

```
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
```

샘플 응답

```
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` 작업은 워크스페이스의 알림 관리자에 구성된 알림 그룹 목록을 검색합니다.

유효한 HTTP 동사:

`GET`

유효한 URI:

`/workspaces/workspaceId/alertmanager/api/v2/alerts/groups`

URL 쿼리 파라미터:

`active` 부울입니다. true인 경우 반환된 목록에 활성 알림이 포함됩니다. 기본값은 true입니다. 선택 사항

`silenced` 부울입니다. true인 경우 반환된 목록에는 무음 알림이 포함됩니다. 기본값은 true입니다. 선택 사항

`inhibited` 부울입니다. true인 경우 반환된 목록에는 금지된 알림이 포함됩니다. 기본값은 true입니다. 선택 사항

`filter` 문자열 배열입니다. 알림을 필터링할 매처의 목록입니다. 선택 사항

`receiver` 문자열입니다. 알림을 필터링할 수신기를 일치시키는 정규 표현식입니다. 선택 사항

샘플 요청

```
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
```

샘플 응답

```
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` 작업은 알림 관리자에 구성된 수신기에 대한 정보를 검색합니다.

유효한 HTTP 동사:

`GET`

유효한 URI:

`/workspaces/workspaceId/alertmanager/api/v2/receivers`

URL 쿼리 파라미터: 없음

샘플 요청

```
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
```

샘플 응답

```
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` 작업은 워크스페이스에 구성된 무음 알림에 대한 정보를 검색합니다.

유효한 HTTP 동사:

`GET`

유효한 URI:

`/workspaces/workspaceId/alertmanager/api/v2/silences`

샘플 요청

```
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
```

샘플 응답

```
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

ListRules는 워크스페이스에 구성된 규칙에 대한 정보를 검색합니다.

유효한 HTTP 동사:

GET

유효한 URI:

/workspaces/*workspaceId*/api/v1/rules

샘플 요청

```
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
```

샘플 응답

```
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` 작업은 새 무음 알림을 생성하거나 기존 무음 알림을 업데이트합니다.

유효한 HTTP 동사:

`POST`

유효한 URI:

`/workspaces/workspaceId/alertmanager/api/v2/silences`

URL 쿼리 파라미터:

`silence` 무음을 나타내는 객체입니다. 형식은 다음과 같습니다.

```
{
  "id": "string",
  "matchers": [
    {
      "name": "string",
      "value": "string",
      "isRegex": Boolean,
      "isEqual": Boolean
    }
  ],
  "startsAt": "timestamp",
  "endsAt": "timestamp",
  "createdBy": "string",
  "comment": "string"
}
```

샘플 요청

```
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"
}
```

샘플 응답

```
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` 작업은 단일 시점 또는 일정 기간 동안 인스턴트 쿼리를 평가합니다.

유효한 HTTP 동사:

- `GET`, `POST`

유효한 URI:

- `/workspaces/workspaceId/api/v1/query` 이 URI는 단일 시점의 인스턴트 쿼리를 평가합니다.
- `/workspaces/workspaceId/api/v1/query_range` 이 URI는 일정 기간 동안의 인스턴트 쿼리를 평가합니다.

URL 쿼리 파라미터:

query=<string> Prometheus 표현식 쿼리 문자열입니다. query 및 query_range 둘 다에 사용됩니다.

time=<rfc3339 | unix_timestamp> (선택 사항) 단일 시점에서 인스턴트 쿼리에 query를 사용하는 경우 평가 타임스탬프입니다.

timeout=<duration> (선택 사항) 평가 시간 초과입니다. 기본값은 -query.timeout 플래그 값으로 제한됩니다. query 및 query_range 둘 다에 사용됩니다.

start=<rfc3339 | unix_timestamp> query_range를 사용하여 기간에 대해 쿼리하는 경우 시작 타임스탬프입니다.

end=<rfc3339 | unix_timestamp> query_range를 사용하여 기간에 대해 쿼리하는 경우 종료 타임스탬프입니다.

step=<duration | float> duration 형식 또는 float초 단위로 나타내는 쿼리 해결 단계 폭입니다. query_range를 사용하여 일정 기간 동안 쿼리하는 경우에만 사용하며, 해당 쿼리에 필요합니다.

지속 시간

Prometheus 호환 API의 duration은 숫자이며, 그 뒤에 바로 다음 단위 중 하나가 따라옵니다.

- ms밀리초
- s초
- m분
- h시간
- d일(항상 하루를 24시간으로 가정)
- w주(항상 한 주를 7일로 가정)
- y년(항상 1년을 365일로 가정)

샘플 요청

```
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
```

샘플 응답

```
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 작업은 Prometheus 서버의 지표를 원격 URL에 표준화된 형식으로 기록합니다. 일반적으로 Prometheus 서버와 같은 기존 클라이언트를 사용하여 이 작업을 호출합니다.

유효한 HTTP 동사:

POST

유효한 URI:

/workspaces/*workspaceId*/api/v1/remote_write

URL 쿼리 파라미터:

None

RemoteWrite의 수집 속도는 초당 70,000개 샘플이고 수집 버스트 크기는 1,000,000개 샘플입니다.

샘플 요청

```
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**
>
> 요청 본문 구문은 https://github.com/prometheus/prometheus/blob/1c624c58ca934f618be737b4995e22051f5724c1/prompb/remote.pb.go#L64에서 프로토콜 버퍼 정의를 참조하세요.

