aws

用户指南

Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus: 用户指南

Table of Contents

什么是 Amazon Managed Service for Prometheus？

Amazon Managed Service for Prometheus 是一项面向容器指标的无服务器 Prometheus 兼容监控服务，有助于更轻松地实现对容器环境的大规模监控。借助 Amazon Managed Service for Prometheus，您可以使用目前所用的开源 Prometheus 数据模型和查询语言来监控容器化工作负载的性能，还可以享受更高的可扩展性、可用性和安全性，而无需管理底层基础设施。

Amazon Managed Service for Prometheus 会随着工作负载的扩展和缩减而自动扩展运行指标的摄取、存储和查询。它与 AWS 安全服务集成，可以快速、安全地访问数据。

Amazon Managed Service for Prometheus 旨在使用多个可用区（多可用区）部署实现高可用性。摄取到工作区的数据将在同一区域的三个可用区中复制。

Amazon Managed Service for Prometheus 适用于在 Amazon Elastic Kubernetes Service 上运行的容器集群和自行管理的 Kubernetes 环境。

使用 Amazon Managed Service for Prometheu，您可以使用与 Prometheus 相同的开源 Prometheus 数据模型和 PromQL 查询语言。工程团队可以使用 PromQL 对指标进行筛选、汇总和设置警报，无需更改任何代码即可快速获得性能可见性。Amazon Managed Service for Prometheus 提供了灵活的查询功能，而不会产生运营成本和复杂性。

默认情况下，采集到工作空间的指标会存储 150 天，然后会自动删除。此长度是可调整的配额。

支持的区域

Amazon Managed Service for Prometheus 目前支持以下区域：

区域名称	区域	端点	协议	
美国东部（俄亥俄州）	us-east-2	aps.us-east-2.amazonaws.com aps-workspaces.us-east-2.amazonaws.com	HTTPS HTTPS	
美国东部（弗吉尼亚州北部）	us-east-1	aps.us-east-1.amazonaws.com aps-workspaces.us-east-1.amazonaws.com	HTTPS HTTPS	

区域名称	区域	端点	协议
美国西部（俄勒冈州）	us-west-2	aps.us-west-2.amazonaws.com	HTTPS
		aps-workspaces.us-west-2.amazonaws.com	HTTPS
亚太地区(孟买)	ap-south-1	aps.ap-south-1.amazonaws.com	HTTPS
		aps-workspaces.ap-south-1.amazonaws.com	HTTPS
亚太地区（首尔）	ap-northeast-2	aps.ap-northeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-2.amazonaws.com	HTTPS
亚太地区（新加坡）	ap-southeast-1	aps.ap-southeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-1.amazonaws.com	HTTPS
亚太地区（悉尼）	ap-southeast-2	aps.ap-southeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-2.amazonaws.com	HTTPS
亚太地区（东京)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-1.amazonaws.com	HTTPS
欧洲（法兰克福）	eu-central-1	aps.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.amazonaws.com	HTTPS
欧洲地区（爱尔兰）	eu-west-1	aps.eu-west-1.amazonaws.com	HTTPS
		aps-workspaces.eu-west-1.amazonaws.com	HTTPS
欧洲地区（伦敦）	eu-west-2	aps.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.amazonaws.com	HTTPS

区域名称	区域	端点	协议
欧洲地区（巴黎）	eu-west-3	aps.eu-west-3.amazonaws.com aps-workspaces.eu-west-3.amazonaws.com	HTTPS HTTPS
欧洲地区（斯德哥尔摩）	eu-north-1	aps.eu-north-1.amazonaws.com aps-workspaces.eu-north-1.amazonaws.com	HTTPS HTTPS
南美洲（圣保罗）	sa-east-1	aps.sa-east-1.amazonaws.com aps-workspaces.sa-east-1.amazonaws.com	HTTPS HTTPS

定价

您需要为摄取和存储指标付费。存储费用基于指标样本和元数据的压缩大小。有关更多信息，请参阅 Amazon Managed Service for Prometheus 定价。

您可以使用 Cost Explorer 和 AWS 成本和使用情况报告来监控您的费用。有关更多信息，请参阅使用 Cost Explorer 浏览您的数据和什么是 AWS 成本和使用情况报告。

高级支持

如果您订阅了任何级别的 AWS 高级支持计划，则您的高级支持适用于适用于 Prometheus 的亚马逊托管服务。

开始使用

本部分介绍如何快速创建 Amazon Managed Service for Prometheus 工作区、如何为这些工作区设置 Prometheus 指标的摄取以及如何查询这些指标。

它还包括有关设置的信息 AWS 账户，以备不时之需 AWS。

主题

设置

完成本节中的任务，以便首次 AWS 进行设置。如果您已经有一个 AWS 帐户，请直接跳至创建工作区。

注册后 AWS，您的 AWS 账户会自动访问中的 AWS所有服务，包括适用于 Prometheus 的亚马逊托管服务。不过，您只需为使用的服务付费。

主题

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。在任何时候，您都可以通过转至 https://aws.amazon.com/ 并选择我的账户来查看当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的以根用户身份登录。
2. 为您的根用户启用多重身份验证 (MFA)。

   有关说明，请参阅 I A M 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备（控制台）。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的启用 AWS IAM Identity Center。
2. 在 IAM Identity Center 中，为用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

有关使用 IAM Identity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的创建权限集。
2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的添加组。

创建工作区

工作区是专用于存储和查询 Prometheus 指标的逻辑空间。工作区支持精细的访问控制，用于授权其管理，例如更新、列出、描述和删除，以及指标的摄取和查询等。您可以在账户的每个区域中有一个或多个工作区。

要设置工作区，请按照以下步骤操作。

> **Note**
>
> 有关创建工作区的更详细信息，请参阅创建工作区。

创建 Amazon Managed Service for Prometheus 工作区

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在工作区别名中，输入新工作区的别名。

   工作区别名是友好名称，有助于您识别工作区。名称没有必要是唯一的。两个工作区可以具有相同的别名，但所有工作区都将有唯一的工作区 ID，这些 ID 由 Amazon Managed Service for Prometheus 生成。
3. （可选）要向命名空间添加标签，请选择添加新标签。

   然后，对于 Key（键），输入标签的名称。您可以在 Value (值) 中添加可选的标签值。

要添加其他标签，请再次选择 Add new tag（添加新标签）。

4. 选择创建工作区。

   此时将会显示工作区详细信息页面。这将显示该工作区的状态、ARN、工作区 ID 和终端节点 URL 等信息，用于远程写入和查询。

   最初，状态可能为正在创建。等到状态变为活动后再继续设置指标摄取。

   记下终端节点 - 远程写入 URL 和终端节点 - 查询 URL 中显示的 URL。当您将 Prometheus 服务器配置为将指标远程写入此工作区以及查询这些指标时，将需要这些 URL。

将 Prometheus 指标摄取到工作区

摄取指标的一种方法是使用独立的 Prometheus 代理（在代理模式下运行的 Prometheus 实例）从集群中抓取指标，然后将其转发到 Amazon Managed Service for Prometheus 进行存储和监控。本节介绍如何通过使用 Helm 设置新的 Prometheus 代理实例，将指标摄取从 Amazon EKS 设置为 Amazon Managed Service for Prometheus 工作区。

有关向 Amazon Managed Service for Prometheus 摄取数据的其他方法（包括如何保护指标和创建高可用性指标）的信息，请参阅将指标提取到您的工作空间。

> **Note**
>
> 默认情况下，采集到工作空间的指标会存储 150 天，然后会自动删除。此长度是可调整的配额。

本部分中的说明有助于您快速启动并运行 Amazon Managed Service for Prometheus。您在 Amazon EKS 集群中设置了一台新的 Prometheus 服务器，新服务器使用默认配置作为代理，向 Amazon Managed Service for Prometheus 发送指标。本方法包含以下先决条件：

- 您必须有一个 Amazon EKS 集群，新的 Prometheus 服务器将从中收集指标。
- 您必须使用 Helm CLI 3.0 或更高版本
- 您必须使用 Linux 或 macOS 计算机来执行以下各部分中的步骤。

步骤 1：添加新的 Helm 图表存储库

输入以下命令以添加新的 Helm 存储库。有关这些命令的更多信息，请参阅 Helm 存储库。

```
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
```

步骤 2：创建 Prometheus 命名空间

输入以下命令，为 Prometheus 服务器和其它监控组件创建 Prometheus 命名空间。将 *prometheus-agent-namespace* 替换为想要的命名空间名称。

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

步骤 3：设置服务账户的 IAM 角色。

要使用这种摄取方法，您需要为运行 Prometheus 代理的 Amazon EKS 集群中的服务账户使用 IAM 角色。

通过服务账户的 IAM 角色，您可以将 IAM 角色与 Kubernetes 服务账户关联。然后，此服务账户可向使用它的任意 Pod 中的容器提供 AWS 权限。有关更多信息，请参阅服务账户的 IAM 角色。

如果您尚未设置这些角色，请按照 设置服务角色从 Amazon EKS 集群中摄取指标 中的说明设置角色。该部分中的说明要求使用 `eksctl`。有关更多信息，请参阅 Amazon Elastic Kubernetes Service 入门 – `eksctl`。

> **Note**
>
> 如果您不在 EKS 上，或者仅使用访问密钥 AWS 和私有密钥访问适用于 Prometheus 的亚马逊托管服务，则无法使用基于的 Sigv4。`EKS-IAM-ROLE`

步骤 4：设置新服务器并开始摄取指标

要安装新 Prometheus 代理并将指标发送到 Amazon Managed Service for Prometheus 工作区，请按照以下步骤操作。

安装新 Prometheus 代理并将指标发送到 Amazon Managed Service for Prometheus 工作区

1. 使用文本编辑器创建名为 my_prometheus_values_yaml 的文件，其中包含以下内容。

   - 将 *IAM_PROXY_PROMETHEUS_ROLE_ARN* 替换为您在 设置服务角色从 Amazon EKS 集群中摄取指标 中创建的 amp-iamproxy-ingest-role。
   - 将 *WORKSPACE_ID* 替换为您 Amazon Managed Service for Prometheus 工作区的 ID。
   - 将 *REGION* 替换为您 Amazon Managed Service for Prometheus 工作区的区域。

```
## 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. 输入以下命令以创建 Prometheus 服务器。

   - 将 *prometheus-chart-name* 替换为您的 Prometheus 版本名称。
   - 将 *prometheus-agent-namespace* 替换为 Prometheus 命名空间的名称。

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

查询 Prometheus 指标

现在，指标已被摄取到工作区，您可以对其进行查询。查询指标的常用方法是使用诸如 Grafana 之类的服务来查询指标。在本节中，您将了解如何使用 Amazon Managed Grafana 从 Amazon Managed Service for Prometheus 中查询指标。

> **Note**
>
> 要了解查询 Amazon Managed Service for Prometheus 指标的其他方法，或使用 Amazon Managed Service for Prometheus API，请参阅查询 Prometheus 指标。

您可以使用标准的 Prometheus 查询语言 PromQL 来执行查询。有关 PromQL 及其语法的更多信息，请参阅 Prometheus 文档中的 Querying Prometheus。

Amazon Managed Grafana 是一项针对开源 Grafana 的完全托管服务，可简化与开源、第三方 ISV 的连接， AWS 以及用于大规模可视化和分析数据源的服务。

Amazon Managed Grafana for Prometheus 支持使用 Amazon Managed Grafana 查询工作区中的指标。在 Amazon Managed Grafana 控制台中，您可以通过发现现有的 Amazon Managed Service for Prometheus 账户，将 Amazon Managed Service for Prometheus 工作区添加为数据来源。Amazon Managed Grafana 管理访问 Amazon Managed Service for Prometheus 所需的身份验证凭证的配置。有关从 Amazon Managed Grafana 创建与 Amazon Managed Service for Prometheus 的连接的详细说明，请参阅 Amazon Managed Grafana 用户指南中的说明。

您还可以在 Amazon Managed Grafana 中查看 Amazon Managed Service for Prometheus 警报。有关设置与警报集成的说明，请参阅将警报与 Amazon Managed Grafana 或开源 Grafana 集成。

> **Note**
>
> 如果您已将 Amazon Managed Grafana 工作区配置为使用私有 VPC，则必须将 Amazon Managed Service for Prometheus 工作区连接到同一 VPC。有关更多信息，请参阅 在私有 VPC 中连接到 Amazon Managed Grafana。

管理工作区

工作区是专用于存储和查询 Prometheus 指标的逻辑空间。工作区支持精细的访问控制，用于授权其管理，例如更新、列出、描述和删除，以及指标的摄取和查询等。您可以在账户的每个区域中有一个或多个工作区。

使用本部分中的过程创建和管理 Amazon Managed Service for Prometheus 工作区。

主题

创建工作区

请按照以下步骤创建 Amazon Managed Service for Prometheus 工作区。您可以选择使用 AWS CLI 或 Amazon Prometheus 托管服务控制台。

> **Note**
>
> 如果您运行的是 Amazon EKS 集群，也可以使用适用于 Kubernetes 的AWS 控制器创建新的工作空间。

要使用创建工作区 AWS CLI

1. 输入以下命令来创建工作区。此示例创建名为 `my-first-workspace` 的工作区，但如果需要，可以使用其他别名（或不使用别名）。工作区别名是友好名称，有助于您识别工作区。名称没有必要是唯一的。两个工作区可以具有相同的别名，但所有工作区都要有唯一的工作区 ID，这些 ID 由 Amazon Managed Service for Prometheus 生成。

   （可选）要使用您自己的 KMS 密钥对存储在工作空间中的数据进行加密，可以在要使用的 AWS KMS 密钥中包含`kmsKeyArn`参数。虽然适用于 Prometheus 的亚马逊托管服务不会向您收取使用客户托管密钥的费用，但可能会产生与来自的密钥相关的费用。 AWS Key Management Service 有关 Amazon Managed Service for Prometheus 工作区中数据加密，或者如何创建、管理和使用您自己的与客户托管密钥的更多信息，请参阅静态加密。

方括号（[]）中的参数是可选参数，不要在命令中包含方括号。

```
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]
```

此命令将返回以下数据：

- workspaceId 是此工作区的唯一 ID。记下此 ID。
- arn 是此工作区的 ARN。
- status 是工作区的当前状态。创建工作区之后，该状态很可能立即变为 CREATING。
- kmsKeyArn 是用于加密工作区数据的客户托管密钥（如果已提供）。

> **Note**
>
> 使用客户托管密钥创建的工作区不能使用 AWS 托管收集器进行摄取。
> 谨慎选择是使用客户托管密钥还是 AWS 自有密钥。使用客户托管密钥创建的工作区以后不能转换为使用 AWS 自有密钥（反之亦然）。

- tags 会列出工作区的标签（如果有）。

2. 如果您的 create-workspace 命令返回的状态为 CREATING，则可以输入以下命令来确定工作区何时准备就绪。*my-workspace-id*替换为create-workspace命令返回的值workspaceId。

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

当 describe-workspace 命令针对 status 返回 ACTIVE 时，工作区就可以使用了。

使用 Amazon Managed Service for Prometheus 控制台创建工作区

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 选择创建。
3. 在工作区别名中，输入新工作区的别名。

工作区别名是友好名称，有助于您识别工作区。名称没有必要是唯一的。两个工作区可以具有相同的别名，但所有工作区都要有唯一的工作区 ID，这些 ID 由 Amazon Managed Service for Prometheus 生成。

4. （可选）要使用您自己的 KMS 密钥对存储在工作空间中的数据进行加密，您可以选择“自定义加密设置”，然后选择要使用的 AWS KMS 密钥（或创建新的密钥）。您可以从下拉列表中选择账户中的密钥，也可以输入您有权访问的任何密钥的 ARN。虽然适用于 Prometheus 的亚马逊托管服务不会向您收取使用客户托管密钥的费用，但可能会产生与来自的密钥相关的费用。 AWS Key Management Service

   有关 Amazon Managed Service for Prometheus 工作区中数据加密，或者如何创建、管理和使用您自己的与客户托管密钥的更多信息，请参阅静态加密。

   > **Note**
   >
   > 使用客户托管密钥创建的工作区不能使用 AWS 托管收集器进行摄取。
   > 谨慎选择是使用客户托管密钥还是 AWS 自有密钥。使用客户托管密钥创建的工作区以后不能转换为使用 AWS 自有密钥（反之亦然）。

5. （可选）要将一个或多个标签添加到工作区，请选择添加新标签。然后，在键中，输入标签的名称。您可以在 Value (值) 中添加可选的标签值。

   要添加其他标签，请再次选择 Add new tag（添加新标签）。

6. 选择创建工作区。

   此时将会显示工作区详细信息页面。这将显示该工作区的状态、ARN、工作区 ID 和终端节点 URL 等信息，用于远程写入和查询。

   在工作区准备就绪之前，状态将返回正在创建。等到状态变为活动后再继续设置指标摄取。

   记下终端节点 - 远程写入 URL 和终端节点 - 查询 URL 中显示的 URL。当您将 Prometheus 服务器配置为将指标远程写入此工作区以及查询这些指标时，将需要这些 URL。

有关如何将指标摄取到工作区的信息，请参阅将 Prometheus 指标摄取到工作区。

编辑工作区

您可以编辑工作区来更改其别名。要使用 AWS CLI更改工作区别名，请输入以下命令。

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

使用 Amazon Managed Service for Prometheus 控制台编辑工作区

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在页面左上角，选择菜单图标，然后选择所有工作区。
3. 选择要编辑的工作区的工作区 ID，然后选择编辑。
4. 为工作区输入新的别名，然后选择保存。

查找工作区 ARN

您可以使用控制台或 AWS CLI查找 Amazon Managed Service for Prometheus 工作区的 ARN。

使用 Amazon Managed Service for Prometheus 控制台查找工作区 ARN

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在页面左上角，选择菜单图标，然后选择所有工作区。
3. 选择工作区的工作区 ID。

   工作区 ARN 会显示在 ARN 下方。

要使用 AWS CLI 来查找您的工作空间 ARN，请输入以下命令。

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

在结果中查找 `arn` 的值。

删除工作区

删除工作区会删除已提取到其中的数据。

> **Note**
>
> 删除适用于 Prometheus 的亚马逊托管服务工作区不会自动删除 AWS 任何正在抓取指标并将其发送到工作区的托管收集器。有关更多信息，请参阅 查找和删除抓取程序。

要删除工作区，请使用 AWS CLI

使用以下命令：

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

使用 Amazon Managed Service for Prometheus 控制台删除工作区

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在页面左上角，选择菜单图标，然后选择所有工作区。
3. 选择要删除工作区的工作区 ID，然后选择删除。
4. 在确认框中，输入 **delete**，然后选择删除。

将指标提取到您的工作空间

必须先将指标导入您的 Amazon Prometheus 托管服务工作区，然后才能查询或提醒这些指标。本部分介绍如何设置将指标摄取到工作区。

> **Note**
>
> 默认情况下，采集到工作空间的指标会存储 150 天，然后会自动删除。此长度由可调整的配额控制。

有两种方法可以将指标摄取到 Amazon Managed Service for Prometheus 工作区。

- 使用 AWS 托管收集器 — 适用于 Prometheus 的亚马逊托管服务提供了一个完全托管、无代理的抓取工具，可以自动从您的亚马逊 Elastic Kubernetes Service (Amazon EKS) 集群中抓取指标。抓取会自动从与 Prometheus 兼容的端点中提取指标。
- 使用客户托管收集器 - 您可以通过多种方式管理自己的收集器。最常用的两个收集器是安装你自己的 Prometheus 实例，在代理模式下运行，或者使用 Distro 来做。 AWS OpenTelemetry以下部分详细介绍了这些内容。

  收集器使用 Prometheus 远程写入功能，将指标发送到 Amazon Managed Service for Prometheus。您可以使用 Prometheus 远程写入功能在您自己的应用程序中，将指标直接发送到 Amazon Managed Service for Prometheus。有关直接使用远程写入的更多详细信息，请参阅 Prometheus 文档中的 remote_write。

主题

AWS 托管收集器

Amazon Managed Service for Prometheus 的常见使用案例是监控由 Amazon Elastic Kubernetes Service（Amazon EKS）管理的 Kubernetes 集群。Kubernetes 集群以及在 Amazon EKS 中运行的许多应用程序会自动导出其指标以供兼容 Prometheus 的抓取程序访问。

> **Note**
>
> 在 Kubernetes 环境中运行的许多技术和应用程序都提供与 Prometheus 兼容的指标。有关明确记录的导出器的列表，请参阅 Prometheus 文档中的 Exporters and integrations。

Amazon Managed Service for Prometheus 提供完全托管的无代理抓取程序或收集器，可自动发现和提取与 Prometheus 兼容的指标。您无需管理、安装、修补或维护代理/抓取程序。Amazon Managed Service for Prometheus 收集器为您的 Amazon EKS 集群提供可靠、稳定、高度可用、可自动扩展的指标集合。适用于 Prometheus 的亚马逊托管服务托管收集器可与亚马逊 EKS 集群配合使用，包括 EC2 和 Fargate。

Amazon Managed Service for Prometheus 收集器为在创建抓取程序时指定的每个子网创建一个弹性网络接口（ENI）。收集器通过这些 ENI 抓取指标，然后使用 `remote_write` 通过 VPC 端点将数据推送到 Amazon Managed Service for Prometheus 工作区。抓取的数据永远不会在公共互联网上传输。

以下主题提供了有关如何在您的 Amazon EKS 集群中使用 Amazon Managed Service for Prometheus 收集器以及所收集的指标的更多信息。

主题

使用 AWS 托管收集器

要使用 Amazon Managed Service for Prometheus 收集器，您必须创建一个抓取程序，用于发现和提取您的 Amazon EKS 集群中的指标。

- 您可以在创建 Amazon EKS 集群的过程中创建抓取程序。有关创建 Amazon EKS 集群（包括创建抓取程序）的更多信息，请参阅《Amazon EKS 用户指南》中的创建 Amazon EKS 集群。
- 您可以使用 AWS API 以编程方式创建自己的抓取工具，也可以使用. AWS CLI

> **Note**
>
> 使用客户托管密钥创建的 Amazon Prometheus 托管服务工作空间不能使用托管收集器进行摄取。 AWS

Amazon Managed Service for Prometheus 收集器会抓取与 Prometheus 兼容的指标。有关与 Prometheus 兼容的指标的更多信息，请参阅与 Prometheus 兼容的指标有哪些？。

以下主题介绍如何创建、管理和配置抓取程序。

主题

- 创建抓取程序
- 配置 Amazon EKS 集群
- 查找和删除抓取程序
- 抓取程序配置
- 排查抓取程序配置问题
- 抓取程序限制

创建抓取程序

Amazon Managed Service for Prometheus 收集器由一个抓取程序组成，用于发现和收集 Amazon EKS 集群中的指标。Amazon Managed Service for Prometheus 为您管理抓取程序，为您提供所需的可扩展性、安全性和可靠性，无需您自行管理任何实例、代理或抓取程序。

当您通过 Amazon EKS 控制台创建 Amazon EKS 集群时，系统会自动为您创建抓取程序。但是，在某些情况下，您可能需要自行创建抓取程序。例如，如果您想向现有 Amazon EKS 集群添加 AWS 托管收集器，或者想要更改现有收集器的配置。

您可以使用 AWS API 或. 创建抓取工具 AWS CLI。

创建您自己的抓取程序时有以下几个先决条件：

- 您必须创建了 Amazon EKS 集群。
- 必须将 Amazon EKS 集群的集群端点访问控制设置为包括私有访问。它可以包括私有和公有访问，但必须包括私有访问。

> **Note**
>
> 集群将通过其 Amazon 资源名称 (ARN) 与抓取工具相关联。如果您删除了一个集群，然后创建了一个同名的新集群，则 ARN 将重新用于新集群。因此，抓取器将尝试收集新集群的指标。您可以将抓取器与删除集群分开删除。

AWS API

使用 AWS API 创建抓取工具

使用 AWS API 通过 `CreateScraper` API 操作创建抓取程序。以下示例在 `us-west-2` 区域中创建抓取程序。您需要将工作空间 AWS 账户、安全和 Amazon EKS 集群信息替换为自己的 ID，并提供用于抓取工具的配置。

> **Note**
>
> 您必须在至少两个可用区中至少包括两个子网。

`scrapeConfiguration` 是一个采用 base64 编码的 Prometheus 配置 YAML 文件。您可以通过 `GetDefaultScraperConfiguration` API 操作下载通用配置。有关格式的更多信息`scrapeConfiguration`，请参阅抓取程序配置。

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

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

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

AWS CLI

要创建刮刀，请使用 AWS CLI

使用`create-scraper`命令创建带有的抓取器 AWS CLI。以下示例在 `us-west-2` 区域中创建抓取程序。您需要将工作空间 AWS 账户、安全和 Amazon EKS 集群信息替换为自己的 ID，并提供用于抓取工具的配置。

> **Note**
>
> 您必须在至少两个可用区中至少包括两个子网。

`scrape-configuration` 是一个采用 base64 编码的 Prometheus 配置 YAML 文件。您可以使用`get-default-scraper-configuration`命令下载通用配置。有关格式的更多信息`scrape-configuration`，请参阅抓取程序配置。

```
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'}"
```

以下是您可以与 AWS API 一起使用的抓取程序操作的完整列表：

- 使用 CreateScraperAPI 操作创建抓取工具。
- 使用 ListScrapersAPI 操作列出您现有的抓取器。
- 使用 DeleteScraperAPI 操作删除抓取工具。

- DescribeScraper通过 API 操作获取有关抓取器的更多详细信息。
- GetDefaultScraperConfiguration通过 API 操作获取抓取器的通用配置。

> **Note**
>
> 必须将您要抓取的 Amazon EKS 集群配置为允许 Amazon Managed Service for Prometheus 访问这些指标。下一个主题介绍如何配置集群。

创建抓取器时的常见错误

以下是尝试创建新抓取器时最常见的问题。

- 所需 AWS 资源不存在。指定的安全组、子网和 Amazon EKS 集群必须存在。
- IP 地址空间不足。在传递给 `CreateScraper` API 的每个子网中，您必须至少有一个可用的 IP 地址。

配置 Amazon EKS 集群

必须将 Amazon EKS 集群配置为允许抓取程序访问指标。此配置有两个选项：

- 使用 Amazon EKS 访问条目自动为亚马逊托管服务提供 Prometheus 收集者访问您的集群的权限。
- 手动配置您的 Amazon EKS 集群以进行托管指标抓取。

以下主题更详细地描述了其中的每一个。

将 Amazon EKS 配置为使用访问条目进行抓取器访问

使用 Amazon EKS 的访问条目是授予适用于 Prometheus 的亚马逊托管服务访问权限以从您的集群中抓取指标的最简单方法。

您要抓取的 Amazon EKS 集群必须配置为允许 API 身份验证。集群身份验证模式必须设置为`API`或`API_AND_CONFIG_MAP`。这可以在 Amazon EKS 控制台的集群详细信息的 “访问配置” 选项卡上查看。有关更多信息，请参阅亚马逊 EKS 用户指南中的允许 IAM 角色或用户访问您的 Amazon EKS 集群上的 Kubernetes 对象。

你可以在创建集群时或在创建集群之后创建抓取器：

- 创建集群时 — 您可以在通过 A mazon EKS 控制台创建 Amazon EKS 集群时配置此访问权限（按照说明在集群中创建抓取工具），并且将自动创建访问进入策略，从而允许适用于 Prometheus 的亚马逊托管服务访问集群指标。
- 在@@ 创建集群后添加 — 如果您的 Amazon EKS 集群已经存在，则将身份验证模式设置为API或API_AND_CONFIG_MAP，并且您通过适用于 Prometheus 的亚马逊托管服务 API 或 CLI 创建的任何抓取器都将自动为您创建正确的访问进入策略，并且抓取器将有权访问您的集群。

访问权限策略已创建

当您创建抓取器并让适用于 Prometheus 的亚马逊托管服务为您生成访问权限策略时，它会生成以下策略。有关访问条目的更多信息，请参阅 A mazon EKS 用户指南中的允许 IAM 角色或用户访问 Kubernetes。

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

```
            ],
            "resources": [
                "ingresses/status",
                "ingresses"
            ],
            "verbs": [
                "get",
                "list",
                "watch"
            ]
        },
        {
            "effect": "allow",
            "nonResourceURLs": [
                "/metrics"
            ],
            "verbs": [
                "get"
            ]
        }
    ]
}
```

手动配置 Amazon EKS 以获取抓取工具访问权限

如果您更喜欢使用`aws-auth ConfigMap`来控制对 kubernetes 集群的访问权限，您仍然可以让适用于 Prometheus 抓取器的亚马逊托管服务访问您的指标。以下步骤将使适用于 Prometheus 的亚马逊托管服务能够从您的亚马逊 EKS 集群中获取指标。

> **Note**
>
> 有关`ConfigMap`和访问条目的更多信息，请参阅 A mazon EKS 用户指南中的允许 IAM 角色或用户访问 Kubernetes。

此过程使用`kubectl`和 AWS CLI。有关安装 `kubectl` 的信息，请参阅《Amazon EKS 用户指南》中的安装 kubectl。

手动配置您的 Amazon EKS 集群以进行托管指标抓取

1. 使用以下文本创建名为 `clusterrole-binding.yml` 的文件：

```
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. 在集群中运行以下命令。

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

这将创建集群角色绑定和规则。此示例使用 aps-collector-role 作为角色名称，使用 aps-collector-user 作为用户名。

3. 以下命令为您提供有关 ID 为 *scraper-id* 的抓取程序的信息。这是您使用上一节中的命令创建的抓取程序。

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

4. 从 describe-scraper 的结果中找到 roleArn。其格式如下所示：

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

Amazon EKS 要求此 ARN 采用不同的格式。您必须调整返回的 ARN 的格式，以便在下一步中使用。对其进行编辑以匹配以下格式：

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

例如，此 ARN：

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

必须重新编写为：

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

5. 使用上一步中修改过的 `roleArn` 以及您的集群名称和区域，在集群中运行以下命令：

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

这允许抓取程序使用您在 `clusterrole-binding.yml` 文件中创建的角色和用户访问集群。

查找和删除抓取程序

您可以使用 AWS API 或列 AWS CLI 出您账户中的抓取器或将其删除。

> **Note**
>
> 请确保您使用的是 AWS CLI 或 SDK 的最新版本。最新版本为您提供最新的特性和功能以及安全更新。或者，也可以使用 AWS Cloudshell，它可以自动提供始终是 up-to-date 命令行体验。

要列出您账户中的所有抓取器，请使用 ListScrapersAPI 操作。

或者，使用 AWS CLI，拨打：

```
aws amp list-scrapers
```

ListScrapers 返回您账户中的所有抓取程序，例如：

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

要删除抓取器，请使用ListScrapers操作找到要删除的抓取器，然后使用该DeleteScraper操作将其删除。scraperId

或者，使用 AWS CLI，拨打：

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

抓取程序配置

您可以使用兼容 Prometheus 的抓取程序配置来控制抓取程序如何发现和收集指标。例如，您可以更改将指标发送到工作区的时间间隔。您还可以使用重新标记来动态重写指标的标签。抓取程序配置是一个 YAML 文件，是抓取程序定义的一部分。

创建新的抓取程序时，您需通过在 API 调用中提供 base64 编码的 YAML 文件来指定配置。您可以在 Amazon Managed Service for Prometheus API 中通过 GetDefaultScraperConfiguration 操作下载通用配置文件。

要修改抓取程序的配置，请删除抓取程序并使用新的配置重新创建它。

支持的配置

有关抓取器配置格式的信息，包括可能值的详细明细，请参阅 Prometheus 文档中的配置。全局配置选项和 <scrape_config> 选项描述了最常用的选项。

由于 Amazon EKS 是唯一支持的服务，因此唯一支持的服务发现配置 (<*_sd_config>) 是<kubernetes_sd_config>。

允许的配置部分的完整列表：

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

这些部分中的限制列在示例配置文件之后。

示例配置文件

以下是抓取间隔为 30 秒的 YAML 配置文件示例。

```
global:
   scrape_interval: 30s
   external_labels:
     clusterArn: apiserver-test-2
scrape_configs:
  - job_name: pod_exporter
    kubernetes_sd_configs:
      - role: pod
  - job_name: cadvisor
    scheme: https
    authorization:
      type: Bearer
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    kubernetes_sd_configs:
      - role: node
    relabel_configs:
      - action: labelmap
        regex: __meta_kubernetes_node_label_(.+)
      - replacement: kubernetes.default.svc:443
        target_label: __address__
      - source_labels: [__meta_kubernetes_node_name]
        regex: (.+)
        target_label: __metrics_path__
        replacement: /api/v1/nodes/$1/proxy/metrics/cadvisor
  # apiserver metrics
  - scheme: https
    authorization:
      type: Bearer
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    job_name: kubernetes-apiservers
    kubernetes_sd_configs:
    - role: endpoints
    relabel_configs:
    - action: keep
      regex: default;kubernetes;https
      source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_service_name
      - __meta_kubernetes_endpoint_port_name
  # kube proxy metrics
  - job_name: kube-proxy
    honor_labels: true
```

```
    kubernetes_sd_configs:
    - role: pod
    relabel_configs:
    - action: keep
      source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_pod_name
      separator: '/'
      regex: 'kube-system/kube-proxy.+'
    - source_labels:
      - __address__
      action: replace
      target_label: __address__
      regex: (.+?)(\\:\\d+)?
      replacement: $1:10249
```

以下是 AWS 托管收集器特有的限制：

- 抓取间隔：抓取程序配置无法将抓取间隔指定为小于 30 秒。
- 目标：必须将 `static_config` 中的目标指定为 IP 地址。
- 授权-如果不需要授权，则省略。如果需要，则授权必须是`Bearer`，并且必须指向文件`/var/run/secrets/kubernetes.io/serviceaccount/token`。换句话说，如果使用，授权部分必须如下所示：

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

  > **Note**
  >
  > `type: Bearer`是默认值，因此可以省略。

排查抓取程序配置问题

Amazon Managed Service for Prometheus 收集器可自动发现和抓取指标。但是，当您在 Amazon Managed Service for Prometheus 工作区中看不到预期的指标时，如何排查问题呢？

up 指标是一种有用的工具。对于 Amazon Managed Service for Prometheus 收集器发现的每个端点，它都会自动提供此指标。此指标有三种状态，可以帮助您对收集器内部发生的问题进行排查。

- up 不存在 - 如果某个端点没有 up 指标，则表示收集器找不到该端点。

  如果您确定端点存在，则可能需要调整抓取配置。发现 `relabel_config` 可能需要调整，或者用于发现的 `role` 可能存在问题。

- up 存在，但始终为 0 - 如果 up 存在，但为 0，则表示收集器能够发现端点，但找不到任何与 Prometheus 兼容的指标。

  在这种情况下，您可以尝试直接对端点使用 `curl` 命令。您可以验证您的详细信息是否正确，例如您正在使用的协议（`http`或`https`）、端点或端口。您还可以检查端点是否使用有效的响应进行响`200`应，并遵循 Prometheus 格式。最后，响应的主体不能大于允许的最大大小。（有关 AWS 托管收集器的限制，请参阅以下部分。）

- up 存在且大于 0 - 如果 up 存在且大于 0，则表示指标将发送到 Amazon Managed Service for Prometheus。

  检查您在 Amazon Managed Service for Prometheus（或您的备用控制面板，例如 Amazon Managed Grafana）中查找的指标是否正确。您可以再次使用 curl 来检查 `/metrics` 端点中的预期数据。还需检查您是否没有超过其他限制，例如每个抓取程序的端点数量。您可以使用，通过检查指标计数来检查正在抓取的`up`指标端点的数量。`count(up)`

抓取程序限制

Amazon Managed Service for Prometheus 提供的完全托管的抓取程序几乎没有限制。

- 区域：您的 EKS 集群、托管抓取程序和 Amazon Managed Service for Prometheus 工作区必须全部位于同一个 AWS 区域。
- 账户：您的 EKS 集群、托管抓取程序和 Amazon Managed Service for Prometheus 工作区必须全部位于同一个 AWS 账户。
- 收集器：每个区域每个账户最多可以有 10 个 Amazon Managed Service for Prometheus 抓取程序。

> **Note**
>
> 您可以通过请求增加配额，请求提高此限额。

- 指标响应：来自任何一个 `/metrics` 端点请求的响应正文不能超过 50 兆字节（MB）。
- 每个抓取程序的端点：一个抓取程序最多可以抓取 3 万个 `/metrics` 端点。
- 抓取间隔：抓取程序配置无法将抓取间隔指定为小于 30 秒。

与 Prometheus 兼容的指标有哪些？

要从您的应用程序和基础设施中抓取 Prometheus 指标，以便在 Amazon Managed Service for Prometheus 中使用，他们必须从与 Prometheus 兼容的 `/metrics` 端点中检测和公开与 Prometheus 兼容的指标。您可以实施自己的指标，但不必这样做。Kubernetes（包括 Amazon EKS）及许多其他库和服务直接实施这些指标。

将 Amazon EKS 中的指标导出到与 Prometheus 兼容的端点时，您可以让 Amazon Managed Service for Prometheus 收集器自动抓取这些指标。

有关更多信息，请参阅以下主题：

- 有关将指标导出为 Prometheus 指标的现有库和服务的更多信息，请参阅 Prometheus 文档中的 Exporters and integrations。
- 有关从自己的代码中导出 Prometheus 兼容指标的更多信息，请参阅 Prometheus 文档中的 Writing exporters。
- 有关如何设置 Amazon Managed Service for Prometheus 收集器以自动从 Amazon EKS 集群中抓取指标的更多信息，请参阅使用 AWS 托管收集器。

客户托管收集器

本节包含以下相关信息：通过设置您自己的收集器来使用 Prometheus 远程写入将指标发送到 Amazon Managed Service for Prometheus，从而摄取数据。

当您使用自己的收集器向 Amazon Managed Service for Prometheus 发送指标时，您有责任保护自己的指标并确保摄取过程满足您的可用性需求。

大多数客户托管收集器都使用以下一种工具：

- AWS Distro for OpenTelemetry (ADOT) — ADOT 是一个完全受支持、安全、可用于生产的开源发行版，它为代理提供了收集指标 OpenTelemetry 的功能。您可以使用 ADOT 收集指标并将其发送到 Amazon Managed Service for Prometheus 工作区。有关 ADOT Collector 的更多信息，请参阅AWS 发行版。 OpenTelemetry
- Prometheus 代理：您可以设置自己的开源 Prometheus 服务器实例（作为代理运行），以收集指标并将其转发到 Amazon Managed Service for Prometheus 工作区。

以下主题介绍如何使用这两种工具，并包括有关设置您自己收集器的一般信息。

主题

保护指标的摄取

Amazon Managed Service for Prometheus 提供了帮助您保护指标摄取的方法。

搭 AWS PrivateLink 配适用于 Prometheus 的亚马逊托管服务

将指标摄取到 Amazon Prometheus 托管服务的网络流量可以通过公共互联网终端节点完成，也可以通过 VPC 终端节点通过。 AWS PrivateLink使用 AWS PrivateLink 可确保来自您 VPC 的网络流量在 AWS 网络中受到保护，而无需通过公共 Internet 完成。要为适用于 Prometheus 的亚马逊托管服务创建 VP AWS PrivateLink C 终端节点，请参阅。将 Amazon Managed Service for Prometheus 与接口 VPC 终端节点结合使用

身份验证和授权

AWS 身份和访问管理 (IAM) Access Management 是一项网络服务，可帮助您安全地控制对资源的访问 AWS 。可以使用 IAM 来控制谁通过了身份验证（准许登录）并获得授权（具有z权限）来使用资源。Amazon Managed Service for Prometheus 与 IAM 集成，可帮助您保护数据安全。设置 Amazon Managed Service for Prometheus 时，您需要创建一些 IAM 角色，使其能够从 Prometheus 服务器摄取指标，并让 Grafana 服务器查询存储在您 Amazon Managed Service for Prometheus 工作区中的指标。有关 IAM 的更多信息，请参阅什么是 IAM？。

另一项可以帮助您为 Prometheus 设置亚马逊托管服务的 AWS 安全功能是 AWS 签名版本 4 签名流程 (Sigv4)。AWS 签名版本 4 是向 HTTP 发送的 AWS 请求添加身份验证信息的过程。为了安全起见，对的大多数请求都 AWS 必须使用访问密钥进行签名，访问密钥由访问密钥 ID 和私有访问密钥组成。这两个密钥通常称为您的安全凭证。有关 SigV4 的更多信息，请参阅签名版本 4 签名流程。

使用 AWS Distro OpenTelemetry 作为收藏家

以下主题描述了将 AWS Distro 设置 OpenTelemetry为指标收集器的不同方法。

主题

- 在亚马逊 Elastic Kubernetes Service 集群上使用 AWS Distro 为开放遥测设置指标提取
- 使用适用于开放遥测的 AWS Distro 设置从 Amazon ECS 获取指标
- 使用远程写入设置从 Amazon EC2 实例摄取指标

在亚马逊 Elastic Kubernetes Service 集群上使用 AWS Distro 为开放遥测设置指标提取

本节介绍如何将 AWS Distro for OpenTelemetry (ADOT) Collector 配置为从装有 Prometheus 的应用程序中抓取，并将指标发送到适用于 Prometheus 的亚马逊托管服务。有关 ADOT Collector 的更多信息，请参阅AWS 发行版。 OpenTelemetry

使用 ADOT 收集 Prometheus 指标涉及三个 OpenTelemetry 组成部分：Prometheus 接收器、Prometheus 远程写入导出器和 Sigv4 身份验证扩展。

您可以使用现有的 Prometheus 配置来配置 Prometheus Receiver，以执行服务发现和指标抓取。Prometheus Receiver 以 Prometheus 展览格式抓取指标。您要抓取的任何应用程序或终端节点都应使用 Prometheus 客户端库进行配置。Prometheus Receiver 支持 Prometheus 文档配置中描述的全套 Prometheus 抓取和重新标记配置。您可以将这些配置直接粘贴到 ADOT 收集器配置中。

Prometheus Remote Write Exporter 使用 `remote_write` 终端节点将抓取的指标发送到您的管理门户工作区。导出数据的 HTTP 请求将使用 AWS Sigv4（安全身份验证 AWS 协议）和 Sigv4 身份验证扩展插件进行签名。有关更多信息，请参阅签名版本 4 签名流程。

收集器会自动发现 Amazon EKS 上的 Prometheus 指标终端节点，并使用 <kubernetes_sd_config> 中发现的配置。

以下演示是在运行 Amazon Elastic Kubernetes Service 或自行管理 Kubernetes 的集群上进行此配置的示例。要执行这些步骤，您必须拥有来自默认 AWS 凭证链中任何潜在选项的 AWS 证书。有关更多信息，请参阅配置 AWS SDK for Go。此演示使用了一个用于流程集成测试的示例应用程序。该示例应用程序在 `/metrics` 端点处公开指标，就像 Prometheus 客户端库一样。

先决条件

在开始以下摄取设置步骤之前，您必须为服务账户和信任策略设置 IAM 角色。

为服务账户和信任策略设置 IAM 角色

1. 按照设置服务角色从 Amazon EKS 集群中摄取指标中的步骤为服务账户创建 IAM 角色。

   ADOT 收集器将在抓取和导出指标时使用此角色。

2. 接下来，编辑信任策略。通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
3. 在左侧导航窗格中，选择角色并找到您在步骤 1 中创建的。amp-iamproxy-ingest-role
4. 选择信任关系选项卡，然后选择编辑信任关系。
5. 在信任关系策略 JSON 中，将 aws-amp 替换为 adot-col，然后选择更新信任策略。最终的信任策略应如下所示：

```
{
  "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. 选择权限选项卡，并确保将以下权限策略附加到该角色。

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

```
}
```

启用 Prometheus 指标收集

> **Note**
>
> 在 Amazon EKS 中创建命名空间时，默认情况下，`alertmanager` 和 Node Exporter 处于禁用状态。

在 Amazon EKS 或 Kubernetes 集群上启用 Prometheus 收集

1. 从存储库中分叉并克隆示例应用程序，网址为aws-otel-community。

   然后，运行以下命令。

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

2. 将此映像推送到注册表，例如 Amazon ECR 或 DockerHub。
3. 通过复制此 Kubernetes 配置并应用，在集群中部署示例应用程序。通过在 `prometheus-sample-app.yaml` 文件中替换 `{{PUBLIC_SAMPLE_APP_IMAGE}}`，将映像更改为刚才推送的映像。

   ```
   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. 输入以下命令以验证示例应用程序是否已启动。在命令的输出中，您将在 `NAME` 列中看到 `prometheus-sample-app`。

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

5. 启动 ADOT 收集器的默认实例。为此，请先输入以下命令来提取 ADOT 收集器的 Kubernetes 配置。

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

然后编辑模板文件，用您 Amazon Managed Service for Prometheus 工作区的 remote_write 终端节点替换 YOUR_ENDPOINT，并用您的区域替换 YOUR_REGION。查看工作区详细信息时，请使用 Amazon Managed Service for Prometheus 控制台中显示的 remote_write 终端节点。

你还需要YOUR_ACCOUNT_ID在 Kubernetes 配置的服务账户部分更改为你的 AWS 账户 ID。

在本示例中，ADOT 收集器配置使用注释（`scrape=true`）来告知要抓取哪些目标终端节点。如此，ADOT 收集器便可以将示例应用程序终端节点与您集群中的 kube-system 终端节点区分开来。如果您想抓取其它示例应用程序，则可以将其从重新标记配置中删除。

6. 输入以下命令以部署 ADOT 收集器。

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

7. 输入以下命令以验证 ADOT 收集器是否已启动。在 NAMESPACE 列中查找 `adot-col`。

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

8. 使用日志导出器验证管道是否正常运行。我们的示例模板已经与日志导出器集成。输入以下命令。

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

从示例应用程序中抓取的一些指标如下所示：

```
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. 要测试 Amazon Managed Service for Prometheus 是否已收到这些指标，请使用 `awscurl`。此工具允许您通过 AWS Sigv4 身份验证通过命令行发送 HTTP 请求，因此您必须在本地设置 AWS 凭证，并具有从亚马逊托管服务查询 Prometheus 的正确权限。有关安装的说明，请参阅 awscurl。`awscurl`

   在以下命令中，将 `AMP_REGION` 和 `AMP_ENDPOINT`替换为您 Amazon Managed Service for Prometheus 工作区的信息。

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

   如果您收到指标响应，则表示您的管道设置已成功，并且该指标已成功从示例应用程序传播到 Amazon Managed Service for Prometheus。

清理

要清理此演示，请输入以下命令。

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

高级配置

Prometheus Receiver 支持 Prometheus 文档配置中描述的全套 Prometheus 抓取和重新标记配置。您可以将这些配置直接粘贴到 ADOT 收集器配置中。

Prometheus Receiver 的配置包括您的服务发现、抓取配置和重新标记配置。接收方配置如下所示。

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

下面是一个配置示例：

```
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
```

如果您已有 Prometheus 配置，则必须将 $ 字符替换为 $$，以避免将值替换为环境变量。*这对于 relabel_configurations 的替换值尤其重要。例如，如果您从以下 relabel_configuration 开始：

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

它将变成以下内容：

```
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 和 Sigv4 Authentication Extension

Prometheus Remote Write Exporter 和 Sigv4 Authentication Extension 的配置比 Prometheus Receiver 简单。在管道的这个阶段，指标已经被摄取完毕，我们准备将这些数据导出到 Amazon Managed Service for Prometheus。对可与 Amazon Managed Service for Prometheus 进行通信的成功配置的最低要求如以下示例所示。

```
extensions:
```

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

此配置使用默认 AWS 凭证链中的 AWS 凭据发送由 AWS Sigv4 签名的 HTTPS 请求。有关更多信息，请参阅配置 AWS SDK for Go。必须将服务指定为 `aps`。

无论采用何种部署方法，ADOT 收集器都必须有权访问默认 AWS 凭证链中列出的选项之一。Sigv4 身份验证扩展依赖于 AWS SDK for Go 并使用它来获取凭据和进行身份验证。您必须确保这些凭证对于 Amazon Managed Service for Prometheus 具有远程写入权限。

使用适用于开放遥测的 AWS Distro 设置从 Amazon ECS 获取指标

本节介绍如何使用开放遥测发行版 (ADOT) 从亚马逊弹性容器服务 (Amazon ECS) 收集指标，并使用开放遥测发行版 (ADOT) 将其采集到适用于普罗米修斯的亚马逊托 AWS 管服务中。它还描述了如何在 Amazon Managed Grafana 中可视化您的指标。

先决条件

> **Important**
>
> 在开始之前，您必须在具有默认设置的 AWS Fargate 集群上拥有一个 Amazon ECS 环境、一个 Amazon Managed Service for Prometheus 工作区以及一个 Amazon Managed Grafana 工作区。我们假设您熟悉容器工作负载、Amazon Managed Service for Prometheus 和 Amazon Managed Grafana。

有关更多信息，请参阅以下链接：

- 有关如何在具有默认设置的 Fargate 集群上创建 Amazon ECS 环境的信息，请参阅《Amazon ECS 开发人员指南》中的创建集群。
- 有关如何创建 Amazon Managed Service for Prometheus 工作区的信息，请参阅《Amazon Managed Service for Prometheus 用户指南》中的创建工作区。
- 有关如何创建 Amazon Managed Grafana 工作区的信息，请参阅《Amazon Managed Grafana 用户指南》中的创建工作区。

定义自定义 ADOT 收集器容器映像

使用以下配置文件作为模板来定义您自己的 ADOT 收集器容器映像。将 *my-remote-URL* 和 *my-region* 替换为您的 `endpoint` 和 `region` 值。将配置保存在名为 adot-config.yaml 的文件中。

> **Note**
>
> 此配置使用 `sigv4auth` 扩展验证对 Amazon Managed Service for Prometheus 的调用。有关配置的更多信息`sigv4auth`，请参阅 Authenticator-Sigv4 on。 GitHub

```
receivers:
  prometheus:
    config:
      global:
        scrape_interval: 15s
        scrape_timeout: 10s
      scrape_configs:
      - job_name: "prometheus"
        static_configs:
        - targets: [ 0.0.0.0:9090 ]
  awsecscontainermetrics:
    collection_interval: 10s
processors:
  filter:
    metrics:
      include:
        match_type: strict
        metric_names:
          - ecs.task.memory.utilized
          - ecs.task.memory.reserved
          - ecs.task.cpu.utilized
          - ecs.task.cpu.reserved
          - ecs.task.network.rate.rx
          - ecs.task.network.rate.tx
          - ecs.task.storage.read_bytes
          - ecs.task.storage.write_bytes
exporters:
  prometheusremotewrite:
    endpoint: my-remote-URL
    auth:
      authenticator: sigv4auth
```

```
  logging:
    loglevel: info
extensions:
  health_check:
  pprof:
    endpoint: :1888
  zpages:
    endpoint: :55679
  sigv4auth:
    region: my-region
    service: aps
service:
  extensions: [pprof, zpages, health_check, sigv4auth]
  pipelines:
    metrics:
      receivers: [prometheus]
      exporters: [logging, prometheusremotewrite]
    metrics/ecs:
      receivers: [awsecscontainermetrics]
      processors: [filter]
      exporters: [logging, prometheusremotewrite]
```

将您的 ADOT 收集器容器映像推送到 Amazon ECR 存储库

使用 Dockerfile 创建容器映像，然后将其推送到 Amazon Elastic Container Registry（ECR）存储库。

1. 构建 Dockerfile 以将您的容器映像复制并添加到 OTEL Docker 映像中。

```
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. 创建 Amazon ECR 存储库。

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

3. 创建容器映像。

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

> **Note**
>
> 这会假设您在运行容器的环境中构建容器。否则，您可能需要在构建映像时使用 `--platform` 参数。

4. 登录 Amazon ECR 存储库。将 *my-region* 替换为您的 `region` 值。

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

5. 推送您的容器映像。

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

创建 Amazon ECS 任务定义以抓取 Amazon Managed Service for Prometheus

创建 Amazon ECS 任务定义以抓取 Amazon Managed Service for Prometheus。您的任务定义应包括一个名为 `adot-collector` 的容器和一个名为 `prometheus` 的容器。`prometheus` 生成指标，`adot-collector` 抓取 `prometheus`。

> **Note**
>
> Amazon Managed Service for Prometheus 作为一项服务运行，从容器中收集指标。在本例中，容器以代理模式在本地运行 Prometheus，将本地指标发送到 Amazon Managed Service for Prometheus。

示例：任务定义

以下是任务定义具体形式的示例。您可以使用此示例作为模板来创建自己的任务定义。将 `adot-collector` 的 `image` 值替换为您的存储库 URL 和映像标签（`$COLLECTOR_REPOSITORY:ecs`）。将 `adot-collector` 和 `prometheus` 的 `region` 值替换为您的 `region` 值。

```
{
  "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"
}
```

将 AWS 托管式策略 **AmazonPrometheusRemoteWriteAccess** 附加到您任务的 IAM 角色

要将抓取的指标发送到适用于 Prometheus 的亚马逊托管服务，您的 Amazon ECS 任务必须具有正确的权限才能 AWS 为您调用 API 操作。您必须为任务创建 IAM 角色和并将 AmazonPrometheusRemoteWriteAccess 策略附加至其中。有关创建该角色并附加策略的更多信息，请参阅为任务创建 IAM 角色和策略。

在您将 AmazonPrometheusRemoteWriteAccess 附加到您的 IAM 角色并使用该角色执行任务后，Amazon ECS 可以将您抓取的指标发送到 Amazon Managed Service for Prometheus。

在 Amazon Managed Grafana 中可视化您的指标

> **Important**
>
> 在开始之前，您必须在 Amazon ECS 任务定义上运行 Fargate 任务。否则，Amazon Managed Service for Prometheus 将无法使用您的指标。

1. 在 Amazon Managed Grafana 工作空间的导航窗格中，选择图标下方的数据源。 AWS
2. 在数据来源选项卡上的服务中，选择 Amazon Managed Service for Prometheus，然后选择您的默认区域。
3. 选择添加数据来源。
4. 使用 `ecs` 和 `prometheus` 前缀查询和查看您的指标。

使用远程写入设置从 Amazon EC2 实例摄取指标

本部分介绍如何运行在 Amazon Elastic Compute Cloud（Amazon EC2）实例中运行远程写入的 Prometheus 服务器。它解释了如何从用 Go 编写的演示应用程序中收集指标，并将这些指标发送到 Amazon Managed Service for Prometheus 工作区。

先决条件

> **Important**
>
> 在开始之前，您必须已经安装了 Prometheus v2.26 或更高版本。我们假设您熟悉 Prometheus、Amazon EC2 和 Amazon Managed Service for Prometheus。有关如何安装 Prometheus 的信息，请参阅 Prometheus 网站上的开始使用。

如果您不熟悉 Amazon EC2 或 Amazon Managed Service for Prometheus，我们建议您先阅读以下部分：

- 什么是 Amazon Elastic Compute Cloud？
- 什么是 Amazon Managed Service for Prometheus？

为 Amazon EC2 创建 IAM 角色

要流式传输指标，您必须先使用 AWS 托管策略创建 IAM 角色AmazonPrometheusRemoteWriteAccess。然后，您可以使用该角色启动实例，并将指标流式传输到您的 Amazon Managed Service for Prometheus 工作区。

1. 通过 https://console.aws.amazon.com/iam/ 打开 IAM 控制台。
2. 从导航窗格中选择角色，然后选择创建角色。
3. 对于信任实体的类型，选择 AWS service (亚马逊云科技服务)。对于使用案例，选择 EC2。选择下一步：权限。
4. 在搜索栏中输入 AmazonPrometheusRemoteWriteAccess。在 “策略名称” 中 AmazonPrometheusRemoteWriteAccess，选择，然后选择 “附加策略”。选择下一步: 标签。
5. （可选）为您的 IAM 角色创建 IAM 标签。选择 下一步: 审核。
6. 输入角色的名称。选择 创建策略。

启动 Amazon EC2 实例

要启动 Amazon EC2 实例，请按照《适用于 Linux 实例的 Amazon Elastic Compute Cloud 用户指南》中启动实例的说明进行操作。

运行演示应用程序

创建 IAM 角色并使用该角色启动 EC2 实例后，您可以运行演示应用程序来查看其运行情况。

运行演示应用程序并测试指标

1. 使用以下模板创建名为 `main.go` 的 Go 文件。