샘플 응답

```
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
```

Amazon Managed Service for Prometheus 사용 설명서의 문서 기록

다음 표에는 Amazon Managed Service for Prometheus 사용 설명서의 중요 설명서 업데이트가 설명되어 있습니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

변경 사항	설명	날짜
API를 별도의 AWS API 참조 가이드로 이동	이제 Prometheus API용 Amazon 관리 서비스 API는 자체 참조인 AWS Prometheus 용 Amazon 관리 서비스 API 참조에서 사용할 수 있습니다. Prometheus 호환 API는 Prometheus용 Amazon 관리형 서비스 사용 설명서에 계속 문서화되어 있습니다.	2024년 2월 7일
WorkSpace 암호화를 위한 고객 관리형 키가 추가되었습니다.	Amazon Managed Service for Prometheus는 WorkSpace 암호화를 위한 고객 관리형 키에 대한 지원을 추가합니다. 자세한 내용은 저장된 데이터 암호화를 참조하세요.	2023년 12월 21일
에 새 권한이 추가되었습니다. AmazonPrometheusFullAccess	Amazon EKS 클러스터용 AmazonPrometheusFullAccess관리형 수집기 생성을 지원하는 새 권한을 AWS 관리형 정책에 추가했습니다.	2023년 11월 26일
새 관리형 정책이 추가되었습니다. AmazonPrometheusScraperServiceLinkedRolePolicy	Amazon EKS 클러스터에서 지표를 수집할 수 있는 AWS 관리형 수집기에 AmazonPrometheusScraperServiceLinked	2023년 11월 26일

	RolePolicy대한 새 관리형 정책이 추가되었습니다.	
AWS 관리 컬렉터를 수집 방법으로 추가했습니다.	Amazon Managed Service for Prometheus는 AWS 관리형 수집기에 대한 지원을 추가합니다.	2023년 11월 26일
Amazon Managed Grafana와의 통합에 대한 지원이 추가되었습니다	Amazon Managed Service for Prometheus는 Amazon Managed Grafana 알림과의 통합을 위한 지원을 추가합니다.	2022년 11월 23일
에 새 권한이 추가되었습니다. AmazonPrometheusConsoleFullAccess	CloudWatch 로그에 경고 관리자 및 눈금자 이벤트 로깅을 지원하는 새 권한을 AmazonPrometheusConsoleFullAccess관리형 정책에 추가했습니다.	2022년 10월 24일
Amazon EKS 관찰성 솔루션이 추가되었습니다.	Prometheus용 Amazon 매니지드 서비스는 옵저버빌리티 액셀러레이터를 사용하는 새로운 솔루션을 추가합니다. AWS 자세한 내용은 AWS Observability Accelerator 사용을 참조하세요.	2022년 10월 14일
Amazon EKS 비용 모니터링에 통합하기 위한 지원이 추가되었습니다.	Amazon Managed Service for Prometheus는 Amazon EKS 비용 모니터링에 통합하기 위한 지원을 추가합니다. 자세한 내용은 Amazon EKS 비용 모니터링과 통합을 참조하세요.	2022년 9월 22일

Amazon CloudWatch Logs의 알림 관리자 및 눈금자 로그에 대한 지원을 시작했습니다.	Prometheus용 아마존 매니지드 서비스가 Amazon Logs의 알림 관리자 및 눈금자 오류 로그에 대한 지원을 시작합니다. CloudWatch 자세한 내용은 Amazon CloudWatch Logs를 참조하십시오.	2022년 9월 1일
사용자 지정 스토리지 보존 지원이 추가되었습니다.	Amazon Managed Service for Prometheus는 해당 WorkSpace의 할당량을 수정하여 WorkSpace별 사용자 지정 스토리지 보존 지원을 추가합니다. Amazon Managed Service for Prometheus의 할당량에 대한 자세한 내용은 서비스 할당량을 참조하세요.	2022년 8월 12일
Amazon에 사용량 지표를 추가했습니다 CloudWatch.	Prometheus용 Amazon 관리 서비스는 Amazon에 사용량 지표를 전송하는 지원을 추가합니다. CloudWatch 자세한 내용은 Amazon CloudWatch 지표를 참조하십시오.	2022년 5월 6일
유럽(런던) 리전에 대한 지원이 추가되었습니다.	Amazon Managed Service for Prometheus에서 유럽(런던) 리전에 대한 지원을 추가합니다.	2022년 5월 4일
Amazon Managed Service for Prometheus가 일반적으로 사용 가능하며 규칙 및 알림 관리자에 대한 지원이 추가되었습니다.	Amazon Managed Service for Prometheus를 일반적으로 사용할 수 있습니다. 규칙 및 알림 관리자도 지원합니다. 자세한 내용을 알아보려면 기록 규칙 및 알림 규칙과 알림 관리자 및 템플릿을 참조하세요.	2021년 9월 29일

태깅 지원이 추가되었습니다.	Amazon Managed Service for Prometheus는 Amazon Managed Service for Prometheus WorkSpace의 태그 지정을 지원합니다.	2021년 9월 7일
활성 시리즈 및 수집 비율 할당량이 증가했습니다.	활성 시리즈 할당량은 1,000,000개로 증가했고 수집 속도 할당량은 초당 70,000개 샘플로 증가했습니다.	2021년 2월 22일
Amazon Managed Service for Prometheus 미리 보기 릴리스.	Amazon Managed Service for Prometheus의 미리 보기가 릴리스되었습니다.	2020년 12월 15일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.