```
package main

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

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

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

2. 运行以下命令以安装相应的依赖项。

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

3. 运行演示应用程序。

```
go run main.go
```

演示应用程序应在端口 8000 上运行，并显示所有公开的 Prometheus 指标。以下是这些指标的示例。

```
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
```

创建 Amazon Managed Service for Prometheus 工作区

要创建 Amazon Managed Service for Prometheus 工作区，请按照创建工作区中的说明进行操作。

运行 Prometheus 服务器

1. 使用以下示例 YAML 文件作为模板来创建名为 `prometheus.yaml` 的新文件。对于`url`，将 *my-region* 替换为您的区域值以及*my-workspace-id*适用于 Prometheus 的亚马逊托管服务为您生成的工作空间 ID。对于 `region`，将 *my-region* 替换为您的区域值。

   示例：YAML 文件

   ```
   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. 运行 Prometheus 服务器，将演示应用程序的指标发送到 Amazon Managed Service for Prometheus 工作区。

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

Prometheus 服务器现在应该将演示应用程序的指标发送到 Amazon Managed Service for Prometheus 工作区。

使用 Prometheus 实例作为收集器

以下主题描述了设置在代理模式下运行的 Prometheus 实例作为指标收集器的不同方法。

> **Warning**
>
> 通过启用安全特征，避免将 Prometheus Scrape 端点暴露给公共 Internet。

如果您设置了多个 Prometheus 实例来监控同一组指标，并将其发送到单个 Amazon Managed Service for Prometheus 工作区以实现高可用性，则需要设置重复数据删除。如果您未按照步骤设置重复数据删除，则将向发送至 Amazon Managed Service for Prometheus 的所有数据样本（包括重复样本）收取费用。有关设置重复数据删除的说明，请参阅 对发送到 Amazon Managed Service for Prometheus 的高可用性指标进行重复数据删除。

主题

- 使用 Helm 设置从新 Prometheus 服务器进行摄取
- 在 EC2 上的 Kubernetes 中设置从现有 Prometheus 服务器进行摄取
- 在 Fargate 上的 Kubernetes 中设置从现有 Prometheus 服务器进行摄取

使用 Helm 设置从新 Prometheus 服务器进行摄取

本部分中的说明有助于您快速启动并运行 Amazon Managed Service for Prometheus。您在 Amazon EKS 集群中设置了一台新的 Prometheus 服务器，新服务器使用默认配置向 Amazon Managed Service for Prometheus 发送指标。本方法包含以下先决条件：

- 您必须有一个 Amazon EKS 集群，新的 Prometheus 服务器将从中收集指标。
- 您必须使用 Helm CLI 3.0 或更高版本
- 您必须使用 Linux 或 macOS 计算机来执行以下各部分中的步骤

步骤 1：添加新的 Helm 图表存储库

输入以下命令以添加新的 Helm 存储库。有关这些命令的更多信息，请参阅 Helm 存储库。

```
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
```

步骤 2：创建 Prometheus 命名空间

输入以下命令，为 Prometheus 服务器和其它监控组件创建 Prometheus 命名空间。将 *prometheus-namespace* 替换为想要的命名空间名称。

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

步骤 3：设置服务账户的 IAM 角色。

要使用我们记录的这种入门方法，您需要为运行 Prometheus 服务器的 Amazon EKS 集群中的服务账户使用 IAM 角色。

通过服务账户的 IAM 角色，您可以将 IAM 角色与 Kubernetes 服务账户关联。然后，此服务账户可向使用它的任意 Pod 中的容器提供 AWS 权限。有关更多信息，请参阅服务账户的 IAM 角色。

如果您尚未设置这些角色，请按照 设置服务角色从 Amazon EKS 集群中摄取指标 中的说明设置角色。该部分中的说明要求使用 `eksctl`。有关更多信息，请参阅 Amazon Elastic Kubernetes Service 入门 – `eksctl`。

> **Note**
>
> 如果您不在 EKS 上，或者仅使用访问密钥 AWS 和私有密钥访问适用于 Prometheus 的亚马逊托管服务，则无法使用基于的 Sigv4。`EKS-IAM-ROLE`

步骤 4：设置新服务器并开始摄取指标

要安装将指标发送到您 Amazon Managed Service for Prometheus 工作区的新 Prometheus 服务器，请按照以下步骤操作。

安装新的 Prometheus 服务器以将指标发送到 Amazon Managed Service for Prometheus 工作区

1. 使用文本编辑器创建名为 `my_prometheus_values_yaml` 的文件，其中包含以下内容。

   - 将 *IAM_PROXY_PROMETHEUS_ROLE_ARN ######### ARN*。amp-iamproxy-ingest-role设置服务角色从 Amazon EKS 集群中摄取指标
   - 将 *WORKSPACE_ID* 替换为您 Amazon Managed Service for Prometheus 工作区的 ID。
   - 将 *REGION* 替换为您 Amazon Managed Service for Prometheus 工作区的区域。

```
## 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. 输入以下命令以创建 Prometheus 服务器。

- *prometheus-chart-name*替换为您的 Prometheus 版本名称。
- 将 *prometheus-namespace* 替换为 Prometheus 命名空间的名称。

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

> **Note**
>
> 您可以通过多种方式自定义 `helm install` 命令。有关更多信息，请参阅 Helm 文档中的 Helm 安装。

在 EC2 上的 Kubernetes 中设置从现有 Prometheus 服务器进行摄取

Amazon Managed Service for Prometheus 支持从在 Amazon EKS 上运行的集群中以及在 Amazon EC2 上运行的自行管理 Kubernetes 集群中摄取指标。本部分中的详细说明适用于 Amazon EKS 集群中的 Prometheus 服务器。对于 Amazon EC2 上的自行管理 Kubernetes 集群，步骤相同，唯一的不同是，您需要在 Kubernetes 集群中自己为服务账户设置 OIDC 提供商和 IAM 角色。

本部分中的说明使用 Helm 作为 Kubernetes 软件包管理器。

主题

步骤 1：设置服务账户的 IAM 角色。

要使用我们记录的这种入门方法，您需要为运行 Prometheus 服务器的 Amazon EKS 集群中的服务账户使用 IAM 角色。此类角色又称服务角色。

借助服务角色，将 IAM 角色与 Kubernetes 服务账户关联。然后，该服务帐号可以为使用该服务帐号的任何 Pod 中的容器提供 AWS 权限。有关更多信息，请参阅服务账户的 IAM 角色。

如果您尚未设置这些角色，请按照 设置服务角色从 Amazon EKS 集群中摄取指标 中的说明设置角色。

步骤 2：使用 Helm 升级现有的 Prometheus 服务器

本部分中的说明包括设置远程写入和 sigv4 以进行身份验证，并授权 Prometheus 服务器远程写入到您的 Amazon Managed Service for Prometheus 工作区。

使用 Prometheus 版本 2.26.0 或更高版本

如果您使用的是带有 2.26.0 或更高版本 Prometheus 服务器映像的 Helm 图表，请按照以下步骤操作。

使用 Helm 图表从 Prometheus 服务器设置远程写入

1. 在您的 Helm 配置文件中创建一个新的远程写入部分：

   - `${IAM_PROXY_PROMETHEUS_ROLE_ARN}`替换为您在中创建amp-iamproxy-ingest-role的 ARN。步骤 1：设置服务账户的 IAM 角色。角色 ARN 的格式应为 `arn:aws:iam::`*`your account ID`*`:role/amp-iamproxy-ingest-role`。

- 将 ${WORKSPACE_ID} 替换为您的 Amazon Managed Service for Prometheus 工作区 ID。
- 将 ${REGION} 替换为 Amazon Managed Service for Prometheus 工作区的区域（如 us-west-2）。

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

2. 使用 Helm 更新您现有的 Prometheus 服务器配置：

- 将 prometheus-chart-name 替换为您的 Prometheus 版本名称。
- 将 prometheus-namespace 替换为安装了 Prometheus 服务器的 Kubernetes 命名空间。
- 将 my_prometheus_values_yaml 替换为 Helm 配置文件的路径。
- 将 current_helm_chart_version 替换为当前版本的 Prometheus 服务器 Helm 图表。您可以使用 helm list 命令找到当前的图表版本。

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

使用早期版本的 Prometheus

如果您使用的是低于 2.26.0 的 Prometheus 版本，请按照以下步骤操作。这些步骤使用边车方法，因为早期版本的 Prometheus 本身不 AWS 支持签名版本 4 签名过程 (Sigv4)。AWS

这些说明假设您使用 Helm 部署 Prometheus。

从 Prometheus 服务器设置远程写入

1. 在您的 Prometheus 服务器上，创建新的远程写入配置。首先，创建一个新的更新文件。我们将调用文件 `amp_ingest_override_values.yaml`。

   向 YAML 文件添加以下值。

```
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
```

   将 `${REGION}` 替换为 Amazon Managed Service for Prometheus 工作区的区域。

${SERVICE_ACCOUNT_IAM_INGEST_ROLE_ARN}替换为您在中创建amp-iamproxy-ingest-role的 ARN。步骤 1：设置服务账户的 IAM 角色。角色 ARN 的格式应为 arn:aws:iam::*your account ID*:role/amp-iamproxy-ingest-role。

将 ${WORKSPACE_ID} 替换为您的工作区 ID。

2. 升级您的 Prometheus Helm 图表。首先，输入以下命令，找到您的 Helm 图表名称。在此命令的输出中，查找名称包含 prometheus 的图表。

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

然后，输入以下命令。

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

*prometheus-helm-chart-name*替换为上一个命令中返回的 Prometheus 头盔图的名称。将 *prometheus-namespace* 替换为您的命名空间的名称。

下载 Helm 图表

如果您尚未在本地下载 Helm 图表，则可以使用以下命令下载。

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

在 Fargate 上的 Kubernetes 中设置从现有 Prometheus 服务器进行摄取

Amazon Managed Service for Prometheus 支持从在 Fargate 上运行的自行管理 Kubernetes 集群中的 Prometheus 服务器摄取指标。要从 Fargate 上运行的 Amazon EKS 集群中的 Prometheus 服务器摄取指标，请覆盖名为 amp_ingest_override_values.yaml 的配置文件中的默认配置，如下所示：

```
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
```

通过以下命令使用覆盖安装 Prometheus：

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

请注意，在 Helm 图表配置中，我们禁用了 Node Exporter 和 Alertmanager，并运行了 Prometheus 服务器部署。

您可以使用以下示例测试查询来验证安装。

```
$ 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
```

设置 Amazon Managed Service for Prometheus 以获取高可用性数据

当您将数据发送到 Amazon Managed Service for Prometheus 时，数据会自动在该区域的 AWS 可用区间复制，并由提供可扩展性、可用性和安全性的主机集群提供给您。您可能需要添加额外的高可用性故障安全功能，具体取决于您的特定设置。有两种常见的方法可以为您的设置增加高可用性安全性：

- 如果您有多个容器或实例具有相同数据，则可以将这些数据发送到 Amazon Managed Service for Prometheus，并自动删除重复数据。这有助于确保您的数据将发送到 Amazon Managed Service for Prometheus 工作区。

  有关对高可用性数据进行重复数据消除的更多信息，请参阅 对发送到 Amazon Managed Service for Prometheus 的高可用性指标进行重复数据删除。
- 如果您想确保即使 AWS 区域不可用的情况下也可以访问数据，则可以将指标发送到另一个区域的第二个工作区。

  有关将指标数据发送到多个工作区的更多信息，请参阅 跨区域可用性。

主题

对发送到 Amazon Managed Service for Prometheus 的高可用性指标进行重复数据删除

您可以将来自多个 Prometheus 代理（在代理模式下运行的 Prometheus 实例）的数据发送到 Amazon Managed Service for Prometheus 工作区。如果其中一些实例记录并发送相同的指标，则您的数据将具有更高的可用性（即使其中一个代理停止发送数据，Amazon Managed Service for Prometheus 工作区仍将接收来自另一个实例的数据）。但是，您希望 Amazon Managed Service for Prometheus 工作区自动删除重复的指标，这样您就不会多次看到这些指标，也不会多次对数据摄取和存储付费。

要让 Amazon Managed Service for Prometheus 自动删除来自多个 Prometheus 代理的重复数据，您需要为发送重复数据的代理组指定一个集群名称，并为每个实例指定一个副本名称。集群名称将实例标识为具有共享数据，副本名称允许 Amazon Managed Service for Prometheus 识别每个指标的来源。最终存储的指标包括集群标签，但不包括副本，因此这些指标似乎来自单一来源。

> **Note**
>
> 某些版本的 Kubernetes（1.28 和 1.29）可能会发布自己的带有标签的指标。`cluster`这可能会导致适用于 Prometheus 的亚马逊托管服务出现重复数据删除问题。有关更多信息，请参阅高可用性常见问题解答。

以下主题介绍如何发送数据并添加`cluster`和`__replica__`标签，以便适用于 Prometheus 的 Amazon 托管服务自动删除重复数据。

> **Important**
>
> 如果您未设置重复数据删除，则需要为发送到 Amazon Managed Service for Prometheus 的所有数据样本付费。这些数据样本包括重复的样本。

使用 Prometheus 将高可用性数据发送到 Amazon Managed Service for Prometheus

要使用 Prometheus 设置高可用性配置，您必须在高可用性组的所有实例上应用外部标签，以便 Amazon Managed Service for Prometheus 可以进行识别。使用 `cluster` 标签将 Prometheus 实例代理标识为高可用性组的一部分。使用 `__replica__` 标签分别标识组中的每个副本。要使重复数据删除功能起作用，您需要同时应用 `__replica__` 和 `cluster` 标签。

> **Note**
>
> `__replica__` 标签的格式为在单词 `replica` 前后使用两个下划线符号。

示例：代码片段

在以下代码片段中，`cluster` 标签标识 Prometheus 实例代理 `prom-team1`，`_replica_` 标签标识副本 `replica1` 和 `replica2`。

```
cluster: prom-team1
__replica__: replica1
```

```
cluster: prom-team1
```

```
__replica__: replica2
```

由于 Amazon Managed Service for Prometheus 存储带有这些标签的高可用性副本的数据样本，因此当样本被接受时，它会删除 `replica` 标签。这意味着您当前的序列只有 1:1 的序列映射，而不是每个副本一个序列。保留了 `cluster` 标签。

> **Note**
>
> 某些版本的 Kubernetes（1.28 和 1.29）可能会发布自己的带有标签的指标。`cluster`这可能会导致适用于 Prometheus 的亚马逊托管服务出现重复数据删除问题。有关更多信息，请参阅高可用性常见问题解答。

使用 Prometheus Operator 将高可用性数据发送到 Amazon Managed Service for Prometheus

要使用 Prometheus Operator 设置高可用性配置，您必须在高可用性组的所有实例上应用外部标签，以便 Amazon Managed Service for Prometheus 可以进行识别。您还必须在 Prometheus Operator Helm 图表上设置 `replicaExternalLabelName` 和 `externalLabels` 属性。

示例：YAML 标头

在以下 YAML 标头中，在 `externalLabel` 中添加了 `cluster` 以将 Prometheus 实例代理标识为高可用性组的一部分，并且 `replicaExternalLabels` 标识该组中的每个副本。

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

> **Note**
>
> 某些版本的 Kubernetes（1.28 和 1.29）可能会发布自己的带有标签的指标。`cluster`这可能会导致适用于 Prometheus 的亚马逊托管服务出现重复数据删除问题。有关更多信息，请参阅高可用性常见问题解答。

使用开放遥测发行版将高可用性数据发送到适用于 Pro AWS metheus 的 Amazon 托管服务

AWS 开放遥测发行版 (ADOT) 是该项目的安全且可随时投入生产的发行版。 OpenTelemetry ADOT 为您提供源 API、库和代理，因此您可以收集分布式跟踪和指标以进行应用程序监控。有关 ADOT 的信息，请参阅关于 Open Tel AWS emetry 发行版。

要将 ADOT 设置为高可用性配置，必须配置 ADOT 收集器容器镜像，并将外部标签`cluster`应用于 Prometheus `__replica__` AWS 远程写入导出器。此导出器通过 `remote_write` 终端节点将您抓取的指标发送到 Amazon Managed Service for Prometheus 工作区。在 Remote Write Exporter 上设置这些标签时，可以防止在冗余副本运行时保留重复的指标。有关 AWS Prometheus 远程写入导出器的更多信息，请参阅适用于 Prometheus 的亚马逊托管服务的 Prometheus 远程写入导出器入门。

> **Note**
>
> 某些版本的 Kubernetes（ 1.28 和 1.29 ）可能会发布自己的带有标签的指标。`cluster`这可能会导致适用于 Prometheus 的亚马逊托管服务出现重复数据删除问题。有关更多信息，请参阅高可用性常见问题解答。

使用 Prometheus 社区 Helm 图表向 Amazon Managed Service for Prometheus 发送高可用性数据

要使用 Prometheus 社区 Helm 图表设置高可用性配置，您必须在高可用性组的所有实例上应用外部标签，以便 Amazon Managed Service for Prometheus 可以进行识别。以下是如何将 `external_labels` 从 Prometheus 社区 Helm 图表中添加到 Prometheus 的单个实例的示例。

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

> **Note**
>
> 如果您想要多个副本，则必须使用不同的副本值多次部署图表，因为 Prometheus 社区 Helm 图表不允许您在直接从控制器组增加副本数量时动态设置副本值。如果您更偏好自动设置 `replica` 标签，请使用 prometheus-operator Helm 图表。

> **Note**
>
> 某些版本的 Kubernetes（1.28 和 1.29）可能会发布自己的带有标签的指标。`cluster`这可能会导致适用于 Prometheus 的亚马逊托管服务出现重复数据删除问题。有关更多信息，请参阅高可用性常见问题解答。

常见问题：高可用性配置

我是否应该在另一个标签中包含值 __replica__ 来跟踪采样点？

在高可用性设置中，Amazon Managed Service for Prometheus 通过在 Prometheus 实例集群中选出领导来确保数据样本不会重复。如果领导副本在 30 秒内停止发送数据样本，则 Amazon Managed Service for Prometheus 会自动将另一个 Prometheus 实例设置为领导副本，并从新领导那里摄取数据，包括任何丢失的数据。因此，答案是否定的，不建议这样做。 这样做可能会导致以下问题：

- 在选择新领导期间，在 PromQL 中查询 `count` 返回的值可能会高于预期值。
- 在选择新领导期间，`active series` 数增加了，达到了 `active series limits`。有关更多信息，请参阅 AMP 配额。

Kubernetes 似乎有自己的集群标签，并且没有对我的指标进行重复数据删除。如何修复此问题？

Kubernetes 1.28 中引入`apiserver_storage_size_bytes`了一个带有标签的新指标。`cluster`这可能会导致适用于 Prometheus 的 Amazon 托管服务出现重复数据删除问题，具体取决于标签。`cluster`在 Kubernetes 1.3 中，该标签被重命名为`storage-cluster_id`（在 1.28 和 1.29 的后续补丁中也将其重命名为）。如果您的集群发出带有`cluster`标签的此指标，则适用于 Prometheus 的亚马逊托管服务无法对关联的时间序列进行重复数据删除。我们建议您将您的 Kubernetes 集群升级到最新的补丁版本以避免出现此问题。或者，您也可以在`apiserver_storage_size_bytes`指标上重新`cluster`标记标签，然后再将其导入亚马逊 Prometheus 托管服务。

> **Note**
>
> 有关 Kubernetes 变更的更多详细信息，请参阅 Kubernetes 项目中的 apiserver_storage_size_bytes 指标将标签集群重命名为 storage_cluster_ id。 GitHub

跨区域可用性

要为您的数据添加跨区域可用性，您可以将指标发送到跨 AWS 区域的多个工作空间。Prometheus 支持多个写入器和跨区域写入。

以下示例说明如何设置在代理模式下运行的 Prometheus 服务器，以便使用 Helm 将指标发送到不同区域的两个工作区。

```
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]
```

查询 Prometheus 指标

现在，指标已被摄取到工作区，您可以对其进行查询。您可以使用诸如 Grafana 之类的服务来查询指标，也可以使用 Amazon Managed Service for Prometheus API。

您可以使用标准的 Prometheus 查询语言 PromQL 来执行查询。有关 PromQL 及其语法的更多信息，请参阅 Prometheus 文档中的 Querying Prometheus。

主题

保护您的指标查询

Amazon Managed Service for Prometheus 提供了帮助您保护指标查询的方法。

搭 AWS PrivateLink 配适用于 Prometheus 的亚马逊托管服务

在适用于 Prometheus 的 Amazon 托管服务中查询指标的网络流量可以通过公共互联网终端节点完成，也可以通过 VPC 终端节点通过。 AWS PrivateLink使用时 AWS PrivateLink，来自您的 VPC 的网络流量可以在 AWS 网络内得到保护，而无需通过公共互联网。要为适用于 Prometheus 的亚马逊托管服务创建 VP AWS PrivateLink C 终端节点，请参阅。将 Amazon Managed Service for Prometheus 与接口 VPC 终端节点结合使用

身份验证和授权

AWS Identity and Access Management 是一项 Web 服务，可帮助您安全地控制对 AWS 资源的访问。可以使用 IAM 来控制谁通过了身份验证（准许登录）并获得授权（具有z权限）来使用资源。Amazon Managed Service for Prometheus 与 IAM 集成，可帮助您保护数据安全。当设置 Amazon Managed Service for Prometheus 时，您需要创建一些 IAM 角色，让 Grafana 服务器能够查

询存储在 Amazon Managed Service for Prometheus 工作区中的指标。有关 IAM 的更多信息，请参阅什么是 IAM？。

另一项可以帮助您为 Prometheus 设置亚马逊托管服务的 AWS 安全功能是 AWS 签名版本 4 签名流程 (Sigv4)。AWS 签名版本 4 是向 HTTP 发送的 AWS 请求添加身份验证信息的过程。为了安全起见，对的大多数请求都 AWS 必须使用访问密钥进行签名，访问密钥由访问密钥 ID 和私有访问密钥组成。这两个密钥通常称为您的安全凭证。有关 SigV4 的更多信息，请参阅签名版本 4 签名流程。

设置 Amazon Managed Grafana 以与 Amazon Managed Service for Prometheus 配合使用

Amazon Managed Grafana 是一项针对开源 Grafana 的完全托管服务，可简化与开源、第三方 ISV 的连接， AWS 以及用于大规模可视化和分析数据源的服务。

Amazon Managed Grafana for Prometheus 支持使用 Amazon Managed Grafana 查询工作区中的指标。在 Amazon Managed Grafana 控制台中，您可以通过发现现有的 Amazon Managed Service for Prometheus 账户，将 Amazon Managed Service for Prometheus 工作区添加为数据来源。Amazon Managed Grafana 管理访问 Amazon Managed Service for Prometheus 所需的身份验证凭证的配置。有关从 Amazon Managed Grafana 创建与 Amazon Managed Service for Prometheus 的连接的详细说明，请参阅 Amazon Managed Grafana 用户指南中的说明。

您还可以在 Amazon Managed Grafana 中查看 Amazon Managed Service for Prometheus 警报。有关设置与警报集成的说明，请参阅将警报与 Amazon Managed Grafana 或开源 Grafana 集成。

在私有 VPC 中连接到 Amazon Managed Grafana

Amazon Managed Service for Prometheus 为 Amazon Managed Grafana 提供了一个服务终端节点，供其在查询指标和警报时连接。

您可以将 Amazon Managed Grafana 配置为使用私有 VPC（有关在 Grafana 中设置私有 VPC 的详细信息，请参阅《Amazon Managed Grafana 用户指南》中的连接 Amazon VPC）。根据设置，此 VPC 可能无法访问 Amazon Managed Service for Prometheus 服务终端节点。

要将 Amazon Managed Service for Prometheus 作为数据来源添加到配置为使用特定私有 VPC 的 Amazon Managed Grafana 工作区，您必须先通过创建 VPC 终端节点将 Amazon Managed Service for Prometheus 连接到同一 VPC。有关创建 VPC 终端节点的更多信息，请参阅 为 Amazon Managed Service for Prometheus 创建接口 VPC 终端节点。

设置 Grafana 开源或 Grafana Enterprise 以与 Amazon Managed Service for Prometheus 配合使用

Amazon Managed Service for Prometheus 支持使用 Grafana 7.3.5 及更高版本来查询工作区中的指标。版本 7.3.5 及更高版本包括对 AWS 签名版本 4 (Sigv4) 身份验证的支持。

有关使用 tar.gz 或 zip 文件设置独立 Grafana 的说明，请参阅 Grafana 文档中的 Install Grafana。如果您安装新的独立 Grafana，系统将提示您输入用户名和密码。默认值为 **admin/admin**。首次登录后，系统将提示您更改密码。有关更多信息，请参阅 Grafana 文档中的 Getting started with Grafana。

要检查 Grafana 版本，请输入以下命令。

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

要将 Grafana 设置为使用适用于 Prometheus 的亚马逊托管服务，您必须登录到具有AmazonPrometheusQueryAccess策略或、、和权限的账户。`aps:QueryMetrics` `aps:GetMetricMetadata` `aps:GetSeries` `aps:GetLabels`有关更多信息，请参阅 IAM 权限和策略。

设置 AWS SigV4

适用于 Prometheus 的亚马逊托管服务 AWS Identity and Access Management 与 (IAM) 合作，使用 IAM 凭证保护对 Prometheus API 的所有调用。默认情况下，Grafana 中的 Prometheus 数据来源假定 Prometheus 不需要身份验证。要让 Grafana 能够利用 Amazon Managed Service for Prometheus 身份验证和授权功能，您将需要在 Grafana 数据来源中启用 Sigv4 身份验证支持。当您使用自行管理的 Grafana 开源服务器或 Grafana 企业服务器时，请按照本页上的步骤进行操作。如果您使用的是 Amazon Managed Grafana，则 SIGv4 身份验证完全自动执行。有关 Amazon Managed Grafana 的更多信息，请参阅 What is Amazon Managed Grafana?

要在 Grafana 上启用 SigV4，请在 `AWS_SDK_LOAD_CONFIG` 和 `GF_AUTH_SIGV4_AUTH_ENABLED` 环境变量设置为 `true` 的情况下启动 Grafana。`GF_AUTH_SIGV4_AUTH_ENABLED` 环境变量将覆盖 Grafana 的默认配置以启用 Sigv4 支持。有关更多信息，请参阅 Grafana 文档中的 Configuration。

Linux

要在 Linux 上的独立 Grafana 服务器上启用 SigV4，请输入以下命令。

```
export AWS_SDK_LOAD_CONFIG=true
```

```
export GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

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

Windows

要使用 Windows 命令提示符在 Windows 的独立 Grafana 上启用 SigV4，请输入以下命令。

```
set AWS_SDK_LOAD_CONFIG=true
```

```
set GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

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

在 Grafana 中添加 Prometheus 数据来源

以下步骤说明了如何在 Grafana 中设置 Prometheus 数据来源，以便查询您的 Amazon Managed Service for Prometheus 指标。

在您的 Grafana 服务器中添加 Prometheus 数据来源

1. 打开 Grafana 控制台。
2. 在配置下，选择数据来源。
3. 选择添加数据来源。
4. 选择 Prometheus。
5. 对于 HTTP URL，请指定 Amazon Managed Service for Prometheus 控制台的工作区详情页面中显示的终端节点 - 查询 URL。
6. 在您刚才指定的 HTTP URL 中，删除附加到该 URL 的 `/api/v1/query` 字符串，因为 Prometheus 数据来源会自动附加该字符串。

   正确的 URL 应类似如下：https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-1234a5b6-78cd-901e-2fgh-3i45j6k178l9。

7. 在身份验证下，选择 Sigv4 身份验证的开关将其启用。
8. 您可以通过直接在 Grafana 中指定您的长期凭证来配置 Sigv4 授权，也可以使用默认的提供商链。直接指定您的长期凭证可以让您更快地启动，以下步骤首先给出了这些说明。在您更加熟悉将 Grafana 与 Amazon Managed Service for Prometheu 一起使用后，我们建议您使用默认的提供商链，因为它提供了更好的灵活性和安全性。有关设置默认提供商链的更多信息，请参阅指定凭证。

   - 要直接使用长期凭证，请执行以下操作：

     a. 在 Sigv4 身份验证详细信息下的身份验证提供商中选择访问和密钥。

     b. 在访问密钥 ID 中，输入您的 AWS 访问密钥 ID。

     c. 在秘密访问密钥中输入您的秘密访问密钥。

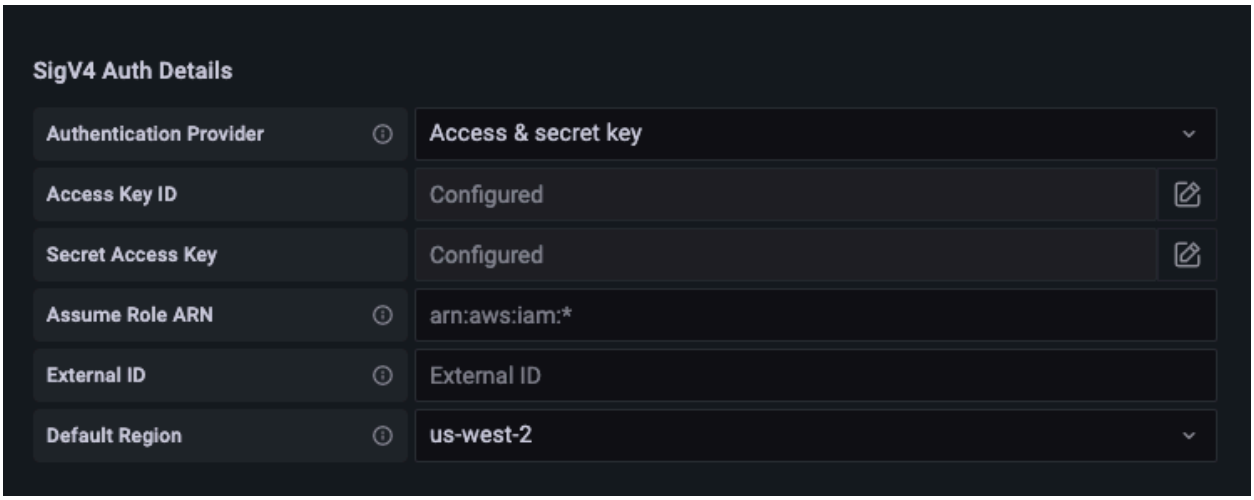
     d. 将担任角色 ARN 和外部 ID字段留空。

     e. 对于默认区域，选择 Amazon Managed Service for Prometheus 工作区的区域。此区域应与您在步骤 5 中列出的 URL 中包含的区域相匹配。

     f. 选择保存并测试。

        您应该看到以下消息：数据来源正在运行

        以下屏幕截图显示了访问密钥、密钥 Sigv4 身份验证详细信息设置。



   - 要改用默认的提供商链（建议在生产环境中使用），请执行以下操作：

     a. 在 Sigv4 身份验证详细信息下的身份验证提供商中选择 AWS SDK 默认。

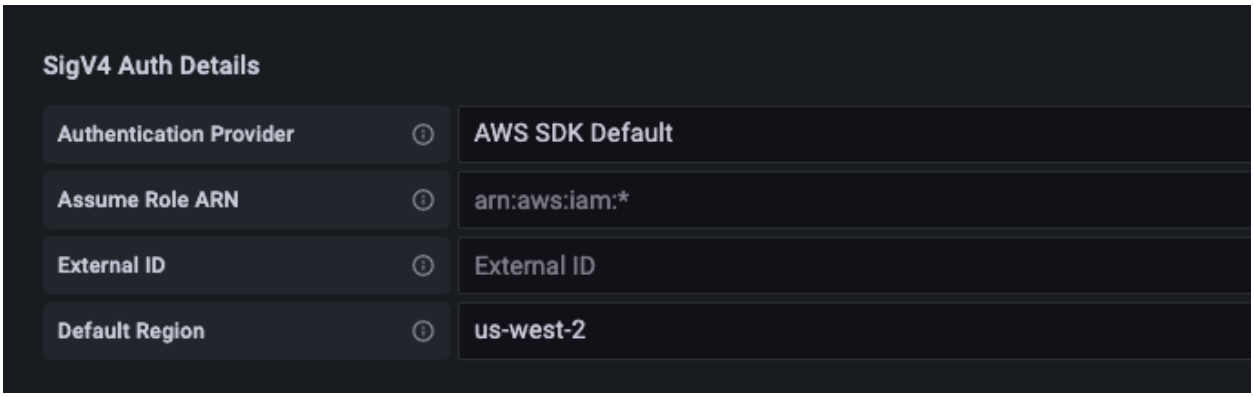
     b. 将担任角色 ARN 和外部 ID字段留空。

     c. 对于默认区域，选择 Amazon Managed Service for Prometheus 工作区的区域。此区域应与您在步骤 5 中列出的 URL 中包含的区域相匹配。

     d. 选择保存并测试。

您应该看到以下消息：数据来源正在运行

以下屏幕截图显示了 SDK 默认 SigV4 身份认证详细信息设置。



9. 针对新的数据来源测试 PromQL 查询：

   a. 选择探索。

   b. 运行示例 PromQL 查询，例如：

```
prometheus_tsdb_head_series
```

“保存并测试”不起作用时进行故障排除

在前面的步骤中，如果您在选择保存并测试时看到错误，请检查以下内容。

HTTP 错误未找到

确保 URL 中的工作区 ID 正确无误。

HTTP 错误禁止

此错误意味着凭证无效。请检查以下事项：

- 检查默认区域中指定的区域是否正确。
- 检查您的凭证是否有拼写错误。
- 请确保您使用的凭证具有该AmazonPrometheusQueryAccess政策。有关更多信息，请参阅 IAM 权限和策略。
- 确保您使用的凭证可以访问此 Amazon Managed Service for Prometheus 工作区。

HTTP 错误错误的网关

查看 Grafana 服务器日志以解决此错误。有关更多信息，请参阅 Grafana 文档中的 Troubleshooting。

如果您看到**`Error http: proxy error: NoCredentialProviders: no valid providers in chain`**，则默认凭证提供商链无法找到要使用的有效 AWS 凭证。确保您已按照指定凭证中所述设置了凭证。如果要使用共享配置，请确保将 `AWS_SDK_LOAD_CONFIG` 环境设置为 `true`。

使用在 Amazon EKS 集群中运行的 Grafana 进行查询

Amazon Managed Service for Prometheus 支持使用 Grafana 7.3.5 及更高版本来查询 Amazon Managed Service for Prometheus 工作区中的指标。版本 7.3.5 及更高版本包括对 AWS 签名版本 4 (Sigv4) 身份验证的支持。

要将 Grafana 设置为使用适用于 Prometheus 的亚马逊托管服务，您必须登录到具有AmazonPrometheusQueryAccess策略或、、和权限的账户。`aps:QueryMetrics` `aps:GetMetricMetadata` `aps:GetSeries` `aps:GetLabels`有关更多信息，请参阅 IAM 权限和策略。

设置 s AWS igV4

Grafana 添加了一项新功能来 AWS 支持签名版本 4 (Sigv4) 身份验证。有关更多信息，请参阅签名版本 4 签名流程。该功能默认在 Grafana 服务器上未启用。以下启用此功能的说明假设您使用 Helm 在 Kubernetes 集群上部署 Grafana。

在 Grafana 7.3.5 或更高版本的服务器上启用 SigV4

1. 创建一个新的更新文件来覆盖您的 Grafana 配置，并将其命名为 `amp_query_override_values.yaml`。
2. 在文件中输入以下内容，然后保存该文件。将*`## ID`* 替换为运行 Grafana 服务器的 AWS 账户 ID。

```
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
```

在该 YAML 文件内容中，amp-iamproxy-query-role 是您将在下一部分 设置服务账户的 IAM 角色 中创建的角色的名称。如果您已经创建了用于查询工作区的角色，则可以将此角色替换为自己的角色名称。

稍后您将在 使用 Helm 升级 Grafana 服务器 中使用此文件。

设置服务账户的 IAM 角色

如果您在 Amazon EKS 集群中使用 Grafana 服务器，我们建议您使用服务账户的 IAM 角色（也称为服务角色）进行访问控制。当你这样做是为了将 IAM 角色与 Kubernetes 服务账户关联时，该服务账号就可以为使用该服务账号的任何 Pod 中的容器提供 AWS 权限。有关更多信息，请参阅服务账户的 IAM 角色。

如果您尚未设置这些角色进行查询，请按照 设置服务账户的 IAM 角色以查询指标 中的说明设置角色。

然后，您需要在信任关系条件中添加 Grafana 服务账户。

在信任关系条件中添加 Grafana 服务账户

1. 在终端窗口中，确定 Grafana 服务器的命名空间和服务账户名称。例如，您可以使用以下命令：

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

2. 在 Amazon EKS 控制台中，为与 EKS 集群关联的服务账户打开 IAM 角色。
3. 选择编辑信任关系。
4. 更新条件以包含您在步骤 1 的命令输出中找到的 Grafana 命名空间和 Grafana 服务账户名称。示例如下：

```
{
  "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. 选择更新信任策略。

使用 Helm 升级 Grafana 服务器

此步骤将升级 Grafana 服务器以使用您在上一部分中添加到 amp_query_override_values.yaml 文件中的条目。

运行以下命令。有关 Grafana 的 Helm 图表的更多信息，请参阅 Grafana 社区 Kubernetes Helm 图表。

```
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
```

在 Grafana 中添加 Prometheus 数据来源

以下步骤说明了如何在 Grafana 中设置 Prometheus 数据来源，以便查询您的 Amazon Managed Service for Prometheus 指标。

在您的 Grafana 服务器中添加 Prometheus 数据来源

1. 打开 Grafana 控制台。
2. 在配置下，选择数据来源。
3. 选择添加数据来源。
4. 选择 Prometheus。
5. 对于 HTTP URL，请指定 Amazon Managed Service for Prometheus 控制台的工作区详情页面中显示的终端节点 - 查询 URL。

6. 在您刚才指定的 HTTP URL 中，删除附加到该 URL 的 `/api/v1/query` 字符串，因为 Prometheus 数据来源会自动附加该字符串。
7. 在身份验证下，选择 Sigv4 身份验证的开关将其启用。

   将担任角色 ARN 和外部 ID字段留空。然后在默认区域中，选择您 Amazon Managed Service for Prometheus 工作区的区域。
8. 选择保存并测试。

   您应该看到以下消息：数据来源正在运行
9. 针对新的数据来源测试 PromQL 查询：

   a. 选择探索。

   b. 运行示例 PromQL 查询，例如：

   ```
   prometheus_tsdb_head_series
   ```

使用与 Prometheus 兼容的 API 进行查询

尽管使用诸如 Amazon Managed Grafana 之类的工具是查看和查询指标的最简单方法，但 Amazon Managed Service for Prometheus 还支持多个与 Prometheus 兼容的 API，您可以使用这些 API 来查询您的指标。有关所有可用的与 Prometheus 兼容的 API 的更多信息，请参阅 与 Prometheus 兼容的 API。

使用这些 API 查询指标时，必须使用签 AWS 名版本 4 签名流程对请求进行签名。您可以设置 AWS 签名版本 4 来简化签名流程。有关更多信息，请参阅 aws-sigv4-proxy。

可使用通过 AWS SigV4 代理进行签名。`awscurl`以下主题使用 awscurl 查询与 Prometheus 兼容的 API 将引导您完成使用 `awscurl` 设置 AWS Sigv4 的过程。

使用 awscurl 查询与 Prometheus 兼容的 API

Amazon Managed Service for Prometheus 的 API 请求必须使用 SigV4 签名。您可以使用 awscurl 来简化查询过程。

要安装 `awscurl`，您需要安装 Python 3 和 pip 软件包管理器。

在基于 Linux 的实例上，以下命令将安装 `awscurl`。

```
$ pip3 install awscurl
```

在 macOS 计算机上，以下命令将安装 awscurl。

```
$ brew install awscurl
```

以下示例是一个示例awscurl查询。将*##*、*Workspace-ID* 和 *QUERY* 输入替换为适合您的用例的值：

```
# 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**
>
> 您的查询字符串必须经过网址编码。

对于类似的查询query=up，你可以得到如下结果：

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

```
          1652452637.636,
          "1"
        ]
      },
    ]
  }
}
```

为了让 awscurl 签署所提供的请求，您需要通过以下方式之一传递有效的凭证：

- 提供 IAM 角色的访问密钥 ID 和密钥。您可以在 https://console.aws.amazon.com/iam/ 中找到该角色的访问密钥和密钥。

  例如：

  ```
  $ 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>"
  ```

- 参考存储在 .aws/credentials 和 /aws/config 文件中的配置文件。您也可以选择指定要使用的配置文件的名称。如果未指定，则将使用 default 文件。例如：

  ```
  $ 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>"
  ```

- 使用与 EC2 实例关联的实例配置文件。

使用 awscurl 容器执行查询请求

当安装不同版本的 Python 和相关的依赖项不可行时，可以使用容器来打包 awscurl 应用程序及其依赖项。以下示例使用 Docker 运行时部署 awscurl，但任何符合 OCI 的运行时和映像都可正常工作。

```
$ 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"
```

在查询 API 响应中查询统计信息

查询定价基于一个月内从执行的查询中处理的查询样本总数。query 或 queryRange API 的查询响应包括有关已处理的查询样本的统计信息。在请求中发送查询参数 stats=all 时，将在 stats 对象中创建一个 samples 对象，并在响应中返回 stats 数据。

samples 对象包含以下属性：

属性	描述
totalQueryableSamples	已处理的查询样本总数。这是用于计费的信息。
totalQueryableSamplesPerStep	每个步骤处理的查询样本数。其结构为一组数组，时间戳以纪元为单位，并包含在特定步骤上加载的样本数量。

包含 stats 信息的示例请求和响应如下所示：

query 的示例：

GET

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

响应

```
{
    "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 的示例：

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
```

响应

```
{
    "status": "success",
    "data": {
```

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

记录规则和警报规则

Amazon Managed Service for Prometheus 支持两种类型的规则，并定期对其进行评估：

- 记录规则让您可以预先计算经常需要或计算成本高昂的表达式，并将其结果另存为一组新的时间序列。相比于每次需要时都运行原始表达式，查询预先计算的结果通常快得多。
- 警报规则让您可以根据 PromQL 和阈值定义警报条件。当规则触发阈值时，会向警报管理器发送通知，警报管理器会将通知向下游转发给 Amazon Simple Notification Service 等接收方。

要在 Amazon Managed Service for Prometheus 中使用规则，您需要创建一个或多个 YAML 规则文件来定义规则。Amazon Managed Service for Prometheus 规则文件的格式与独立 Prometheus 中的规则文件格式相同。有关更多信息，请参阅 Prometheus 文档中的 Defining Recording rules 和 Alerting rules。

一个工作区中可以有多个规则文件。每个单独的规则文件都包含在单独的命名空间中。有了多个规则文件，您便可以将现有 Prometheus 规则文件导入工作区，而无需对其进行更改或合并。不同的规则组命名空间也可以有不同的标签。

规则排序

在规则文件中，规则包含在规则组中。规则文件中单个规则组中的规则始终按从上到下的顺序进行评估。因此，在记录规则中，一条记录规则的结果可以用于计算以后的记录规则，也可以用于同一规则组中的警报规则。但是，由于您无法指定运行单独规则文件的顺序，因此不能使用一条记录规则的结果来计算其它规则组或其它规则文件中的规则。

主题

必要的 IAM 权限

必须向用户授予使用 Amazon Managed Service for Prometheus 中规则的权限。创建具有以下权限的 AWS Identity and Access Management (IAM) 策略，并将该策略分配给您的用户、群组或角色。

> **Note**
>
> 有关 IAM 的更多信息，请参阅。Amazon Managed Service for Prometheus 的身份和访问管理

授权用户使用规则的策略

以下策略授权使用账户中所有资源的规则。

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

仅授予一个命名空间访问权限的策略

您也可以创建仅允许访问特定策略的策略。以下示例策略仅授予对指定的 RuleGroupNamespace 的访问权限。要使用此策略，请将 *<account>*、*<region>*、*<workspace-id>* 和 *<namespace-name>* 替换为您账户对应的值。

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

创建规则文件

要在 Amazon Managed Service for Prometheus 中使用规则，您需要创建一个规则文件来定义规则。Amazon Managed Service for Prometheus 规则文件的格式与独立 Prometheus 中的规则文件格式相同。有关更多信息，请参阅定义记录规则和警报规则。

以下是规则文件的基本示例：

```
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
```

有关更多警报规则示例，请参阅警报规则示例。

> **Note**
>
> 您可以在本地创建规则定义文件，然后将其上传到适用于 Prometheus 的亚马逊托管服务，也可以直接在适用于 Prometheus 的亚马逊托管服务控制台中创建、编辑和上传定义。无论哪种

方式，都适用相同的格式规则。要了解有关上传和编辑文件的更多信息，请参阅将规则配置文件上传到 Amazon Managed Service for Prometheus。

将规则配置文件上传到 Amazon Managed Service for Prometheus

一旦知道要对规则配置文件进行哪些更改，就可以在控制台中对其进行编辑，也可以使用控制台上传替换文件或 AWS CLI。

> **Note**
>
> 如果您运行的是 Amazon EKS 集群，也可以使用适用于 Kubernetes 的AWS 控制器上传规则配置文件。

使用适用于 Prometheus 的亚马逊托管服务控制台编辑或替换您的规则配置并创建命名空间

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在页面左上角，选择菜单图标，然后选择所有工作区。
3. 选择工作区的工作区 ID，然后选择规则管理选项卡。
4. 选择添加命名空间。
5. 选择选择文件，然后选择规则定义文件。

   或者，您可以选择定义配置，直接在适用于 Prometheus 的亚马逊托管服务控制台中创建和编辑规则定义文件。这将创建一个示例默认定义文件，供您在上传之前对其进行编辑。
6. （可选）要向命名空间添加标签，请选择添加新标签。

   然后，对于 Key（键），输入标签的名称。您可以在 Value (值) 中添加可选的标签值。

   要添加其它标签，添加新标签。
7. 选择继续。Amazon Managed Service for Prometheus 会创建一个与您选择的规则文件同名的新命名空间。

使用将警报管理器配置上传 AWS CLI 到新命名空间中的工作区

1. Base64 对警报管理器文件的内容进行编码。在 Linux 系统上，您可以使用以下命令：

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

在 macOS 系统上，您可以使用以下命令：

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

2. 输入以下命令之一即可创建命名空间并上传文件。

   在 AWS CLI 版本 2 上，输入：

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

   在 AWS CLI 版本 1 上，输入：

   ```
   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. 您的警报管理器配置需要几秒钟才能生效。要检查状态，请输入以下命令：

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

   如果 `status` 是 `ACTIVE`，则您的规则文件已生效。

编辑规则配置文件

您可以上传新的规则文件来替换现有配置，也可以直接在控制台中编辑当前配置。或者，您可以下载当前文件，在文本编辑器中对其进行编辑，然后上传新版本。

使用 Amazon Managed Service for Prometheus 控制台编辑您的规则配置

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在页面左上角，选择菜单图标，然后选择所有工作区。
3. 选择工作区的工作区 ID，然后选择规则管理选项卡。
4. 选择要编辑的规则配置文件的名称。

5. （可选）如果要下载当前规则配置文件，请选择下载或复制。
6. 选择 M od ify 可直接在控制台中编辑配置。完成后选择 “保存”。

   或者，您可以选择 “替换配置” 来上传新的配置文件。如果是，请选择新的规则定义文件，然后选择继续上传。

AWS CLI 使用编辑规则配置文件

1. Base64 对规则文件的内容进行编码。在 Linux 系统上，您可以使用以下命令：

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

   在 macOS 系统上，您可以使用以下命令：

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

2. 输入下列命令之一即可上传新文件。

   在 AWS CLI 版本 2 上，输入：

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

   在 AWS CLI 版本 1 上，输入：

   ```
   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. 您的规则文件需要几秒钟才能生效。要检查状态，请输入以下命令：

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

   如果 `status` 是 `ACTIVE`，则您的规则文件已生效。在此之前，此规则文件的先前版本仍处于活动状态。

规则器故障排除

使用 CloudWatch 日志，您可以对警报管理器和规则器相关问题进行故障排除。本部分包含与规则器相关的故障排除主题。

当日志包含以下规则器失败错误时

```
{
    "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"
}
```

这意味着在执行规则时出现了一些错误。

要采取的操作

使用错误消息对规则执行进行故障排除。

警报管理器

当 Amazon Managed Service for Prometheus 运行的警报规则触发时，警报管理器会处理发送的警报。警报管理器会对警报进行重复数据删除、分组并路由到下游接收方。Amazon Managed Service for Prometheus 仅支持 Amazon Simple Notification Service 作为接收方，并且可以在同一个账户中将消息路由到 Amazon SNS 主题。您还可以使用警报管理器来静默和抑制警报。

警报管理器提供的功能与 Prometheus 中的 Alertmanager 类似。

您可以使用警报管理器的配置文件进行以下操作：

- 分组 - 分组操作会将类似的警报收集到单个通知中。当许多系统同时出现故障并且可能同时触发数百个警报时，这在较大的停机故障中特别有用。例如，假设网络故障导致多个节点同时出现故障。如果将这些类型的警报分组，警报管理器会向您发送一条通知。

  警报分组和分组通知的时间由警报管理器配置文件中的路由树配置。有关更多信息，请参阅 <route>。
- 抑制 - 如果某些其它警报已经触发，则抑制功能会抑制某些警报的通知。例如，如果针对集群无法访问触发警报，则可以将警报管理器配置为将与该集群有关的所有其它警报静音。这样可以防止收到与实际问题无关的成百甚至数千个触发警报的通知。有关如何编写抑制规则的更多信息，请参阅 <inhibit_rule>。
- 静默 – 在指定时间（例如维护时段）内将静音警报设置为静默。检查传入的警报是否与活动静默的所有等式匹配器或正则表达式匹配器匹配。如果匹配，则不会针对该警报发送任何通知。

  要创建静默，请使用 `PutAlertManagerSilences` API。有关更多信息，请参阅 PutAlertManagerSilences。

Prometheus 模板

独立的 Prometheus 支持使用分隔模板文件进行模板化。模板可以使用条件语句和格式化数据等。

在适用于 Prometheus 的亚马逊托管服务中，您可以将模板放在与警报管理器配置相同的警报管理器配置文件中。

主题

- 必要的 IAM 权限
- 创建警报管理器配置文件

- 设置警报接收方
- 将您的警报管理器配置文件上传到 Amazon Managed Service for Prometheus
- 将警报与 Amazon Managed Grafana 或开源 Grafana 集成
- 警报管理器故障排除

必要的 IAM 权限

必须向用户授予使用 Amazon Managed Service for Prometheus 中规则的权限。创建具有以下权限的 AWS Identity and Access Management (IAM) 策略，并将该策略分配给您的用户、群组或角色。

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

创建警报管理器配置文件

要在 Amazon Managed Service for Prometheus 中使用警报管理器和模板，您需要创建警报管理器配置 YAML 文件。Amazon Managed Service for Prometheus 警报管理器文件分为两个主要部分：

- `template_files`：包含用于接收方发送的消息的模板。有关更多信息，请参阅 Prometheus 文档中的 Template Reference 和 Template Examples。
- `alertmanager_config`：包含警报管理器配置。它使用的结构与独立的 Prometheus 中的警报管理器配置文件相同。有关更多信息，请参阅 Alertmanager 文档中的 Configuration。

> **Note**
>
> 以上 Prometheus 文档中所述的 `repeat_interval` 配置在 Amazon Managed Service for Prometheus 中还有一个额外的限制。允许的最大值为五天。如果您将其设置为高于五天，则会将其视为五天，并且将在五天期限过后再次发送通知。

> **Note**
>
> 您也可以直接在适用于 Prometheus 的亚马逊托管服务控制台中编辑配置文件，但它仍必须遵循此处指定的格式。有关上传或编辑配置文件的更多信息，请参阅将您的警报管理器配置文件上传到 Amazon Managed Service for Prometheus。

在 Amazon Managed Service for Prometheus 中，您的警报管理器配置文件必须将所有警报管理器配置内容包含在 YAML 文件根目录的 `alertmanager_config` 密钥中。

以下是警报管理器配置文件基本示例：

```
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
```

目前支持的唯一接收方是 Amazon Simple Notification Service (Amazon SNS)。如果配置中列出了其它类型的接收方，则该接收方将被拒绝。

这是另一个同时使用 `template_files` 数据块和 `alertmanager_config` 数据块的警报管理器配置文件示例。

```
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
```

默认 Amazon SNS 模板数据块

除非您明确覆盖以下模板，否则默认 Amazon SNS 配置将使用以下模板。

```
{{ 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 }}
```

设置警报接收方

Amazon Managed Service for Prometheus 目前支持的唯一警报接收方是 Amazon Simple Notification Service (Amazon SNS)。有关更多信息，请参阅 Amazon SNS 是什么？。

主题

- （可选）创建新的 Amazon SNS 主题
- 授予 Amazon Managed Service for Prometheus 向您的 Amazon SNS 主题发送消息的权限
- 在警报管理器配置文件中指定您的 Amazon SNS 主题
- （可选）配置警报管理器以将 JSON 输出到 Amazon SNS
- （可选）从 Amazon SNS 发送到其它目标
- SNS 接收方消息验证和截断规则

（可选）创建新的 Amazon SNS 主题

您可以使用现有的 Amazon SNS 主题或创建一个新主题。我们建议您使用标准类型的主题，这样您就可以将来自该主题的警报转发到电子邮件、短信或 HTTP。

要创建新的 Amazon SNS 主题作为警报管理器接收方，请按照步骤 1：创建主题中的步骤进行操作。主题类型请务必选择标准。

如果您希望每次向该 Amazon SNS 主题发送消息时都收到电子邮件，请按照步骤 2：创建主题订阅中的步骤进行操作。

授予 Amazon Managed Service for Prometheus 向您的 Amazon SNS 主题发送消息的权限

您必须授予 Amazon Managed Service for Prometheus 向您的 Amazon SNS 主题发送消息的权限。以下策略声明包括一份 `Condition` 声明，旨在帮助防止出现混淆代理人安全问题。该 `Condition` 声

明限制了对 Amazon SNS 主题的访问权限，仅允许来自该特定账户和 Amazon Managed Service for Prometheus 工作区的操作。有关混淆代理人问题的更多信息，请参阅防止跨服务混淆座席。

授予 Amazon Managed Service for Prometheus 向您的 Amazon SNS 主题发送消息的权限

1. 通过 https://console.aws.amazon.com/sns/v3/home 打开 Amazon SNS 控制台。
2. 在导航窗格中，选择 Topics（主题）。
3. 选择您正用于 Amazon Managed Service for Prometheus 的主题的名称。
4. 选择编辑。
5. 选择访问策略，将以下策略声明添加到现有策略。

```
{
    "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"
}
```

[可选] 如果您的 SNS 主题启用了服务端加密（SSE），则需要在 "Action" 数据块中向 KMS 密钥策略添加以下权限。有关更多信息，请参阅 SNS 主题的AWS KMS 权限。

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

6. 选择保存更改。

> **Note**
>
> 默认情况下，Amazon SNS 会创建带有 `AWS:SourceOwner` 条件的访问策略。有关更多信息，请参阅 SNS 访问策略。

> **Note**
>
> IAM 遵循最严格的策略优先规则。在您的 SNS 主题中，如果存在比记录在案的 Amazon SNS 策略块更严格的策略数据块，则不会授予该主题策略的权限。要评估您的策略并了解已授予的权限，请参阅策略评估逻辑。

防止跨服务混淆座席

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务以使用其权限对另一个客户的资源进行操作，否则该服务不应有访问权限。为了防止这种情况，我们 AWS 提供了一些工具，帮助您保护所有服务的数据，这些服务委托人已被授予访问您账户中资源的权限。

我们建议在资源策略中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键，以限制 Amazon Managed Service for Prometheus 为 Amazon SNS 提供的资源访问权限。如果使用两个全局条件上下文键，在同一策略语句中使用时，`aws:SourceAccount` 值和 `aws:SourceArn` 值中的账户必须使用相同的账户 ID。

`aws:SourceArn` 的值必须为 Amazon Managed Service for Prometheus 工作区的 ARN。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符 (*) 的 `aws:SourceArn` 全局上下文条件键。例如，`arn:aws:`*`servicename`*`::`*`123456789012`*`:*`。

授予 Amazon Managed Service for Prometheus 向您的 Amazon SNS 主题发送消息的权限 中所示的策略演示了如何使用 Amazon Managed Service for Prometheus 中的 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键来防范混淆代理人问题。

在警报管理器配置文件中指定您的 Amazon SNS 主题

现在，您可以将 Amazon SNS 接收方添加到警报管理器配置中。为此，您必须知道 Amazon SNS 主题的 Amazon 资源名称 (ARN)。

有关 Amazon SNS 接收方配置的更多信息，请参阅 Prometheus 配置文档中的 <sns_configs>。

不支持的属性

Amazon Managed Service for Prometheus 支持 Amazon SNS 作为警报接收方。但是，由于服务限制，并非支持 Amazon SNS 接收方的所有属性。Amazon Managed Service for Prometheus 警报管理器配置文件中不允许使用以下属性：

- `api_url:` – Amazon Managed Service for Prometheus 会为您设置 `api_url`，因此不允许使用此属性。
- `Http_config` – 此属性允许您设置外部代理。Amazon Managed Service for Prometheus 目前不支持此功能。

此外，还需要 SigV4 设置才能具有 Region 属性。如果没有 Region 属性，Amazon Managed Service for Prometheus 就没有足够的信息来提出授权请求。

配置将您的 Amazon SNS 主题作为接收方的警报管理器

1. 如果您使用的是现有的警报管理器配置文件，请在文本编辑器中打开该文件。
2. 如果 `receivers` 数据块中当前有 Amazon SNS 以外的接收方，请将其移除。您可以将多个 Amazon SNS 主题配置为接收方，方法是将它们放在 `receivers` 数据块内单独的 `sns_config` 数据块中。
3. 在 `receivers` 部分中添加以下 YAML 数据块。

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

如果未指定 `subject`，则默认情况下，将使用带有标签名称和值的默认模板生成主题，这可能会导致值对于 SNS 来说太长。要更改应用于主题的模板，请参阅本指南中的 （可选）配置警报管理器以将 JSON 输出到 Amazon SNS。

现在，必须将警报管理器配置文件上传到 Amazon Managed Service for Prometheus。有关更多信息，请参阅 将您的警报管理器配置文件上传到 Amazon Managed Service for Prometheus。

（可选）配置警报管理器以将 JSON 输出到 Amazon SNS

您可以将警报管理器配置为以 JSON 格式发送警报，以便可以在接收 webhook 的终端节点中从 Amazon SNS 的下游 AWS Lambda 或接收网络挂钩的终端节点中进行处理。Amazon Managed Service for Prometheus 附带的默认模板以文本列表格式输出消息负载，这可能不容易解析。您可以定义一个自定义模板来以 JSON 格式输出消息内容，这样可以更轻松地在下游函数中进行解析，而不必使用默认模板。

要以 JSON 格式将警报管理器中的消息输出到 Amazon SNS，请更新警报管理器配置，在 `template_files` 根部分中包含以下代码：

```
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**
>
> 此模板根据字母数字数据创建 JSON。如果您的数据包含特殊字符，请在使用此模板之前对其进行编码。

为确保在传出通知中使用此模板，请在您的 `alertmanager_config` 数据块中引用该模板，如下所示：

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

> **Note**
>
> 此模板适用于整个消息正文，采用 JSON 格式。此模板会覆盖整个消息正文。如果您想使用此特定模板，则不能覆盖消息正文。任何手动完成的覆盖都将优先于模板。

有关以下内容的更多信息：

- 警报管理器配置文件，请参阅 创建警报管理器配置文件。
- 上传您的配置文件，请参阅 将您的警报管理器配置文件上传到 Amazon Managed Service for Prometheus。

（可选）从 Amazon SNS 发送到其它目标

目前，Amazon Managed Service for Prometheus 只能直接向Amazon SNS 发送警报消息。您可以将 Amazon SNS 配置为将这些消息发送到其他目的地，例如电子邮件、webhook、Slack 和。 OpsGenie

Email

要将 Amazon SNS 主题配置为将消息输出到电子邮件，请创建订阅。在 Amazon SNS 控制台中，选择订阅选项卡以打开订阅列表页面。选择创建订阅，然后选择电子邮件。Amazon SNS 将向列出的电

子邮件地址发送确认电子邮件。接受确认后，您就可以通过电子邮件接收来自您订阅主题的 Amazon SNS 通知。有关更多信息，请参阅订阅 Amazon SNS 主题。

Webhook

要将 Amazon SNS 主题配置为将消息输出到 Webhook 终端节点，请创建订阅。在 Amazon SNS 控制台中，选择订阅选项卡以打开订阅列表页面。选择创建订阅，然后选择 HTTP/HTTPS。创建订阅后，必须按照确认步骤将其激活。当订阅处于活动状态时，您的 HTTP 终端节点应该会收到 Amazon SNS 通知。有关更多信息，请参阅订阅 Amazon SNS 主题。有关使用 Slack Webhook 向各目标发布消息的更多信息，请参阅如何使用 Webhook 将 Amazon SNS 消息发布到 Amazon Chime、Slack 或 Microsoft Teams？

Slack

要将 Amazon SNS 主题配置为向 Slack 输出消息，您有两个选择。你可以与 Slack 的 email-to-channel 集成集成，允许 Slack 接受电子邮件并将其转发到 Slack 频道，也可以使用 Lambda 函数将亚马逊 SNS 通知重写到 Slack。有关将电子邮件转发到 slack 频道的更多信息，请参阅确认 Slack Webhook 的 AWS SNS 主题订阅。有关构建 Lambda 函数以将 Amazon SNS 消息转换为 Slack 的更多信息，请参阅如何将 Amazon Managed Service for Prometheus 与 Slack 集成。

OpsGenie

有关如何配置要向其输出消息的 Amazon SNS 主题的信息，请参阅将 Opsgenie 与传入的亚马逊 SNS 集成。 OpsGenie

SNS 接收方消息验证和截断规则

如有必要，SNS 接收方将根据以下规则验证、截断或修改 SNS 消息：

- 消息包含非 UTF 字符。
  - 消息将替换为 “Error - not a valid UTF-8 encoded string.”
  - 将添加一个消息属性，键为“truncated”，值为“true”
  - 将添加一个消息属性，键为“modified”，值为“Message: Error - not a valid UTF-8 encoded string.”
- 消息为空。
  - 消息将替换为“Error - Message should not be empty.”
  - 将添加一个消息属性，键为“modified”，值为“MMessage: Error - Message should not be empty.”
- 消息已被截断。

  - 消息将包含被截断的内容。
  - 将添加一个消息属性，键为“truncated”，值为“true”
  - 将添加一个消息属性，键为“modified”，值为“Message: Error - Message has been truncated from *X* KB, because it exceeds the 256 KB size limit.”
- 主题不是 ASCII 字符。
  - 主题将替换为“Error - contains non printable ASCII characters.”
  - 将添加一个消息属性，键为“modified”，值为 “Subject: Error - contains non-printable ASCII characters.”
- 主题已被截断。
  - 主题将包含被截断的内容。
  - 将添加一个消息属性，键为“modified”，值为“Subject: Error - Subject has been truncated from *X* characters, because it exceeds the 100 character size limit.”
- 消息属性的键/值无效。
  - 无效的消息属性将被删除。
  - 将添加一个消息属性，键为 “已修改”，值为 “MessageAttribute: Error-由于无效 MessageAttributeKey 或，消息属性中有 *X* 个已被删除。” MessageAttributeValue
- 消息属性已被截断。
  - 额外的消息属性将被删除。
  - 将添加一个带有 “已修改” 键的消息属性，值为 “MessageAttribute: Error-*X*” 的消息属性已被删除，因为它超过了 256KB 的大小限制。

将您的警报管理器配置文件上传到 Amazon Managed Service for Prometheus

一旦您知道要对警报管理器配置文件进行哪些更改，就可以在控制台中对其进行编辑，也可以使用控制台上传替换文件或 AWS CLI。

> **Note**
>
> 如果您运行的是 Amazon EKS 集群，也可以使用适用于 Kubernetes 的AWS 控制器上传警报管理器配置文件。

使用适用于 Prometheus 的亚马逊托管服务控制台编辑或替换您的警报管理器配置

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在页面左上角，选择菜单图标，然后选择所有工作区。
3. 选择工作区的工作区 ID，然后选择警报管理器选项卡。
4. 如果工作区还没有警报管理器定义，请选择添加定义。

> **Note**
>
> 如果工作区有要替换的警报管理器定义，请改为选择修改。

5. 选择选择文件，选择警报管理器定义文件，然后选择继续。

> **Note**
>
> 或者，您可以通过选择 “创建定义” 选项来创建新文件并直接在控制台中对其进行编辑。这将创建一个默认配置示例，供您在上传之前对其进行编辑。

首次使用 AWS CLI 将警报管理器配置上传到工作区

1. Base64 对警报管理器文件的内容进行编码。在 Linux 系统上，您可以使用以下命令：

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

在 macOS 系统上，您可以使用以下命令：

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

2. 要上传文件，请输入以下命令之一。

在 AWS CLI 版本 2 上，输入：

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

在 AWS CLI 版本 1 上，输入：

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

3. 您的警报管理器配置需要几秒钟才能生效。要检查状态，请输入以下命令：

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

如果 `status` 是 `ACTIVE`，则表示您的新警报管理器定义已生效。

使用将工作区的警报管理器配置替换为新的警报管理器配置 AWS CLI

1. Base64 对警报管理器文件的内容进行编码。在 Linux 系统上，您可以使用以下命令：

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

在 macOS 系统上，您可以使用以下命令：

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

2. 要上传文件，请输入以下命令之一。

在 AWS CLI 版本 2 上，输入：

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

在 AWS CLI 版本 1 上，输入：

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

3. 您的新警报管理器配置需要几秒钟才能生效。要检查状态，请输入以下命令：

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

如果 `status` 是 `ACTIVE`，则表示您的新警报管理器定义已生效。在此之前，您之前的警报管理器配置仍处于活动状态。

将警报与 Amazon Managed Grafana 或开源 Grafana 集成

您在 Amazon Managed Service for Prometheus 的 Alertmanager 中创建的警报规则可以在 Amazon Managed Grafana 和 Grafana 中转发和查看，从而在一个环境中统一您的警报规则和警报。在 Amazon Managed Grafana 中，您可以查看您的警报规则和生成的警报。

先决条件

在开始将 Amazon Managed Service for Prometheus 集成到 Amazon Managed Grafana 之前，您必须满足以下先决条件：

- 您必须拥有现有 AWS 账户和 IAM 凭证，才能以编程方式创建 Amazon Managed Service for Prometheus 和 IAM 角色。

  有关创建 AWS 账户和 IAM 凭证的更多信息，请参阅设置。
- 您必须拥有 Amazon Managed Service for Prometheus 工作区，并且要向其中摄取数据。要设置新的工作区，请参阅创建工作区。您还应该熟悉 Prometheus 的概念，例如 Alertmanager 和 Ruler。有关这些主题的更多信息，请参阅 Prometheus 文档。
- 您已经在 Amazon Managed Service for Prometheus 中配置了 Alertmanager 配置和规则文件。有关 Amazon Managed Service for Prometheus 中的 Alertmanager 的更多信息，请参阅警报管理器。有关规则的更多信息，请参阅记录规则和警报规则。
- 您必须设置 Amazon Managed Grafana，或者运行开源版本的 Grafana。
  - 如果您使用的是 Amazon Managed Grafana，则必须使用 Grafana 警报。有关更多信息，请参阅将旧版控制面板警报迁移到 Grafana 警报。
  - 如果您使用的是开源版本的 Grafana，则必须运行版本 9.1 或更高版本。

> **Note**
>
> 您可以使用早期版本的 Grafana，但必须启用统一警报（Grafana 警报）功能，并且可能需要设置 sigv4 代理才能从 Grafana 调用 Amazon Managed Service for Prometheus。有关更多信息，请参阅设置 Grafana 开源或 Grafana Enterprise 以与 Amazon Managed Service for Prometheus 配合使用。

- Amazon Managed Grafana 必须具有以下权限才能访问您的 Prometheus 资源。您必须将其添加到 https://docs.aws.amazon.com/grafana/latest/userguide/AMG-manage-permissions.html 中所述的服务管理策略或客户管理策略中。
  - `aps:ListRules`
  - `aps:ListAlertManagerSilences`
  - `aps:ListAlertManagerAlerts`
  - `aps:GetAlertManagerStatus`
  - `aps:ListAlertManagerAlertGroups`
  - `aps:PutAlertManagerSilences`
  - `aps:DeleteAlertManagerSilence`

设置 Amazon Managed Grafana

如果您已经在 Amazon Managed Service for Prometheus 实例中设置了规则和警报，则使用 Amazon Managed Grafana 作为这些警报控制面板的配置完全在 Amazon Managed Grafana 中完成。

将 Amazon Managed Grafana 配置为警报控制面板

1. 打开工作区的 Grafana 控制台。
2. 在配置下，选择数据来源。
3. 创建或打开您的 Prometheus 数据来源。如果您之前未设置 Prometheus 数据来源，请参见在 Grafana 中添加 Prometheus 数据来源以获取更多信息。
4. 在 Prometheus 数据来源中，选择通过 Alertmanager UI 管理警报。
5. 返回数据来源界面。
6. 创建新的 Alertmanager 数据来源。
7. 在 Alertmanager 数据来源配置页面中，添加以下设置：
   - 将实施设置为 `Prometheus`。
   - 对于 URL 设置，请使用您的 Prometheus 工作区的 URL，删除工作区 ID 之后的所有内容，然后在末尾附加 `/alertmanager`。例如，*`https://aps-workspaces.us-east1.amazonaws.com/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager`*。
   - 在身份验证下，开启 SigV4Auth。这会告诉 Grafana 对请求使用 AWS 身份验证。

- 例如，在 Sigv4Auth 详细信息下的默认区域中，提供您的 Prometheus 实例所在的区域，例如 `us-east-1`。
- 将默认选项设置为 `true`。

8. 选择保存并测试。
9. 现在，您的 Amazon Managed Service for Prometheus 警报应配置为与您的 Grafana 实例配合使用。确认您可以在 Grafana 警报页面的 Amazon Managed Service for Prometheus 实例中看到所有警报规则、警报组（包括活动警报）和静默。

警报管理器故障排除

使用 CloudWatch 日志，您可以对警报管理器和规则器相关问题进行故障排除。本部分包含与警报管理器相关的故障排除主题。

主题

空内容警告

当日志包含以下警告时

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

这表示警报管理器模板已将出站警报解析为空消息。

要采取的操作

验证您的警报管理器模板并确保所有接收方路径都有一个有效的模板。

非 ASCII 警告

当日志包含以下警告时

```
{
   "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"
}
```

这表示主题包含非 ASCII 字符。

要采取的操作

删除模板主题字段中对可能包含非 ASCII 字符的标签的引用。

`key/value` 警告无效

当日志包含以下警告时

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

这表示由于键/值无效，某些消息属性已被删除。

要采取的操作

重新评估您用来填充消息属性的模板，并确保其解析为有效的 SNS 消息属性。有关验证 Amazon SNS 主题的更多信息，请参阅[验证 SNS 主题]

消息限制警告

当日志包含以下警告时

```
{
   "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"
}
```

这表示有些消息大小太大。

要采取的操作

查看警报接收方消息模板，然后对其进行修改以满足大小限制。

没有基于资源的策略错误

当日志包含以下错误时

```
{
   "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"
}
```

这表示 Amazon Managed Service for Prometheus 无权向指定的 SNS 主题提交警报。

要采取的操作

验证 Amazon SNS 主题访问策略是否授权 Amazon Managed Service for Prometheus 向该主题发送 SNS 消息。创建 SNS 访问策略，授予服务`aps.amazonaws.com`（适用于 Prometheus 的亚马逊托

管服务）访问您的亚马逊 SNS 主题的权限。有关 SNS 访问策略的更多信息，请参阅《亚马逊简单通知服务开发者指南》中的使用访问策略语言和 Amazon SNS 访问控制示例案例。

日记账记录和监控

您可以使用亚马逊日志和监控功能管理适用于 Prometheus 的 CloudWatch 亚马逊托管服务资源使用情况。

- 使用 CloudWatch 指标 监控 Amazon Managed Service for Prometheus。
- 使用 CloudWatch 日志 查询和查看 Amazon Managed Service for Prometheus 警报管理器和规则器事件。

CloudWatch 指标

适用于 Prometheus 的亚马逊托管服务将使用量指标提供给。 CloudWatch这些指标可让您了解您的工作区利用率。出售的指标可以在中的AWS/Usage和AWS/Prometheus命名空间中找到。 CloudWatch这些指标是免费提供 CloudWatch的。有关使用量指标的更多信息，请参阅CloudWatch 使用量指标。

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	描述
ResourceCount	IngestionRate	AWS/Usage	样本摄取率 单位：每秒计数 有效统计数据：Average、Minimum、Maximum、Sum
ResourceCount	ActiveSeries	AWS/Usage	每个工作区的活跃系列数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
ResourceCount	ActiveAlerts	AWS/Usage	每个工作区的活动警报数 单位：计数

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	描述
			有效统计数据：Average、Minimum、Maximum、Sum
ResourceCount	SizeOf警报	AWS/Usage	工作空间中所有警报的总大小，以字节为单位 单位：字节 有效统计数据：Average、Minimum、Maximum、Sum
ResourceCount	SuppressedAlerts	AWS/Usage	每个工作区处于抑制状态的警报数量。可以通过静默或抑制来抑制警报。 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
ResourceCount	UnprocessedAlerts	AWS/Usage	每个工作区处于未处理状态的警报数量。警报一经接收，即处于未处理状态 AlertManager，但正在等待下一次聚合组评估。 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	描述
ResourceCount	AllAlerts	AWS/Usage	每个工作区处于任何状态的警报数量。 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
AlertManagerAlertsReceived	-	AWS/Prometheus	警报管理器收到的成功警报总数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
AlertManagerNotificationsFailed	-	AWS/Prometheus	发送失败的警报数量 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
AlertManagerNotificationsThrottled	-	AWS/Prometheus	限制的警报数量 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	描述
Discarded Samples*	-	AWS/Prometheus	按原因划分的丢弃样本数量 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
RuleEvaluations	-	AWS/Prometheus	规则评估总数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
RuleEvaluation 失败	-	AWS/Prometheus	间隔内规则评估失败的次数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
RuleGroup IterationsMissed	-	AWS/Prometheus	间隔内错过的规则组迭代次数。 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum

*导致样本被丢弃的一些原因如下。

Reason	含义
greater_than_max_sample_age	丢弃超过一小时的样本。
new-value-for-timestamp	发送重复样本的时间戳与之前记录的时间戳不同。
per_metric_series_limit	用户已达到每个指标的活跃序列上限。
per_user_series_limit	用户已达到活动系列总数上限。
rate_limited	摄取速率有限。
sample-out-of-order	样品是乱发的，无法处理。
label_value_too_long	标签值超过允许的字符限制。
max_label_names_per_series	用户已点击每个指标的标签名称。
missing_metric_name	未提供指标名称。
metric_name_invalid	提供的指标名称无效。
label_invalid	提供的标签无效。
duplicate_label_names	提供了重复的标签名称。

Note

指标不存在或缺失等同于该指标的值为 0。

Note

`RuleGroupIterationsMissed`、`RuleEvaluations` 和 `RuleEvaluationFailures` 具有以下结构的 `RuleGroup` 维度：

`RuleGroup####`；*`RuleGroup`*

对 Prometheus 出售的指标设置 CloudWatch 警报

您可以使用警报监控 Prometheus 资源的使用情况。 CloudWatch

在 Prometheus 中为 prometheus ActiveSeries中的数字设置警报

1. 选择 “图表化指标” 选项卡，然后向下滚动到ActiveSeries标签。

   在 Graphed 指标视图中，只会显示当前所摄取的指标。
2. 在操作列中选择通知图标。
3. 在指定指标和条件中的条件值字段中输入阈值条件，然后选择下一步。
4. 在配置操作中，选择现有的 SNS 主题或创建一个新 SNS 主题以将通知发送到该 SNS 主题。
5. 在添加名称和描述中，添加警报的名称和可选描述。
6. 选择创建警报。

CloudWatch 日志

适用于 Prometheus 的亚马逊托管服务在亚马逊日志的日志组中记录警报管理器和统治者的错误和警告事件。 CloudWatch 有关警报管理器和规则器的更多信息，请参阅本指南中的警报管理器主题。您可以在 Logs 中将工作空间日志数据发布到 CloudWatch 日志流中。您可以在 Amazon Managed Service for Prometheus 控制台中配置要监控的日志，也可以使用 AWS CLI配置。您可以在 CloudWatch 控制台中查看或查询这些日志。有关在控制台中查看 CloudWatch 日志日志流的更多信息，请参阅 CloudWatch 用户指南 CloudWatch中的使用日志组和日志流。

CloudWatch 免费套餐允许在日志中发布最多 5Gb 的 CloudWatch 日志。超过免费套餐限额的日志将根据CloudWatch 定价计划收费。

主题

- 配置 CloudWatch 日志

配置 CloudWatch 日志

适用于 Prometheus 的亚马逊托管服务在亚马逊日志的日志组中记录警报管理器和统治者的错误和警告事件。 CloudWatch

您可以在适用于 Prometheus 的亚马逊托管服务控制台中设置 CloudWatch 日志记录配置，也可以通过调用 API 请求在中 AWS CLI 设置日志记录配置。`create-logging-configuration`

先决条件

在调用之前`create-logging-configuration`，请将以下策略或等效权限附加到您将用于配置 CloudWatch 日志的 ID 或角色。

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

配置 CloudWatch 日志

您可以使用控制台或 Prometheus 配置登录亚马逊托管服务。 AWS AWS CLI

Console

在 Amazon Managed Service for Prometheus 控制台中配置日志记录

1. 导航到工作区详细信息面板中的日志选项卡。
2. 选择日志面板右上角的管理日志。
3. 在日志级别下拉列表中选择全部。
4. 在日志组下拉列表中选择要向其发布日志的日志组。

您也可以在 CloudWatch 控制台中创建新的日志组。

5. 选择保存更改。

AWS CLI

您可以使用设置日志配置 AWS CLI。

要配置日志记录，请使用 AWS CLI

- 使用 AWS CLI，运行以下命令。

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

限制

- 并未记录所有事件

  Amazon Managed Service for Prometheus 仅记录处于 `warning` 或 `error` 级别的事件。
- 策略大小限制

  CloudWatch 日志资源策略限制为 5120 个字符。当 CloudWatch Logs 检测到策略接近此大小限制时，它会自动启用以开头的日志组`/aws/vendedlogs/`。

  当您创建启用日志记录的警报规则时，适用于 Prometheus 的 Amazon 托管服务必须使用您指定的日志 CloudWatch 组更新您的日志资源策略。为避免达到 CloudWatch 日志资源策略大小限制，请在日志 CloudWatch 日志组名称前加上`/aws/vendedlogs/`。在 Amazon Managed Service for Prometheus 控制台中创建日志组时，日志组名称的前缀为 `/aws/vendedlogs/`。有关更多信息，请参阅《日志用户指南》中的启用某些 AWS 服务的 CloudWatch 日志记录。

了解和优化成本

以下常见问题及其答案可能有助于了解和优化与 Amazon Managed Service for Prometheus 相关的成本。

哪些因素会增加我的成本？

对于大多数客户而言，指标摄取占据了大部分成本。查询使用率高的客户也会看到一些基于已处理的查询样本的成本，而指标存储仅占总成本的一小部分。有关每个的价格的更多信息，请参阅“Amazon Managed Service for Prometheus 产业页面”中的定价。

降低成本的最佳方法是什么？ 如何降低摄取成本？

摄取率（不是指标的存储）是大多数客户的主要成本。您可以通过降低收集频率（增加收集间隔）或减少摄入的活跃系列数量来降低摄取率。

您可以从收集代理增加收集（抓取）间隔：Prometheus 服务器（在代理模式下运行）和 Distro for (ADOT) 收集器都 OpenTelemetry 支持 AWS 该配置。`scrape_interval`例如，将收集间隔从 30 秒增加到 60 秒会将摄取使用量减少一半。

您也可以使用 `<relabel_config>` 筛选发送到 Amazon Managed Service for Prometheus 的指标。有关在 Prometheus 代理配置中重新标记的更多信息，请参阅 Prometheus 文档中的 https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config。

降低我的查询成本的最佳方法是什么？

查询成本基于已处理的样本数量。您可以降低查询频率来降低查询成本。

为了更清楚地了解对您查询成本影响最大的查询，您可以向您的支持联系人提交工单。Amazon Managed Service for Prometheus 团队可以帮助您了解对您的成本影响最大的查询。

如果我缩短指标的保留期，这会有助于减少我的账单总额吗？

您可以缩短保留期，但是，这不太可能大幅降低成本。

如果您想缩短（或延长）保留期，可以提交服务限制请求来更改 `Retention time for ingested data` 配额。

如何才能将警报查询费用保持在较低水平？

提醒会针对您的数据创建查询，这会增加您的查询成本。以下是一些可用于优化警报查询并降低成本的策略。

- 使用亚马逊托管服务进行 Prometheus 警报 — Amazon Prometheus 托管服务外部的警报系统可能需要额外查询才能增加弹性或高可用性，因为外部服务会从多个可用区或区域查询指标。这包括在 Grafana 中发出高可用性警报。这可能会使您的成本乘以三倍或更多。Amazon Prometheus 托管服务中的警报经过优化，可通过最少的查询次数为您提供高可用性和弹性。

  我们建议使用适用于 Prometheus 的亚马逊托管服务中的原生警报，而不是使用外部警报系统。
- 优化警报间隔-优化警报查询的一种快速方法是延长自动刷新间隔。如果您有一个每分钟查询一次，但每五分钟才需要一次的提醒，那么增加自动刷新间隔可以为您节省五倍的该警报的查询费用。
- 使用最佳回顾-查询中较大的回顾窗口会增加查询的成本，因为它会提取更多的数据。确保 PromQL 查询中的回顾窗口大小合理，可以容纳您需要提醒的数据。例如，在以下规则中，表达式包括十分钟的回顾窗口：

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

  将更改`expr`为`avg(rate(container_cpu_usage_seconds_total[5m])) > 0`有助于降低查询成本。

通常，请查看您的警报规则，并确保您针对服务的最佳指标发出警报。可以很容易地针对相同的指标创建重叠的警报，或者为你提供相同信息的多个警报，尤其是在你随着时间的推移添加警报时。如果您发现经常看到一组警报同时发生，则可以优化警报，而不必将所有警报都包括在内。

这些建议可以帮助您降低成本。归根结底，您必须在成本与创建正确的警报集以了解系统状态之间取得平衡。

有关在适用于 Prometheus 的亚马逊托管服务中提醒的更多信息，请参阅。警报管理器

我可以使用哪些指标来监控我的成本？

`IngestionRate`在 Amazon 中进行监控 CloudWatch 以跟踪您的摄取成本。有关监控 Amazon Prometheus 托管服务指标的更多信息，请参阅。 CloudWatch CloudWatch 指标

我可以随时查看账单吗？

它会 AWS 成本和使用情况报告 跟踪您的 AWS 使用情况，并提供账单周期内与您的账户相关的预估费用。有关更多信息，请参阅什么是 AWS 成本和使用情况报告？ 在《AWS 成本和使用情况报告用户指南》中

为什么我月初的账单高于月末？

Amazon Managed Service for Prometheus 采用分层定价模式，这会导致您的初始使用费用较高。当您的使用量达到更高的摄取等级时，如果费用较低，您的费用就会降低。有关定价（包括摄取等级）的更多信息，请参阅“Amazon Managed Service for Prometheus 产业页面”中的定价。

> **Note**
>
> - 等级适用于在区域内使用，而不是跨区域使用。一个区域内的使用量必须达到下一个等级，才能使用较低的费率。
> - 在中的组织中 AWS Organizations，等级使用是按付款人账户计算的，而不是按账户计算的（付款人账户始终是组织管理账户）。当组织中所有账户的摄取指标总量（在一个区域内）达到下一级别时，所有账户都将按较低的费率收费。

我删除了所有适用于 Prometheus 工作空间的亚马逊托管服务，但我似乎仍然需要付费。可能会发生什么？

在这种情况下，一种可能性是，您仍然有 AWS 托管抓取器，这些抓取器设置为将指标发送到已删除的工作区。按照说明进行操作查找和删除抓取程序。

与其他 AWS 服务集成

Amazon Managed Service for Prometheus 与其它 AWS 服务集成。本部分介绍如何与 Amazon Elastic Kubernetes Service（Amazon EKS）成本监控集成（具有 Kubecost），以及使用 Terraform 模块通过 AWS Observability Accelerator 为您的 EKS 项目创建完整的可观察性解决方案。

主题

与 Amazon EKS 成本监控集成

Amazon Managed Service for Prometheus 与 Amazon Elastic Kubernetes Service（Amazon EKS）成本监控（具有 Kubecost）集成，以执行成本分配计算并提供有关优化 Kubernetes 集群的洞察。将 Amazon Managed Service for Prometheus 与 Kubecost 配合使用，您可以可靠地扩展成本监控以支持更大的集群。

通过与 Kubecost 集成，您可以精细地了解您的 Amazon EKS 集群成本。您可以按大多数 Kubernetes 上下文汇总成本，从容器级别一直到集群级别，甚至是多集群级别。您可以出于记账或退款目的跨容器或集群生成报告来跟踪成本。

以下内容提供了关于在单集群或多集群场景中与 Kubecost 集成的说明：

- 单集群集成 – 要了解如何将 Amazon EKS 成本监控与单个集群集成，请参阅 AWS 博客文章 Integrating Kubecost with Amazon Managed Service for Prometheus。
- 多集群集成 – 要了解如何将 Amazon EKS 成本监控与多个集群集成，请参阅 AWS 博客文章 Multi-cluster cost monitoring for Amazon EKS using Kubecost and Amazon Managed Service for Prometheus。

> **Note**
>
> 有关使用 Kubecost 的更多信息，请参阅《Amazon EKS 用户指南》中的成本监控。

使用 AWS Observability Accelerator

AWS 可为您的 Amazon Elastic Kubernetes Service（Amazon EKS）项目提供可观察性工具，包括监控、日志记录、警报和控制面板。这包括 Amazon Managed Service for Prometheus、Amazon Managed Grafana、AWS Distro for OpenTelemetry 以及其它工具。为了方便您结合使用这些工具，AWS 提供了用于通过这些服务配置可观察性的 Terraform 模块，名为 AWS Observability Accelerator。

AWS Observability Accelerator 提供了监控基础设施、NGINX 部署和其它场景的示例。本部分举例说明了如何监控您 Amazon EKS 集群内的基础设施。

Terraform 模板和详细说明可以在 AWS Observability Accelerator for Terraform GitHub 页面上找到。您也可以阅读宣布推出 AWS Observability Accelerator 的博客文章。

先决条件

要使用 AWS Observability Accerator，您必须具有现有 Amazon EKS 集群并满足以下先决条件：

- AWS CLI – 用于从命令行调用 AWS 功能。
- kubectl – 用于从命令行控制您的 EKS 集群。
- Terraform – 用于自动为该解决方案创建资源。您必须使用有权在您的 AWS 账户中创建和管理 Amazon Managed Service for Prometheus、Amazon Managed Grafana 和 IAM 的 IAM 角色设置 AWS 提供商。有关如何为 Terraform 配置 AWS 提供商的更多信息，请参阅 Terraform 文档中的 AWS 提供商。

使用基础设施监控示例

AWS Observability Accelerator 提供了示例模板，这些模板使用随附的 Terraform 模块为您的 Amazon EKS 集群设置和配置可观察性。此示例演示如何使用 AWS Observability Accelerator 来设置基础设施监控。有关使用此模板及其包含的其它功能的更多详细信息，请参阅 GitHub 上的 Existing Cluster with the AWS Observability Accelerator base and Infrastructure monitoring 页面。

使用基础设施监控 Terraform 模块

1. 在要创建项目的文件夹中，使用以下命令克隆存储库。

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

2. 使用以下命令初始化 Terraform。

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

terraform init
```

3. 创建一个新 `terraform.tfvars` 文件，如以下示例所示。使用您 Amazon EKS 集群的 AWS 区域和集群 ID。

```
# (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. 如果还没有要使用的 Amazon Managed Grafana 工作区，请创建一个。有关如何创建新工作区的信息，请参阅《Amazon Managed Grafana 用户指南》中的创建您的首个工作区。

5. 在命令行中运行以下命令，为 Terraform 创建两个变量以使用您的 Grafana 工作区。您需要将 *grafana-workspace-id* 替换为 Grafana 工作区中的 ID。

```
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. [可选] 要使用现有的 Amazon Managed Service for Prometheus 工作区，请将 ID 添加到 `terraform.tfvars` 文件中，如以下示例所示，将 *prometheus-workspace-id* 替换为您的 Prometheus 工作区 ID。如果您未指定现有工作区，则将为您创建一个新的 Prometheus 工作区。

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

7. 使用以下命令部署解决方案。

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

这将在您的 AWS 账户中创建资源，包括以下内容：

- 一个新的 Amazon Managed Service for Prometheus 工作区（除非您选择使用现有工作区）。

- 您 Prometheus 工作区中的警报管理器配置、警报和规则。
- 您当前工作区中的全新 Amazon Managed Grafana 数据来源和控制面板。数据来源将命名为 `aws-observability-accelerator`。控制面板将列在 Observability Accelerator 控制面板下。
- 在提供的 Amazon EKS 集群中设置的 AWS Distro for OpenTelemetry Operator，用于向您的 Amazon Managed Service for Prometheus 工作区发送指标。

要查看您的新控制面板，请在您的 Amazon Managed Grafana 工作区中打开特定的控制面板。有关使用 Amazon Managed Grafana 的更多信息，请参阅《Amazon Managed Grafana 用户指南》中的使用 Grafana 工作区。

与 Kubernet AWS es 控制器集成

Amazon Managed Service for Prometheus 与 AWS Controllers for Kubernetes（ACK）集成，支持在 Amazon EKS 中管理您的工作区、警报管理器和规则器资源。你可以使用 Kubernetes 的 AWS 控制器自定义资源定义 (CRD) 和原生 Kubernetes 对象，而不必在集群之外定义任何资源。

本节介绍如何在现有亚马逊 EKS 集群中为 Kubernetes 设置 AWS 控制器和针对 Prometheus 的亚马逊托管服务。

您还可以阅读介绍适用于 Kubernetes 的 AWS 控制器和介绍适用于 Promethe us 的亚马逊托管服务的 ACK 控制器的博客文章。

先决条件

在开始将适用于 Kubernet AWS es 的控制器和适用于 Prometheus 的亚马逊托管服务与您的 Amazon EKS 集群集成之前，您必须具备以下先决条件。

- 您必须拥有现有角色 AWS 账户 和权限，才能以编程方式创建适用于 Prometheus 的亚马逊托管服务和 IAM 角色。
- 您必须拥有启用了 OpenID Connect（OIDC）的现有 Amazon EKS 集群。

  如果未启用 OIDC，则可以使用以下命令来启用。请记得用账户的相应值替换 *YOUR_CLUSTER_NAME* 和 *AWS_REGION*。

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

有关将 OIDC 与 Amazon EKS 配合使用的更多信息，请参阅《Amazon EKS 用户指南》中的 OIDC 身份提供者身份验证和创建 IAM OIDC 提供者。

- 您必须在 Amazon EKS 集群上安装了Amazon EBS CSI 驱动程序。
- 您必须已安装 AWS CLI。 AWS CLI 用于从命令行调用 AWS 功能。
- 必须安装 Kubernetes 的软件包管理器 Helm。
- 必须在 Amazon EKS 集群中设置 Prometheus 的控制面板指标。
- 您必须有 Amazon Simple Notification Service (Amazon SNS) 主题，以便从新工作区发送警报。请确保您已授予 Amazon Managed Service for Prometheus向该主题发送消息的权限。

正确配置您的 Amazon EKS 集群后，您应该能够通过调用 `kubectl get --raw /metrics` 查看针对 Prometheus 格式化的指标。现在，您可以为 Kubernetes 服务 AWS 控制器安装控制器，并使用它为 Prometheus 资源部署亚马逊托管服务。

使用适用于 Kubernetes 的 AWS 控制器部署工作空间

要部署适用于 Prometheus 的新亚马逊托管服务工作空间，您需要 AWS 为 Kubernetes 控制器安装控制器，然后使用它来创建工作空间。

部署适用于 Prometheus 的全新 Amazon 托管服务工作空间，其中包含适用于 Kubernetes 的控制器 AWS

1. 使用以下命令通过 Helm 安装 Amazon Managed Service for Prometheus 服务控制器。有关更多信息，请参阅在 Kubernetes 控制器 AWS 文档中安装 ACK 控制器。 GitHub为您的系统使用相应的*##*，例如 `us-east-1`。

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

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

稍等片刻，您应能够看到类似于以下内容的响应，表示成功。

```
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"
```

您可以选择使用以下 AWS 命令验证 Kubernetes 控制器的控制器是否已成功安装。

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

这将返回有关控制器 `ack-prometheusservice-controller` 的信息，包括 `status: deployed`。

2. 使用以下文本创建名为 `workspace.yaml` 的文件。这将用作您正在创建的工作区的配置。

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

3. 运行以下命令来创建您的工作区（此命令取决于您在步骤 1 中设置的系统变量）。

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

稍等片刻，您应该就能看到在您的账户中有一个名为 `my-amp-workspace` 的新工作区。

运行以下命令查看您工作区的详细信息和状态，包括工作区 ID。或者，您可以在 Amazon Managed Service for Prometheus 控制台中查看新的工作区。

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

> **Note**
>
> 您也可以使用现有工作区，而不是创建新的工作区。

4. 创建两个新的 yaml 文件作为规则组的配置，接下来您将使用以下配置创建 AlertManager 这两个文件。

   将此配置另存为 `rulegroup.yaml`。将 *WORKSPACE-ID* 替换为上一步中的工作区 ID。

```
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 }}"
```

   将以下配置另存为 `alertmanager.yaml`。将 *WORKSPACE-ID* 替换为上一步中的工作区 ID。*# TOPIC-AR N ########## Amazon SNS ### ARN#################* AWS 区域 记住，Amazon Managed Service for Prometheus 对 Amazon SNS 主题必须具有权限。

```
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**
>
> 要了解有关这些配置文件格式的更多信息，请参阅 RuleGroupsNamespaceData和 AlertManagerDefinitionData。

5. 运行以下命令来创建您的规则组和警报管理器配置（此命令取决于您在步骤 1 中设置的系统变量）。

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

稍等片刻，这些更改将会变得可用。

> **Note**
>
> 要更新资源，而不是创建资源，只需更新 yaml 文件，然后再次运行 `kubectl apply` 命令即可。
> 要删除资源，请运行以下命令。*ResourceType*替换为要删除的资源类型WorkspaceAlertManagerDefinition、或RuleGroupNamespace。*ResourceName*替换为要删除的资源的名称。

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

至此，新工作区的部署就完成了。下一部分将介绍如何配置您的集群以向该工作区发送指标。

配置您的 Amazon EKS 集群以写入 Amazon Managed Service for Prometheus 工作区

本部分介绍如何使用 Helm 配置在您的 Amazon EKS 集群中运行的 Prometheus，以便将指标远程写入您在上一部分中创建的 Amazon Managed Service for Prometheus 工作区。

在此过程中，您将需要自己创建的用于摄取指标的 IAM 角色的名称。如果尚未执行此操作，请参阅设置服务角色从 Amazon EKS 集群中摄取指标以获取更多信息和说明。如果您按照这些说明进行操作，IAM 角色将叫名为 `amp-iamproxy-ingest-role`。

配置 Amazon EKS 集群进行远程写入

1. 使用以下命令获取工作区的 `prometheusEndpoint`。将 *WORKSPACE-ID* 替换为上一部分中的工作区 ID。

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

   prometheusEndpoint 将出现在返回结果中，其格式如下所示：

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

   保存此 URL 以供后续步骤使用。

2. 创建一个包含以下文本的新文件，并将其命名为 `prometheus-config.yaml`。将 *account* 替换为您的账户 ID，将 *workspaceURL/* 替换为您刚才找到的 URL，将 *region* 替换为您系统的相应 AWS 区域 。

   ```
   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. 使用以下 Helm 命令查找 Prometheus 图表和命名空间名称以及图表版本。

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

根据到目前为止的步骤，Prometheus 图表和命名空间都应该名为 `prometheus`，图表版本可能为 `15.2.0`

4. 使用上一步中*PrometheusChartVersion*找到的*PrometheusChartName*、*PrometheusNamespace*和，运行以下命令。

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

几分钟后，您将看到一条消息，提示升级成功。

5. （可选）通过 `awscurl` 查询 Amazon Managed Service for Prometheus 终端节点，以此验证指标是否成功发送。将*##*替换为您正在使用的区域 AWS 区域 ，将 *WorkspaceURL/* 替换为您在步骤 1 中找到的网址。

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

现在，您已经创建了 Amazon Managed Service for Prometheus 工作区，并使用 YAML 文件作为配置从您的 Amazon EKS 集群与其连接。这些文件称为自定义资源定义（CRD），位于您的 Amazon EKS 集群中。您可以使用适用于 Kubernetes 的 AWS 控制器直接从集群管理所有适用于 Prometheus 的亚马逊托管服务资源。

将 CloudWatch 指标与 Firehose 集成

本节介绍如何检测亚马逊 CloudWatch 指标流并使用亚马逊数据 Fireh os e，以及如何将指标提取AWS Lambda到适用于 Prometheus 的亚马逊托管服务中。

您将使用AWS 云开发套件 (CDK) 设置堆栈来创建 Firehose Delivery Stream、Lambda 和 Amazon S3 存储桶，以演示完整的场景。

基础设施

首先，您必须为该配方设置基础设施。

CloudWatch 指标流允许将流式指标数据转发到 HTTP 终端节点或 Amazon S3 存储桶。

设置基础设施涵盖 4 个步骤：

- 配置先决条件
- 创建 Amazon Managed Service for Prometheus 工作区。
- 安装依赖项
- 部署堆栈

先决条件

- 已在您的环境中安装和配置。 AWS CLI
- 已在您的环境中安装 AWS CDK Typescript。
- 已在您的环境中安装 Node.js 和 Go。
- AWS 可观察性 CloudWatch 指标导出器 github 存储库 (`CWMetricsStreamExporter`) 已克隆到您的本地计算机上。

创建 Amazon Managed Service for Prometheus 工作区

1. 此配方中的演示应用程序将基于 Amazon Managed Service for Prometheus 运行。通过以下命令创建 Amazon Managed Service for Prometheus 工作区：

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

2. 使用以下命令确保您的工作区已创建：

```
aws amp list-workspaces
```

有关 Amazon Managed Service for Prometheus 的更多信息，请参阅 Amazon Managed Service for Prometheus 用户指南。

安装依赖项

1. 安装依赖项

   在 aws-o11y-recipes 存储库的根目录中，使用以下命令将您的目录更改为 CWMetricStreamExporter：

   ```
   cd sandbox/CWMetricStreamExporter
   ```

   今后，这将被视为存储库的根目录。

2. 通过以下命令将目录更改为 /cdk：

   ```
   cd cdk
   ```

3. 通过以下命令安装 CDK 依赖项。

   ```
   npm install
   ```

4. 将目录更改回存储库的根目录，然后使用以下命令将目录更改为 /lambda：

   ```
   cd lambda
   ```

5. 进入 /lambda 文件夹后，使用以下命令安装 Go 依赖项：

   ```
   go get
   ```

   所有依赖项现已安装完毕。

部署堆栈

1. 在存储库的根目录中，打开 `config.yaml`，将 `{workspace}` 替换为新创建的工作区 ID，将区域替换为您的 Amazon Managed Service for Prometheus 工作区所在的区域，从而修改 Amazon Managed Service for Prometheus 工作区 URL。

   例如，修改以下内容：

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

   根据自己的喜好更改 Firehose 传输流和 Amazon S3 存储桶的名称。

2. 要构建 AWS CDK 和 Lambda 代码，请在存储库的根目录中运行以下推荐：

   ```
   npm run build
   ```

   此构建步骤可确保构建 Go Lambda 二进制文件，并将 CDK 部署到。 CloudFormation

3. 要完成部署，请查看并接受堆栈所需的 IAM 更改。

4. （可选）可以运行以下命令确认堆栈是否已创建。

   ```
   aws cloudformation list-stacks
   ```

   名为 `CDK Stack` 的堆栈将出现在列表中。

创建 Amazon CloudWatch 直播

现在，您有了 lambda 函数来处理指标，可以从 Amazon CloudWatch 创建指标流。

创建 CloudWatch 指标流

1. 导航到 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/home#metric-streams:streamsList，然后选择创建指标流。

2. 选择所需的指标，可以是所有指标，也可以仅仅是选定命名空间中的指标。

3. 在 `Configuration` 下，选择选择您的账户拥有的现有 Firehose。

4. 您将使用之前通过 CDK 创建的 Firehose。在选择您的 Kinesis Data Firehose 流下拉列表中，选择之前创建的流。其名字将为 `CdkStack-KinesisFirehoseStream123456AB-sample1234`。
5. 将输出格式设置为 JSON。
6. 为指标流创建一个对您有意义的名称。
7. 选择 Create metric filter（创建指标流）。
8. （可选）要验证 Lambda 函数的调用，请导航到 Lambda 控制台并选择函数 `KinesisMessageHandler`。选择监控选项卡和 Logs 子选项卡，在最近调用下应该有所触发的 Lambda 函数的条目。

> **Note**
>
> 最长可能需要 5 分钟，调用才会开始显示在监控选项卡中。

您的指标正在从亚马逊 CloudWatch 流式传输到适用于 Prometheus 的亚马逊托管服务。

清理

您可能需要清除您在本示例中使用的资源。以下步骤将说明如何操作。这将停止您创建的指标流。

清理资源

1. 首先使用以下命令删除 CloudFormation 堆栈：

```
cd cdk
cdk destroy
```

2. 删除 Amazon Managed Service for Prometheus 工作区：

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

3. 最后，使用亚马逊 CloudWatch 控制台移除亚马逊 CloudWatch 指标流。

Amazon Managed Service for Prometheus 中的安全性

AWS 十分重视云安全性。为了满足对安全性最敏感的组织的需求，我们打造了具有超高安全性的数据中心和网络架构。作为 AWS 的客户，您也可以从这些数据中心和网络架构受益。

安全性是 AWS 和您的共同责任。责任共担模式将其描述为云的 安全性和云中 安全性：

- 云的安全性 - AWS 负责保护在 AWS 云中运行 AWS 服务的基础设施。AWS 还向您提供可安全使用的服务。第三方审核员定期测试和验证我们的安全性的有效性，作为 AWS 合规性计划的一部分。要了解适用于 Amazon Managed Service for Prometheus 的合规性计划，请参阅合规性计划范围内的 AWS 服务。
- 云中的安全性——您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括数据的敏感性、公司的要求以及适用的法律法规。

该文档有助于您了解如何在使用 Amazon Managed Service for Prometheus 时应用责任共担模式。以下主题说明如何配置 Amazon Managed Service for Prometheus 以实现您的安全性和合规性目标。您还会了解如何使用其它 AWS 服务来监控和保护 Amazon Managed Service for Prometheus 资源。

主题

Amazon Managed Service for Prometheus 的数据保护

AWS 分担责任模式分担责任模型适用于适用于 Prometheus 的亚马逊托管服务中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础设

施上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 AWS 责任共担模式和 GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与资源通信。 AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API 或软件开发工具包使用适用于 Prometheus 的亚马逊托管服务 AWS 服务 或其他服务时。 AWS CLI AWS 在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

主题

Amazon Managed Service for Prometheus 收集的数据

Amazon Managed Service for Prometheus 收集并存储您配置为从您账户中运行的 Prometheus 服务器发送到 Amazon Managed Service for Prometheus 的运行指标。该数据包括以下内容：

- 指标值
- 有助于识别和分类数据的指标标签（或任意键值对）

- 数据样本的时间戳

用来隔离来自不同客户的数据的唯一租户 ID 。这些 ID 限制了可访问的客户数据。客户无法更改租户 ID。

适用于 Prometheus 的亚马逊托管服务使用 () 密钥对其存储的数据进行加密。 AWS Key Management Service AWS KMS Amazon Managed Service for Prometheus 负责管理这些密钥。

> **Note**
>
> 适用于 Prometheus 的亚马逊托管服务支持创建用于加密数据的客户托管密钥。有关亚马逊 Prometheus 托管服务默认使用的密钥以及如何使用您自己的客户托管密钥的更多信息，请参阅。静态加密

传输中的数据将使用 HTTPS 自动加密。适用于 Prometheus 的亚马逊托管服务在内部使用 HTTPS 保护区域内可用区之间的连接。 AWS

静态加密

默认情况下，适用于 Prometheus 的亚马逊托管服务会自动为您提供静态加密，并使用自有的加密密钥执行此操作。 AWS

- AWS 自有密钥 — 适用于 Prometheus 的亚马逊 Prometheus 托管服务使用这些密钥自动加密上传到您的工作空间的数据。您无法查看、管理或使用 AWS 自有密钥，也无法审核其使用情况。但是，无需采取任何措施或更改任何计划即可保护用于加密数据的密钥。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的 AWS 自有密钥。

静态数据加密有助于减少保护敏感客户数据（例如个人身份信息）所需的运维开销和复杂性。其支持构建符合严格加密合规性和监管要求的安全应用程序。

您也可以选择在创建工作区时使用客户托管密钥：

- 客户托管密钥：Amazon Managed Service for Prometheus 支持使用您创建、拥有和管理的对称客户托管密钥，来加密工作区中的数据。由于您可以完全控制此加密，因此可以执行以下任务：
  - 制定和维护关键策略
  - 制定和维护 IAM policy 和授权
  - 启用和禁用密钥策略

- 轮换密钥加密材料
- 添加标签
- 创建密钥别名
- 计划删除密钥

有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的客户托管密钥。

谨慎选择是使用客户托管密钥还是 AWS 自有密钥。使用客户托管密钥创建的工作区以后无法转换为使用 AWS 自有密钥（反之亦然）。

> **Note**
>
> Amazon Prometheus 托管服务 AWS 使用自有密钥自动启用静态加密，从而免费保护您的数据。
> 但是，使用客户管理的密钥需要 AWS KMS 付费。有关定价的更多信息，请参阅 AWS Key Management Service 定价。

有关的更多信息 AWS KMS，请参阅什么是 AWS Key Management Service？

> **Note**
>
> 使用客户托管密钥创建的工作区不能使用 AWS 托管收集器进行摄取。

适用于 Prometheus 的亚马逊托管服务如何使用补助金 AWS KMS

Amazon Managed Service for Prometheus 需要三种授权才能使用客户托管密钥。

当您创建使用客户托管密钥加密的 Amazon Prometheus 托管服务工作空间时，适用于 Prometheus 的亚马逊托管服务通过向发送请求来代表您创建三项授权。CreateGrant AWS KMS中的授权 AWS KMS 用于授予适用于 Prometheus 的亚马逊托管服务访问您账户中的 KMS 密钥的权限，即使不是直接代表您调用（例如，存储从 Amazon EKS 集群中抓取的指标数据时）。

Amazon Managed Service for Prometheus 需要授权，才能将客户托管密钥用于以下内部操作：

- 向发送DescribeKey请求， AWS KMS 以验证创建工作空间时给出的对称客户托管 KMS 密钥是否有效。

- 向发送GenerateDataKey请求 AWS KMS 以生成由您的客户托管密钥加密的数据密钥。
- 将 Decrypt 请求发送 AWS KMS 到以解密加密的数据密钥，以便它们可用于加密您的数据。

适用于 Prometheus 的亚马逊托管服务为 AWS KMS 密钥创建了三项授权，允许适用于 Prometheus 的亚马逊托管服务代表您使用密钥。您可以通过更改密钥政策、禁用密钥或撤消授权来删除对密钥的访问权限。在执行这些操作之前，您应该了解这些操作的后果。这可能会导致工作区中的数据丢失。

如果您以任何方式删除对任何授权的访问权限，则 Amazon Managed Service for Prometheus 将无法访问由客户托管密钥加密的任何数据，也无法存储发送到工作区的新数据，这会影响依赖于该数据的操作。发送到工作区的新数据将无法访问，并且可能会永久丢失。

⚠ Warning

- 如果您禁用密钥，或者在密钥政策中删除了 Amazon Managed Service for Prometheus 访问权限，则无法再访问工作区数据。发送到工作区的新数据将无法访问，并且可能会永久丢失。

  通过还原对密钥的 Amazon Managed Service for Prometheus 访问权限，您可以访问工作区数据并重新开始接收新数据。
- 如果您撤销 授权，则无法重新创建该授权，并且工作区中的数据将永久丢失。

步骤 1：创建客户托管式密钥

您可以使用或 AWS KMS API 创建对称的客户托管密钥。 AWS Management Console只要您通过策略提供正确的访问权限，密钥就不必与 Amazon Managed Service for Prometheus 工作区位于同一个账户中，如下所述。

创建对称的客户托管密钥

按照《AWS Key Management Service 开发人员指南》中创建对称的客户托管密钥的步骤进行操作。

密钥策略

密钥策略控制对客户托管密钥的访问。每个客户托管密钥必须只有一个密钥策略，其中包含确定谁可以使用该密钥以及如何使用该密钥的声明。创建客户托管密钥时，可以指定密钥策略。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的管理对客户托管密钥的访问。

要将您的客户托管密钥用于 Amazon Managed Service for Prometheus 工作区，密钥政策中必须允许以下 API 操作：

- kms:CreateGrant – 添加客户托管密钥授权。授予对指定 KMS 密钥的控制访问权限，这允许访问 Amazon Managed Service for Prometheus 要求的授权操作。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的使用授权。

  这允许 Amazon Managed Service for Prometheus 执行以下操作：

  - 调用 GenerateDataKey 生成加密的数据密钥并将其存储，因为数据密钥不会立即用于加密。
  - 调用 Decrypt 来使用存储的加密数据密钥访问加密数据。
- kms:DescribeKey：提供客户托管密钥详细信息以允许 Amazon Managed Service for Prometheus 验证密钥。

以下是您可以为 Amazon Managed Service for Prometheus 添加的策略语句示例：

```
 "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>
 ]
```

- 有关在策略中指定权限的更多信息，请参阅《AWS Key Management Service 开发人员指南》。
- 有关密钥访问故障排除的信更多息，请参阅《AWS Key Management Service 开发人员指南》。

第 2 步：为适用于 Prometheus 的亚马逊托管服务指定客户托管密钥

创建工作区时，您可以通过输入 KMS 密钥 ARN 来指定客户托管密钥，Amazon Managed Service for Prometheus 使用该密钥来加密工作区存储的数据。

第 3 步：访问来自其他服务的数据，例如亚马逊托管 Grafana

此步骤是可选的，只有当您需要从其他服务访问亚马逊托管服务 Prometheus 数据时，才需要执行此步骤。

除非其他服务也有使用密 AWS KMS 钥的权限，否则无法从其他服务访问您的加密数据。例如，如果您想使用亚马逊托管 Grafana 来创建控制面板或提醒您的数据，则必须授予亚马逊托管 Grafana 访问密钥的权限。

让 Amazon Managed Grafana 访问您的客户托管密钥

1. 在您的亚马逊托管 Grafana 工作空间列表中，选择您想要访问适用于 Prometheus 的亚马逊托管服务的工作空间名称。这会向您显示有关您的亚马逊托管 Grafana 工作空间的摘要信息。
2. 记下您的工作空间使用的 IAM 角色的名称。名称的格式为`AmazonGrafanaServiceRole-<unique-id>`。控制台会显示该角色的完整 ARN。您将在稍后的步骤中在 AWS KMS 控制台中指定此名称。
3. 在您的AWS KMS 客户托管密钥列表中，选择您在创建适用于 Prometheus 的亚马逊托管服务工作区时使用的客户托管密钥。这将打开密钥配置详细信息页面。
4. 在 “关键用户” 旁边，选择 “添加” 按钮。

5. 从名称列表中，选择您在上面提到的亚马逊托管 Grafana IAM 角色。为了便于查找，您也可以按名称进行搜索。
6. 选择添加将 IAM 角色添加到密钥用户列表中。

您的亚马逊托管 Grafana 工作区现在可以访问适用于 Prometheus 的亚马逊托管服务工作区中的数据。您可以向关键用户添加其他用户或角色，以使其他服务能够访问您的工作空间。

Amazon Managed Service for Prometheus 加密上下文

加密上下文是一组可选的键值对，包含有关数据的其他上下文信息。

AWS KMS 使用加密上下文作为其他经过身份验证的数据来支持经过身份验证的加密。当您在加密数据的请求中包含加密上下文时，会将加密上下文 AWS KMS 绑定到加密数据。要解密数据，您必须在请求中包含相同的加密上下文。

Amazon Managed Service for Prometheus 加密上下文

适用于 Prometheus 的亚马逊托管服务在 AWS KMS 所有加密操作中使用相同的加密环境，其中密钥`aws:amp:arn`为，值为工作空间的亚马逊资源名称 (ARN)。

Example

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

使用加密上下文进行监控

使用对称的客户托管密钥来加密您的工作区数据时，您还可以使用审计记录和日志中的加密上下文来识别客户托管密钥的使用情况。加密上下文还会显示在AWS CloudTrail 或 Amazon Logs 生成的 CloudWatch 日志中。

使用加密上下文控制对客户托管式密钥的访问

您可以使用密钥策略和 IAM 策略中的加密上下文作为 `conditions` 来控制对您的对称客户托管密钥的访问。您还可以在授权中使用加密上下文约束。

Amazon Managed Service for Prometheus 在授权中使用加密上下文约束来控制对您账户或区域中客户托管密钥的访问。授权约束要求授权允许的操作使用指定的加密上下文。

Example

以下是密钥政策语句示例，用于授予对特定加密上下文的客户托管密钥的访问权限。此策略语句中的条件要求授权具有指定加密上下文的加密上下文约束。

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

监控 Amazon Managed Service for Prometheus 的加密密钥

当您在适用于 Prometheus 工作空间的亚马逊托管服务中使用 AWS KMS 客户托管密钥时，您可以使用AWS CloudTrail或 Amazon L CloudWatch ogs 来跟踪亚马逊 Prometheus 托管服务向其发送的请求。 AWS KMS

以下示例是`CreateGrant`、`GenerateDataKey`、和`DescribeKey`监控 KMS 操作 AWS CloudTrail 的事件`Decrypt`，这些操作由亚马逊托管服务调用，让 Prometheus 访问由您的客户托管密钥加密的数据：

CreateGrant

当您使用 AWS KMS 客户托管密钥加密工作空间时，适用于 Prometheus 的亚马逊托管服务会代表您发送`CreateGrant`三个访问您指定的 KMS 密钥的请求。Amazon Managed Service for Prometheus 创建的授权特定于与 AWS KMS 客户托管密钥关联的资源。

以下示例事件记录了 `CreateGrant` 操作：

```
{
    "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

当您为工作空间启用 AWS KMS 客户托管密钥时，适用于 Prometheus 的亚马逊托管服务会创建一个唯一的密钥。它向发送GenerateDataKey请求 AWS KMS ，指定资源的 AWS KMS客户托管密钥。

以下示例事件记录了 GenerateDataKey 操作：

```
{
    "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

在加密工作区上生成查询时，Amazon Managed Service for Prometheus 会调用 `Decrypt` 操作，以使用存储的加密数据密钥来访问加密数据。

以下示例事件记录了 `Decrypt` 操作：

```
{
    "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 Prometheus 使用 `DescribeKey` 操作，来验证账户和区域中是否存在与您的工作区关联的 AWS KMS 客户托管密钥。

以下示例事件记录了 `DescribeKey` 操作：

```
{
    "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"
}
```

了解更多信息

以下资源提供有关静态数据加密的更多信息。

- 有关 AWS Key Management Service 基本概念的更多信息，请参阅 《AWS Key Management Service 开发人员指南》。
- 有关安全最佳实践的更多信息 AWS Key Management Service，请参阅《AWS Key Management Service 开发人员指南》。

Amazon Managed Service for Prometheus 的身份和访问管理

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（具有权限）来使用 Amazon Managed Service for Prometheus 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在适用于 Prometheus 的亚马逊托管服务中所做的工作。

服务用户 – 如果您使用 Amazon Managed Service for Prometheus 服务来完成任务，则您的管理员会为您提供所需的凭证和权限。随着您使用更多 Amazon Managed Service for Prometheus 功能来完成工作，您可能需要额外权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 Amazon Managed Service for Prometheus 中的功能，请参阅Amazon Managed Service for Prometheus 身份和访问故障排除。

服务管理员 – 如果您在公司负责管理 Amazon Managed Service for Prometheus 资源，您可能对 Amazon Managed Service for Prometheus 具有完全访问权限。您有责任确定您的服务用户应访问哪些 Amazon Managed Service for Prometheus 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 Amazon Managed Service for Prometheus 搭配使用的更多信息，请参阅Amazon Managed Service for Prometheus 如何与 IAM 配合使用。

IAM 管理员 – 如果您是 IAM 管理员，您可能需要了解如何编写策略以管理对 Amazon Managed Service for Prometheus 的访问的详细信息。要查看您可在 IAM 中使用的 Amazon Managed Service for Prometheus 基于身份的策略示例，请参阅Amazon Managed Service for Prometheus 的基于身份的策略示例。

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的签署 AWS API 请求。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的 多重身份验证和《IAM 用户指南》中的 在 AWS中使用多重身份验证（MFA）。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的何时创建 IAM 用户（而不是角色）。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以 AWS Management Console 通过切换角色在中临时担任 IAM 角色。您可以通过调用 AWS CLI 或

AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的为第三方身份提供商创建角色。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的 权限集。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解角色和基于资源的跨账户访问策略之间的区别，请参阅 IA M 用户指南中的跨账户资源访问。
- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service (Amazon S3) 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM policy，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM policy。

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅 IAM 用户指南中的在托管式策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACL)

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。 AWS WAF要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的访问控制列表（ACL）概览。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。 AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 SCP 的工作原理。
- 会话策略 – 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

Amazon Managed Service for Prometheus 如何与 IAM 配合使用

在使用 IAM 管理针对 Amazon Managed Service for Prometheus 的访问权限之前，您应该了解哪些 IAM 功能可与 Amazon Managed Service for Prometheus 配合使用。

可与 Amazon Managed Service for Prometheus 配合使用的 IAM 功能

IAM 功能	Amazon Managed Service for Prometheus 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件密钥	否
ACL	否
ABAC（策略中的标签）	是
临时凭证	是
转发访问会话 (FAS)	否
服务角色	否
服务相关角色	是

要全面了解适用于 Prometheus 的亚马逊托管服务 AWS 和其他服务如何与大多数 IAM 功能配合使用，AWS 请参阅 IAM 用户指南中与 IAM 配合使用的服务。

Amazon Managed Service for Prometheus 的基于身份的策略

支持基于身份的策略	是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM policy。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Amazon Managed Service for Prometheus 的基于身份的策略示例

要查看 Amazon Managed Service for Prometheus 基于身份的策略示例，请参阅Amazon Managed Service for Prometheus 的基于身份的策略示例。

Amazon Managed Service for Prometheus 中基于资源的策略

支持基于资源的策略	否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户存取，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅 IAM 用户指南中的跨账户访问 IAM 中的资源。

Amazon Managed Service for Prometheus 的策略操作

支持策略操作	是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

要查看 Amazon Managed Service for Prometheus 操作的列表，请参阅《服务授权参考》中的 Amazon Managed Service for Prometheus 定义的操作。

Amazon Managed Service for Prometheus 中的策略操作在操作前面使用以下前缀：

```
aps
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

要查看 Amazon Managed Service for Prometheus 基于身份的策略示例，请参阅Amazon Managed Service for Prometheus 的基于身份的策略示例。

Amazon Managed Service for Prometheus 的策略资源

支持策略资源	是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称 (ARN) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

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

要查看 Amazon Managed Service for Prometheus 的资源类型及其 ARN 的列表，请参阅《服务授权参考》中的 Amazon Managed Service for Prometheus 定义的资源。要了解您可以在哪些操作中指定每个资源的 ARN，请参阅 Amazon Managed Service for Prometheus 定义的操作。

要查看 Amazon Managed Service for Prometheus 基于身份的策略示例，请参阅Amazon Managed Service for Prometheus 的基于身份的策略示例。

Amazon Managed Service for Prometheus 的策略条件键

支持特定于服务的策略条件密钥	否

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM policy 元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 Amazon Managed Service for Prometheus 条件键的列表，请参阅《服务授权参考》中的 Amazon Managed Service for Prometheus 的条件键。要了解您可以对哪些操作和资源使用条件键，请参阅 Amazon Managed Service for Prometheus 定义的操作。

要查看 Amazon Managed Service for Prometheus 基于身份的策略示例，请参阅Amazon Managed Service for Prometheus 的基于身份的策略示例。

Amazon Managed Service for Prometheus 中的访问控制列表（ACL）

支持 ACL	否

访问控制列表 (ACL) 控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

使用 Amazon Managed Service for Prometheus 的基于属性的访问权限控制（ABAC）

支持 ABAC（策略中的标签）	是

基于属性的访问控制 (ABAC) 是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以向 IAM 实体（用户或角色）和许多 AWS 资源附加标签。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/`*`key-name`*、`aws:RequestTag/`*`key-name`* 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息,请参阅《IAM 用户指南》中的什么是 ABAC？。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

将临时凭证与 Amazon Managed Service for Prometheus 结合使用

支持临时凭证	是

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的AWS 服务 与 IA M 配合使用的信息。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的 切换到角色（控制台）。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。 AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

Amazon Managed Service for Prometheus 的转发访问会话

支持转发访问会话（FAS）	否

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

Amazon Managed Service for Prometheus 的服务角色

支持服务角色	否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会破坏 Amazon Managed Service for Prometheus 的功能。仅当 Amazon Managed Service for Prometheus 提供相关指导时才编辑服务角色。

Amazon Managed Service for Prometheus 的服务相关角色

支持服务相关角色	是

服务相关角色是一种链接到的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理 Amazon Managed Service for Prometheus 服务相关角色的详细信息，请参阅使用 Amazon Managed Service for Prometheus 的服务相关角色。

Amazon Managed Service for Prometheus 的基于身份的策略示例

默认情况下，用户和角色没有创建或修改 Amazon Managed Service for Prometheus 资源的权限。他们也无法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM policy。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅 IAM 用户指南中的 创建 IAM 策略。

有关 Amazon Managed Service for Prometheus 定义的操作和资源类型的详细信息，包括每种资源类型的 ARN 格式，请参阅《服务授权参考》中的 Amazon Managed Service for Prometheus 的操作、资源和条件键。

主题

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon Managed Service for Prometheus 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管策略 或 工作职能的AWS 托管策略。
- 应用最低权限 – 在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定 AWS 服务的（例如）使用的，则也可以使用条件来授予对服务操作的访问权限 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件 。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM policy语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Acess Analyzer 策略验证。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实操。

使用 Amazon Managed Service for Prometheus 控制台

要访问 Amazon Managed Service for Prometheus 控制台，您必须具有一组最低的权限。这些权限必须允许您列出和查看有关您的 AWS 账户中的 Amazon Managed Service for Prometheus 资源的详细信息。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用适用于 Prometheus 的亚马逊托管服务控制台，还要将适用于 Prometheus 的亚马逊托管服务或托管策略附加到这些实ConsoleAccess体。ReadOnly AWS 有关更多信息，请参阅《IAM 用户指南》中的为用户添加权限。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

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

AWS 适用于 Prometheus 的亚马逊托管服务的托管政策

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AmazonPrometheusFullAccess

您可以将 `AmazonPrometheusFullAccess` 策略附加到 IAM 身份。

权限详细信息

该策略包含以下权限。

- `aps`：允许完整访问 Amazon Managed Service for Prometheus
- `eks`：允许 Amazon Managed Service for Prometheus 服务读取有关 Amazon EKS 集群的信息。这是允许创建托管抓取程序并在集群中发现指标所必需的。
- `ec2`：允许 Amazon Managed Service for Prometheus 服务读取有关 Amazon EC2 网络的信息。这是允许创建可访问您 Amazon EKS 指标的托管抓取程序所必需的。

- iam：允许主体为托管指标抓取程序创建服务相关角色。

的内容AmazonPrometheusFullAccess如下：

```
{
 "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

您可以将 AmazonPrometheusConsoleFullAccess 策略附加到 IAM 身份。

权限详细信息

该策略包含以下权限。

- aps：允许完整访问 Amazon Managed Service for Prometheus
- tag：允许主体在 Amazon Managed Service for Prometheus 控制台中查看标签建议。

```
{
 "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

的内容AmazonPrometheusRemoteWriteAccess如下：

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

AmazonPrometheusQueryAccess

的内容AmazonPrometheusQueryAccess如下：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aps:GetLabels",
                "aps:GetMetricMetadata",
```

```
                "aps:GetSeries",
                "aps:QueryMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略： AmazonPrometheusScraperServiceRolePolicy

您无法附加 AmazonPrometheusScraperServiceRolePolicy 到您的 IAM 实体。此附加到服务相关角色的策略允许 Amazon Managed Service for Prometheus 代表您执行操作。有关更多信息，请参阅 使用角色从 EKS 中抓取指标。

此策略向贡献者授予权限，以允许从您的 Amazon EKS 集群读取和写入到 Amazon Managed Service for Prometheus 工作区。

> **Note**
>
> 此用户指南以前错误地称之为此政策 `AmazonPrometheusScraperServiceLinkedRolePolicy`

权限详细信息

该策略包含以下权限。

- `aps`：允许服务主体将指标写入 Amazon Managed Service for Prometheus 工作区。
- `ec2`：允许服务主体读取和修改网络配置，以连接到包含您的 Amazon EKS 集群的网络。
- `eks`：允许服务主体访问您的 Amazon EKS 集群。这是必需的，这样它才能自动抓取指标。还允许委托人在移除抓取器时清理 Amazon EKS 资源。

```
{
 "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:aws: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:aws:eks:*:*:cluster/*"
   },
   {
    "Sid": "DeleteEKSAccessEntry",
    "Effect": "Allow",
    "Action": "eks:DeleteAccessEntry",
    "Resource": "arn:aws:eks:*:*:access-entry/*/role/*",
    "Condition": {
     "StringEquals": {
      "aws:PrincipalAccount": "${aws:ResourceAccount}"
     },
     "ArnLike": {
      "eks:principalArn": "arn:aws:iam::*:role/aws-service-role/
scraper.aps.amazonaws.com/AWSServiceRoleForAmazonPrometheusScraper*"
     }
    }
   },
   {
    "Sid": "APSWriting",
    "Effect": "Allow",
    "Action": "aps:RemoteWrite",
    "Resource": "arn:aws:aps:*:*:workspace/*",
    "Condition": {
     "StringEquals": {
      "aws:PrincipalAccount": "${aws:ResourceAccount}"
     }
    }
   }
```

```
 ]
}
```

适用于 Prometheus 的亚马逊托管服务更新了托管政策 AWS

查看有关自亚马逊 AWS 托管服务 Prometheus 托管政策开始跟踪这些更改以来该服务更新的详细信息。有关此页面更改的自动提示，请订阅 Amazon Managed Service for Prometheus 文档历史记录页面上的 RSS 源。

更改	描述	日期
AmazonPrometheusScraperServiceRolePolicy – 对现有策略的更新	适用于 Prometheus 的亚马逊托管服务增加了新的权限，AmazonPrometheusScraperServiceRolePolicy以支持使用亚马逊 EKS 中的访问条目。 包括管理 Amazon EKS 访问条目的权限，允许在删除抓取器时清理资源。 **Note** 之前的用户指南错误地称之为此政策 `AmazonPrometheusScraperServiceLinkedRolePolicy`	2024年5月2日
AmazonPrometheusFullAccess – 更新了现有策略	Amazon Managed Service for Prometheus 在 AmazonPrometheusFullAccess 中添加了新的权限，以支持为 Amazon	2023 年 11 月 26 日

更改	描述	日期
	EKS 集群中的指标创建托管抓取程序。 包括连接到 Amazon EKS 集群、读取 Amazon EC2 网络以及为抓取程序创建服务相关角色的权限。	
AmazonPrometheusScraperServiceLinkedRolePolicy：新策略	Amazon Managed Service for Prometheus 添加了一项新的服务相关角色策略，用于从 Amazon EKS 容器中读取，以允许自动抓取指标。 包括连接到 Amazon EKS 集群、读取 Amazon EC2 网络、创建和删除标记为 `AMPAgentlessScraper` 的网络的权限，以及写入 Amazon Managed Service for Prometheus 工作区的权限。	2023 年 11 月 26 日

更改	描述	日期
AmazonPrometheusConsoleFullAccess – 更新了现有策略	适用于 Prometheus 的亚马逊托管服务为支持在日志中记录警报管理器和标尺事件添加了新的权限AmazonPrometheusConsoleFullAccess。CloudWatch 添加了 `aps:CreateLoggingConfiguration` 、`aps:UpdateLoggingConfiguration` 、`aps:DeleteLoggingConfiguration` 、`aps:DescribeLoggingConfiguration` 权限。	2022 年 10 月 24 日

更改	描述	日期
AmazonPrometheusConsoleFullAccess – 更新了现有策略	Amazon Managed Service for Prometheus 向 AmazonPrometheusConsoleFullAccess 添加了新的权限，以支持新的 Amazon Managed Service for Prometheus 功能，因此，使用此策略的用户在将标签应用于 Amazon Managed Service for Prometheus 资源时可以看到标签建议列表。 添加了 `tag:GetTagKeys` 、`tag:GetTagValues` 、`aps:CreateAlertManagerDefinition` 、`aps:CreateRuleGroupsNamespace` 、`aps:DeleteAlertManagerDefinition` 、`aps:DeleteRuleGroupsNamespace` 、`aps:DescribeAlertManagerDefinition` 、`aps:DescribeRuleGroupsNamespace` 、`aps:ListRuleGroupsNamespaces` 、`aps:PutAlertManagerDefinition` 、`aps:PutRuleGroupsNamespace` 、`aps:TagResource` 和 `aps:UntagResource` 权限。	2021 年 9 月 29 日

更改	描述	日期
Amazon Managed Service for Prometheus 开始跟踪更改	适用于 Prometheus 的亚马逊托管服务已开始跟踪其托管政策的变更。 AWS	2021 年 9 月 15 日

Amazon Managed Service for Prometheus 身份和访问故障排除

您可以使用以下信息，来诊断和修复在使用 Amazon Managed Service for Prometheus 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 Amazon Managed Service for Prometheus 中执行操作
- 我无权执行 iam：PassRole
- 我想允许 AWS 账户之外的用户访问我的 Prometheus 亚马逊托管服务资源

我无权在 Amazon Managed Service for Prometheus 中执行操作

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当 `mateojackson` IAM 用户尝试使用控制台查看有关虚构 *`my-example-widget`* 资源的详细信息，但不拥有虚构 `aps:`*`GetWidget`* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aps:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 `mateojackson` 用户的策略，以允许使用 `aps:`*`GetWidget`* 操作访问 *`my-example-widget`* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一个错误，指明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 Amazon Managed Service for Prometheus。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 Amazon Managed Service for Prometheus 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许 AWS 账户之外的用户访问我的 Prometheus 亚马逊托管服务资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 Amazon Managed Service for Prometheus 是否支持这些功能，请参阅Amazon Managed Service for Prometheus 如何与 IAM 配合使用。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过联合身份验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（联合身份验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问的区别，请参阅 IA M 用户指南中的跨账户资源访问。

IAM 权限和策略

访问 Amazon Managed Service for Prometheus 操作和数据需要凭证。这些凭证必须有权执行操作和访问 AWS 资源，例如检索关于您的云资源的 Amazon Managed Service for Prometheus 数据。以下部分提供详细信息来说明如何使用 AWS Identity and Access Management（IAM）和 Amazon Managed Service for Prometheus 控制谁能访问您的资源，从而对这些资源进行保护。有关更多信息，请参阅 IAM 中的策略和权限。

Amazon Managed Service for Prometheus 权限

下表显示了可能的 Amazon Managed Service for Prometheus 操作及其所需的权限。这些操作可能还需要其它服务的权限，此处未详细介绍。

操作	所需的权限
创建警报。	`aps:CreateAlertManagerAlerts`
在工作区创建警报管理器定义。有关更多信息，请参阅警报管理器。	`aps:CreateAlertManagerDefinition`
在工作区创建规则组命名空间。有关更多信息，请参阅记录规则和警报规则。	`aps:CreateRuleGroupsNamespace`
创建 Amazon Managed Service for Prometheus 工作区。工作区是专用于存储和查询 Prometheus 指标的逻辑空间。	`aps:CreateWorkspace`
从工作区删除警报管理器定义。	`aps:DeleteAlertManagerDefinition`
删除警报静默。	`aps:DeleteAlertManagerSilence`
删除 Amazon Managed Service for Prometheus 工作区。	`aps:DeleteWorkspace`
检索有关警报管理器定义的详细信息。	`aps:DescribeAlertManagerDefinition`
检索有关规则组命名空间的详细信息。	`aps:DescribeRuleGroupsNamespace`
检索有关 Amazon Managed Service for Prometheus 工作区的详细信息。	`aps:DescribeWorkspace`
检索有关警报静默的详细信息。	`aps:GetAlertManagerSilence`
检索工作区中警报管理器的状态。	`aps:GetAlertManagerStatus`
检索标签。	`aps:GetLabels`

操作	所需的权限
检索 Amazon Managed Service for Prometheus 指标的元数据。	aps:GetMetricMetadata
检索时间序列数据。	aps:GetSeries
检索警报管理器定义中定义的警报组列表。	aps:ListAlertManagerAlertGroups
检索警报管理器中定义的警报列表。	aps:ListAlertManagerAlerts
检索警报管理器定义中定义的接收方列表。	aps:ListAlertManagerReceivers
检索已定义的警报静默列表。	aps:ListAlertManagerSilences
检索活动警报列表。	aps:ListAlerts
检索工作区中规则组命名空间中的规则列表。	aps:ListRules
检索工作区中规则组命名空间的列表。	aps:ListRuleGroupsNamespaces
检索与 Amazon Managed Service for Prometheus 资源关联的标签。	aps:ListTagsForResource
检索您账户中存在的 Amazon Managed Service for Prometheus 工作区的列表。	aps:ListWorkspaces
更新工作区中现有的警报管理器定义。	aps:PutAlertManagerDefinition
创建警报静默。	aps:PutAlertManagerSilences
更新现有规则组命名空间。	aps:PutRuleGroupsNamespace
对 Amazon Managed Service for Prometheus 指标运行查询。	aps:QueryMetrics

操作	所需的权限
执行远程写入操作以启动将指标从 Prometheus 服务器流式传输到 Amazon Managed Service for Prometheus。	aps:RemoteWrite
为 Amazon Managed Service for Prometheus 资源分配标签。	aps:TagResource
删除 Amazon Managed Service for Prometheus 资源中的标签。	aps:UntagResource
修改现有工作区的别名。	aps:UpdateWorkspaceAlias
创建日志记录配置。	aps:CreateLoggingConfiguration
删除日志记录配置	aps:DeleteLoggingConfiguration
描述工作区日志记录配置。	aps:DescribeLoggingConfiguration
更新日志记录配置。	aps:UpdateLoggingConfiguration

示例 IAM 策略

本部分提供了您可以创建的其它自行管理策略的示例。

以下 IAM 策略授予对 Amazon Managed Service for Prometheus 的完全访问权限，还支持用户发现 Amazon EKS 集群并查看有关集群的详细信息。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:*",
                "eks:DescribeCluster",
                "eks:ListClusters"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Amazon Managed Service for Prometheus 的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在这些基础上 AWS 部署以安全性和合规性为重点的基准环境的步骤。
- 在 A@@ mazon Web Services 上构建 HIPAA 安全与合规架构 — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅符合 HIPAA 要求的服务参考。

- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 Security Hub 控件参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务检测您的工作负载、容器和数据面临的潜在威胁。 GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。

- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

Amazon Managed Service for Prometheus 中的数据恢复

AWS全球基础设施围绕 AWS 区域和可用区构建。AWS区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅 AWS全球基础设施。

除了 AWS 全球基础设施之外，Amazon Managed Service for Prometheus 还提供多种功能，以协助支持您的数据恢复和备份需求，包括支持高可用性数据。

Amazon Managed Service for Prometheus 的基础设施安全性

作为一项托管式服务，Amazon Managed Service for Prometheus 受 AWS 全球网络安全保护。有关 AWS 安全服务以及 AWS 如何保护基础设施的信息，请参阅 AWS 云安全。要按照基础设施安全最佳实践设计您的 AWS 环境，请参阅《安全性支柱 AWS Well-Architected Framework》中的基础设施保护。

您可以使用 AWS 发布的 API 调用通过网络访问 Amazon Managed Service for Prometheus。客户端必须支持以下内容：

- 传输层安全性协议 (TLS) 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用AWS Security Token Service (AWS STS) 生成临时安全凭证来对请求进行签名。

使用 Amazon Managed Service for Prometheus 的服务相关角色

适用于 Prometheus 的亚马逊托管服务 AWS Identity and Access Management 使用 (IAM) 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon Managed Service for Prometheus 直接

相关。服务相关角色由 Amazon Managed Service for Prometheus 预定义，并包含服务代表您调用其他 AWS 服务所需的所有权限。

服务相关角色可让您更轻松地设置 Amazon Managed Service for Prometheus，因为您不必手动添加必要的权限。Amazon Managed Service for Prometheus 定义其服务相关角色的权限，除非另有定义，否则仅 Amazon Managed Service for Prometheus 可以代入该角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其它 IAM 实体。

使用角色从 EKS 中抓取指标

使用适用于 Prometheus 的亚马逊托管服务托管收集器自动抓取指标时，AWSServiceRoleForAmazonPrometheusScraper服务相关角色用于简化托管收集器的设置，因为您不必手动添加必要的权限。Amazon Managed Service for Prometheus 定义权限，且仅 Amazon Managed Service for Prometheus 可以代入该角色。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的AWS 服务，并查找服务相关角色列中显示为是的服务。选择是和链接，查看该服务的服务相关角色文档。

Amazon Managed Service for Prometheus 的服务相关角色权限

适用于 Prometheus 的亚马逊托管服务使用AWSServiceRoleForAmazonPrometheusScraper以前缀命名的服务相关角色允许适用于 Prometheus 的亚马逊托管服务自动抓取您的亚马逊 EKS 集群中的指标。

AWSServiceRoleForAmazonPrometheusScraper 服务相关角色信任以下服务来代入该角色：

- `scraper.aps.amazonaws.com`

名为的角色权限策略AmazonPrometheusScraperServiceRolePolicy允许适用于 Prometheus 的亚马逊托管服务对指定资源完成以下操作：

- 准备好并修改网络配置，以连接到包含您的 Amazon EKS 集群的网络。
- 从 Amazon EKS 集群中读取指标，并将指标写入 Amazon Managed Service for Prometheus 工作区。

您必须配置允许用户、组或角色创建服务相关角色的权限。有关更多信息，请参阅《IAM 用户指南》中的服务相关角色权限。

为 Amazon Managed Service for Prometheus 创建服务相关角色

您无需手动创建服务相关角色。当您在、或 AWS API 中使用亚马逊 EKS 或亚马逊 Prometheus 托管服务创建托管收集器实例时，AWS CLI适用于 Prometheus 的亚马逊托管服务会为您创建服务相关角色。 AWS Management Console

> **Important**
>
> 如果您在其他使用此角色支持的功能的服务中完成某个操作，此服务相关角色可以出现在您的账户中。要了解更多信息，请参阅 “我的” 中出现了一个新角色 AWS 账户。

如果您删除该服务相关角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您使用 Amazon EKS 或 Amazon Managed Service for Prometheus 创建托管收集器实例时，Amazon Managed Service for Prometheus 会再次为您创建服务相关角色。

编辑 Amazon Managed Service for Prometheus 的服务相关角色

适用于 Prometheus 的亚马逊托管服务不允许您编辑服务相关角色。AWSServiceRoleForAmazonPrometheusScraper 创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除 Amazon Managed Service for Prometheus 的服务相关角色

您无需手动删除该 AWSServiceRoleForAmazonPrometheusScraper 角色。当您删除与 AWS Management Console、或 AWS API 中的角色关联的所有托管收集器实例时 AWS CLI，适用于 Prometheus 的亚马逊托管服务会清理资源并为您删除服务相关角色。

Amazon Managed Service for Prometheus 服务相关角色支持的区域

Amazon Managed Service for Prometheus 支持在所有服务可用区域中使用服务相关角色。有关更多信息，请参阅 支持的区域。

使用 AWS CloudTrail 记录 Amazon Managed Service for Prometheus API 调用

适用于 Prometheus 的亚马逊托管服务 AWS CloudTrail与一项服务集成，可记录用户、角色或服务在 Prometheus 的亚马逊托管 AWS 服务中采取的操作。 CloudTrail 将适用于 Prometheus 的亚马逊

Prometheus 托管服务的所有 API 调用捕获为事件。捕获的调用包括来自 Amazon Managed Service for Prometheus 控制台的调用和对 Amazon Managed Service for Prometheus API 操作的代码调用。如果您创建跟踪，则可以将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括适用于 Prometheus 的亚马逊托管服务的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的事件历史记录中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向亚马逊 Prometheus 托管服务发出的请求、发出请求的 IP 地址、谁提出请求、何时提出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅AWS CloudTrail 用户指南。

适用于 Prometheus 的亚马逊托管服务信息 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当 Amazon Prometheus 托管服务中发生活动时，该活动会 AWS 与其他服务事件一起记录在 CloudTrail 事件历史记录中。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅使用事件历史记录查看 CloudTrail 事件。

要持续记录您的 AWS 账户中的事件，包括适用于 Prometheus 的亚马逊托管服务的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，当您在控制台中创建跟踪时，该跟踪将应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传输到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析 CloudTrail 日志中收集的事件数据并对其采取行动。有关更多信息，请参阅下列内容：

- 创建跟踪记录概览
- CloudTrail 支持的服务和集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

Amazon Managed Service for Prometheus 支持记录以下操作：

- CreateAlertManagerAlerts
- CreateAlertManagerDefinition
- CreateRuleGroupsNamespace
- CreateWorkspace
- DeleteAlertManagerDefinition
- DeleteAlertManagerSilence
- DeleteWorkspace

- DeleteRuleGroupsNamespace
- DescribeAlertManagerDefinition
- DescribeRulesGroupsNamespace
- DescribeWorkspace
- ListRuleGroupsNamespaces
- ListWorkspaces
- PutAlertManagerDefinition
- PutAlertManagerSilences
- PutRuleGroupsNamespace
- UpdateWorkspaceAlias

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根证书还是 AWS Identity and Access Management (IAM) 用户凭证发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

了解 Amazon Managed Service for Prometheus 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序出现。

示例： CreateWorkspace

以下示例显示了演示该 CreateWorkspace操作的 CloudTrail 日志条目。

```
{
    "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"
}
```

示例： CreateAlertManagerDefinition

以下示例显示了演示该 CreateAlertManagerDefinition 操作的 CloudTrail 日志条目。

```
{
    "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"
}
```

示例： CreateRuleGroupsNamespace

以下示例显示了演示该 CreateRuleGroupsNamespace 操作的 CloudTrail 日志条目。

```
{
    "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"
}
```


设置服务账户的 IAM 角色

通过服务账户的 IAM 角色，您可以将 IAM 角色与 Kubernetes 服务账户关联。然后，此服务账户可向使用它的任意 Pod 中的容器提供 AWS 权限。有关更多信息，请参阅服务账户的 IAM 角色。

服务账户的 IAM 角色也称为服务角色。

在 Amazon Managed Service for Prometheus 中，使用服务角色有助于您获取在 Amazon Managed Service for Prometheus、Prometheus 服务器和 Grafana 服务器之间进行授权和身份验证所需的角色。

先决条件

本页上的步骤要求您安装 AWS CLI 和 EKSCTL 命令行界面。

设置服务角色从 Amazon EKS 集群中摄取指标

要设置服务角色以使 Amazon Managed Service for Prometheus 能够从 Amazon EKS 集群中的 Prometheus 服务器摄取指标，您必须登录到具有以下权限的账户：

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

设置服务角色以摄取到 Amazon Managed Service for Prometheus

1. 使用以下内容创建名为 `createIRSA-AMPIngest.sh` 的文件。
   将 <my_amazon_eks_clustername> 替换为您集群的名称，并将 <my_prometheus_namespace> 替换为您的 Prometheus 命名空间。

```
#!/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. 要赋予脚本必要的权限，请输入以下命令。

```
chmod +x createIRSA-AMPIngest.sh
```

3. 运行 脚本。

设置服务账户的 IAM 角色以查询指标

要为服务账户设置 IAM 角色（服务角色）以便从 Amazon Managed Service for Prometheus 工作区查询指标，您必须登录到具有以下权限的账户：

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

设置服务角色以查询 Amazon Managed Service for Prometheus 指标：

1. 使用以下内容创建名为 `createIRSA-AMPQuery.sh` 的文件。将 <my_amazon_eks_clustername> 替换为集群的名称，并将 <my_prometheus_namespace> 替换为您的 Prometheus 命名空间。

```
#!/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. 要赋予脚本必要的权限，请输入以下命令。

```
chmod +x createIRSA-AMPQuery.sh
```

3. 运行 脚本。

将 Amazon Managed Service for Prometheus 与接口 VPC 终端节点结合使用

如果您使用 Amazon Virtual Private Cloud（Amazon VPC）托管 AWS 资源，则可以在您的 VPC 和 Amazon Managed Service for Prometheus 之间建立专有连接。您可以使用这些连接让 Amazon Managed Service for Prometheus 与您的 VPC 上的资源之间进行通信，而不用访问公共 Internet。

Amazon VPC 是一项 AWS 服务，可用来启动在虚拟网络中定义的 AWS 资源。借助 VPC，您可以控制您的网络设置，如 IP 地址范围、子网、路由表和网络网关。要将您的 VPC 连接到 Amazon Managed Service for Prometheus，您需要定义一个接口 VPC 终端节点来将您的 VPC 连接到 AWS 服务。该终端节点提供了到 Amazon Managed Service for Prometheus 的可靠、可扩展的连接，无需 Internet 网关、网络地址转换（NAT）实例或 VPN 连接。有关更多信息，请参阅《Amazon VPC 用户指南》中的什么是 Amazon VPC。

接口 VPC 终端节点由 AWS PrivateLink 提供支持，后者是一种 AWS 技术，可将弹性网络接口与私有 IP 地址结合使用来支持 AWS 服务之间的专有通信。有关更多信息，请参阅新增 – 适用于 AWS 服务的 AWS PrivateLink 博客文章。

以下信息面向的是 Amazon VPC 用户。有关如何开始使用 Amazon VPC 的更多信息，请参阅《Amazon VPC 用户指南》中的开始使用。

为 Amazon Managed Service for Prometheus 创建接口 VPC 终端节点

创建接口 VPC 终端节点以开始使用 Amazon Managed Service for Prometheus。从以下服务名称终端节点中进行选择：

- `com.amazonaws.region.aps-workspaces`

  选择此服务名称即可使用与 Prometheus 兼容的 API。有关更多信息，请参阅《Amazon Managed Service for Prometheus 用户指南》中的与 Prometheus 兼容的 API。

- `com.amazonaws.region.aps`

  选择此服务名称来执行工作区管理任务。有关更多信息，请参阅《Amazon Managed Service for Prometheus 用户指南》中的 Amazon Managed Service for Prometheus API。

> **Note**
>
> 如果您在无法直接访问 Internet 的 VPC 中使用 remote_write，则还必须为 AWS Security Token Service 创建接口 VPC 终端节点，以便 sigv4 可用于该终端节点。有关为 AWS STS 创建 VPC 终端节点的更多信息，请参阅《AWS Identity and Access Management 用户指南》中的使用 AWS STS 接口 VPC 终端节点。您必须将 AWS STS 设置为使用区域化终端节点。

有关更多信息，包括创建接口 VPC 终端节点的分步说明，请参阅《Amazon VPC 用户指南》中的创建接口终端节点。

> **Note**
>
> 您可以使用 VPC 终端节点策略来控制对 Amazon Managed Service for Prometheus 接口 VPC 终端节点的访问。有关更多信息，请参见下一节。

如果为 Amazon Managed Service for Prometheus 创建接口 VPC 终端节点，并且您已有流向 VPC 上的工作区的数据，默认情况下，指标将流过该接口 VPC 终端节点。Amazon Managed Service for Prometheus 使用公共终端节点或私有接口终端节点（以正在使用的终端节点为准）来执行此任务。

控制对 Amazon Managed Service for Prometheus VPC 终端节点的访问

您可以使用 VPC 终端节点策略来控制对 Amazon Managed Service for Prometheus 接口 VPC 终端节点的访问。VPC 端点策略是一种 IAM 资源策略，您在创建或修改端点时可将它附加到端点。如果您在

创建端点时未附加策略，Amazon VPC 会为您附加一个默认策略，该策略允许对服务的完全访问。终端节点策略不会覆盖或替换 IAM 基于身份的策略或服务特定的策略。这是一个单独的策略，用于控制从端点中对指定服务进行的访问。

有关更多信息，请参阅《Amazon VPC 用户指南》中的使用 VPC 端点控制对服务的访问权限。

下面是用于 Amazon Managed Service for Prometheus 的终端节点策略示例。该策略允许具有 PromUser 角色的用户通过 VPC 连接到 Amazon Managed Service for Prometheus 来查看工作区和规则组，但不能创建或删除工作区。

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

以下示例显示的策略仅允许来自指定 VPC 中指定 IP 地址的请求成功。来自其它 IP 地址的请求将失败。

```
{
    "Statement": [
        {
            "Action": "aps:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
```

```
            "Condition": {
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
}
```

故障排除

利用以下部分来协助排查使用 Amazon Managed Service for Prometheus 时遇到的问题。

主题

429 或超出限制的错误

如果您看到类似于以下示例的 429 错误，则说明您的请求已超过 Amazon Managed Service for Prometheus 摄取配额。

```
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
```

如果您看到类似于以下示例的 429 错误，则说明您的请求已超过 Amazon Managed Service for Prometheus 配额，即工作区中活跃指标数量的配额。

```
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
```

如果您看到类似于以下示例的 400 错误，则说明您的请求已超过亚马逊托管服务 Prometheus 的有效时间序列配额。有关如何处理活跃时间序列配额的详细信息，请参阅默认活跃系列。

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

有关 Amazon Managed Service for Prometheus 服务配额以及如何请求提升配额的更多信息，请参阅 Amazon Managed Service for Prometheus 服务配额

我看到重复的样本

如果您使用的是高可用性 Prometheus 组，则需要在 Prometheus 实例上使用外部标签来设置重复数据删除。有关更多信息，请参阅 对发送到 Amazon Managed Service for Prometheus 的高可用性指标进行重复数据删除。

有关重复数据的其他问题将在下一节中讨论。

我看到有关样本时间戳的错误

Amazon Prometheus 托管服务按顺序提取数据，并期望每个样本的时间戳晚于前一个样本。

如果您的数据未按顺序送达，则可能会看到有关`out-of-order samples`、`duplicate sample for timestamp`或的错误`samples with different value but same timestamp`。这些问题通常是由向亚马逊 Prometheus 托管服务发送数据的客户端设置不正确造成的。如果您使用的是以代理模式运行的 Prometheus 客户端，请检查配置中是否存在系列名称重复或目标重复的规则。如果您的指标直接提供时间戳，请检查它们是否不合时宜。

有关其工作原理或检查设置方法的更多详细信息，请参阅 Prom Labs 的博客文章《了解 Prometheus 中的重复样本和 O ut-of-order 时间戳错误》。

我看到一条与限制相关的错误消息

> **Note**
>
> 适用于 Prometheus 的亚马逊托管服务提供CloudWatch 使用率指标来监控 Prometheus 的资源使用情况。使用 CloudWatch使用情况指标警报功能，您可以监控 Prometheus 的资源和使用情况，以防止出现限制错误。

如果您看到以下错误消息之一，则可以请求增加其中一个 Amazon Managed Service for Prometheus 配额来解决问题。有关更多信息，请参阅 Amazon Managed Service for Prometheus 服务配额。

- 超过了每个用户系列 <#> 的限制，请联系管理员以提高限制
- 超过了每个指标系列 <#> 的限制，请联系管理员以提高限制
- 超过了摄取率限制(...)
- 系列有太多标签(...)系列: '%s'
- 查询时间范围超出限制(查询长度: xxx，限制: yyy)
- 查询在从摄取器获取组块时达到最大组块数限制
- 超出了限制。每个账户的最大工作区。

您的本地 Prometheus 服务器输出超出了限制。

Amazon Managed Service for Prometheus 为工作区可以从 Prometheus 服务器接收的数据量设定了服务配额。要查找您的 Prometheus 服务器向 Amazon Managed Service for Prometheus 发送的数据量，您可以在 Prometheus 服务器上运行以下查询。如果您发现自己的 Prometheus 输出超出了 Amazon Managed Service for Prometheus 的限制，则可以请求增加相应的服务配额。有关更多信息，请参阅 Amazon Managed Service for Prometheus 服务配额。

查询您的本地自运行 Prometheus 服务器以查找输出限制。

数据类型	要使用的查询	
当前活跃系列	`prometheus_tsdb_head_series`	

数据类型	要使用的查询	
当前摄取率	`rate(prometheus_tsdb_head_samples_appended_total[5m])`	
每个指标名称的 M 个活跃系列ost-to-least 列表	`sort_desc(count by(__name__)({__name__!=""}))`	
每个指标系列的标签数	`group by(mylabelname)({__name__!=""})`	

我的一些数据没有出现

出于各种原因，发送到亚马逊 Prometheus 托管服务的数据可能会被丢弃。下表显示了数据可能被丢弃而不是被摄取的原因。

您可以使用 Amazon CloudWatch 跟踪丢弃数据的数量和原因。有关更多信息，请参阅 CloudWatch 指标。

Reason	含义
greater_than_max_sample_age	丢弃早于当前时间的日志行
new-value-for-timestamp	发送重复样本的时间戳与之前记录的时间戳不同
per_metric_series_limit	用户已达到每个指标活跃系列数上限

Reason	含义
per_user_series_limit	用户已达到活跃系列总数上限
rate_limited	摄取率受限制
sample-out-of-order	样本发送顺序混乱，无法处理
label_value_too_long	标签值超过允许的字符限制
max_label_names_per_series	用户已达到每个指标的标签名称数
missing_metric_name	未提供指标名称
metric_name_invalid	提供的指标名称无效
label_invalid	提供的标签无效
duplicate_label_names	提供的标签名称重复

Tagging

标签是您或 AWS 分配给 AWS 资源的自定义属性标签。每个 AWS 标签具有两个部分：

- 标签键 （例如，CostCenter、Environment、Project 或 Secret）。标签键区分大小写。
- 一个称为标签值的可选字段（例如，111122223333、Production 或团队名称）。省略标签值与使用空字符串效果相同。与标签键一样，标签值区分大小写。

这些被统称为键-值对。您最多可以向每个工作区指定 50 个标签。

标签有助于您标识和组织 AWS 资源。许多 AWS 服务支持标记，因此，您可以将同一标签分配给来不同服务的资源，以指示这些资源是相关的。例如，您可以将相同的标签分配给为 Amazon S3 桶分配的 Amazon Managed Service for Prometheus 工作区。有关标记策略的更多信息，请参阅标记 AWS 资源。

在 Amazon Managed Service for Prometheus 中，可以标记工作区和规则组命名空间。您可以使用控制台、AWS CLI、 API 或开发工具包为这些资源添加、管理和移除标签。除了通过标签标识、组织和跟踪工作区之外，您还可以在 IAM 策略中使用标签，进而控制哪些人可以查看并与您的资源交互。

标签限制

下面是适用于标签的基本限制：

- 每个资源最多可以有 50 个标签。
- 对于每个资源，每个标签键都必须是唯一的，每个标签键只能有一个值。
- 最大标签键长度为 128 个 Unicode 字符 (采用 UTF-8 格式)。
- 最大标签值长度为 256 个 Unicode 字符 (采用 UTF-8 格式)。
- 如果您的标记方案针对多个 AWS 服务和资源使用，请记得其它服务可能对允许使用的字符有限制。通常允许使用的字符包括可用 UTF-8 格式表示的字母、数字和空格，以及以下字符：. : + = @ _ / - （连字符）。
- 标签键和值区分大小写。最佳实践是，决定利用标签的策略并在所有资源类型中一致地实施该策略。例如，决定是否使用 Costcenter、costcenter 或 CostCenter，以及是否对所有标签使用相同的约定。避免将类似的标签用于不一致的案例处理。
- 请不要使用 aws:、AWS: 或任何大写或小写组合（例如，键或值的前缀）。它们保留供 AWS 使用。您无法编辑或删除带此前缀的标签键或值。具有此前缀的标签不计入每个资源的标签数限制。

主题

标记工作区

使用本部分中的过程处理 Amazon Managed Service for Prometheus 工作区的标签。

主题

向工作区添加标签

为 Amazon Managed Service for Prometheus 工作区添加标签有助于您标识和组织您的 AWS 资源并管理对这些资源的访问。首先，为工作区添加一个或多个标签（键值对）。有了标签后，您可以创建 IAM 策略来根据这些标签管理对工作区的访问。您可以使用控制台或 AWS CLI 向 Amazon Managed Service for Prometheus 工作区添加标签。

> **Important**
>
> 向工作区添加标签可能会影响对该工作区的访问。为工作区添加标签之前，请务必查看是否存在任何 IAM 策略可能使用标签来控制对资源的访问。

有关在创建 Amazon Managed Service for Prometheus 工作区时为其添加标签的更多信息，请参阅 创建工作区。

主题

向工作区添加标签（控制台）

您可以使用控制台向 Amazon Managed Service for Prometheus 工作区添加一个或多个标签。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择标签选项卡。
6. 如果尚未向 Amazon Managed Service for Prometheus 工作区添加任何标签，请选择创建标签。否则，请选择管理标签。
7. 在 Key (键) 中，输入标签的名称。您可以在 Value (值) 中添加可选的标签值。
8. （可选）要添加其他标签，请再次选择 Add tag (添加标签)。
9. 添加完标签后，选择保存更改。

向工作区添加标签（AWS CLI）

按照以下步骤使用 AWS CLI 向 Amazon Managed Service for Prometheus 工作区添加标签。要在创建工作区时为其添加标签，请参阅创建工作区。

在这些步骤中，我们假设您已安装最新版本的 AWS CLI 或已更新到当前版本。有关更多信息，请参阅安装 AWS Command Line Interface。

在终端或命令行运行 tag-resource 命令，指定要为其添加标签的工作区的 Amazon 资源名称（ARN），以及要添加的标签的键/值。您可以向工作区添加多个标签。例如，使用两个标签标记名为 My-Workspace 的 Amazon Managed Service for Prometheus 工作区，一个标签键名为 *Status*，标签值为 *Secret*；另一个标签键名为 *Team*，标签值为 *My-Team*：

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
--tags Status=Secret,Team=My-Team
```

如果成功，该命令不返回任何内容。

查看工作区的标签

标签可以帮助您标识和组织您的 AWS 资源并管理对其的访问。有关标记策略的更多信息，请参阅标记 AWS 资源。

查看 Amazon Managed Service for Prometheus 工作区的标签（控制台）

您可以使用控制台查看与 Amazon Managed Service for Prometheus 工作区关联的标签。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择标签选项卡。

查看 Amazon Managed Service for Prometheus 工作区的标签（AWS CLI）

可以按照以下步骤使用 AWS CLI 查看工作区的 AWS 标签。如果尚未添加标签，则返回的列表为空。

在终端或命令行中，运行 list-tags-for-resource 命令。例如，要查看工作区的标签键和标签值列表，请执行以下操作：

```
aws amp list-tags-for-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
```

如果成功，该命令返回类似以下内容的信息：

```
{
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

编辑工作区的标签

您可以更改与工作区关联的标签值。您也可以更改标签键的名称，这相当于删除当前的标签并使用新名称和相同的值添加一个不同的标签。

> **Important**
>
> 编辑 Amazon Managed Service for Prometheus 工作区的标签可能会影响对该工作区的访问。编辑工作区的标签名称（键）或值之前，请务必查看是否存在任何 IAM 策略可能使用标签的键或值来控制对资源（如存储库）的访问。

编辑 Amazon Managed Service for Prometheus 工作区的标签（控制台）

您可以使用控制台编辑与 Amazon Managed Service for Prometheus 工作区关联的标签。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择标签选项卡。
6. 如果尚未向此工作区添加任何标签，请选择创建标签。否则，请选择管理标签。
7. 在 Key (键) 中，输入标签的名称。您可以在 Value (值) 中添加可选的标签值。
8. （可选）要添加其他标签，请再次选择 Add tag (添加标签)。
9. 添加完标签后，选择保存更改。

编辑 Amazon Managed Service for Prometheus 工作区的标签（AWS CLI）

按照以下步骤使用 AWS CLI 来更新工作区的标签。您可以更改现有键的值或添加另一个键。

在终端或命令行中运行 tag-resource 命令，并指定要更新标签的 Amazon Managed Service for Prometheus 工作区的 Amazon 资源名称（ARN）以及标签键和标签值：

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tags Team=New-Team
```

从工作区中删除标签

您可以删除与工作区关联的一个或多个标签。删除标签不会从与该标签关联的其他 AWS 资源中删除该标签。

> **Important**
>
> 从 Amazon Managed Service for Prometheus 工作区中删除标签可能会影响对该工作区的访问。从工作区中删除标签之前，请务必查看是否存在任何 IAM 策略可能使用标签的键或值来控制对资源（如存储库）的访问。

从 Amazon Managed Service for Prometheus 工作区中删除标签（控制台）

您可以使用控制台移除标签和工作区之间的关联。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择标签选项卡。
6. 选择 Manage tags（管理标签）。
7. 找到要删除的标签，选择删除。

从 Amazon Managed Service for Prometheus 工作区删除标签（AWS CLI）。

执行以下步骤，使用 AWS CLI 从工作区中删除标签。删除标签并不会将其删除，而只是删除标签和工作区之间的关联。

> **Note**
>
> 如果您删除 Amazon Managed Service for Prometheus 工作区，则所有标签关联都将从已删除的工作区中删除。您无需在删除工作区之前删除标签。

在终端或命令行中运行 untag-resource 命令，并指定要移除标签的工作区的 Amazon 资源名称（ARN）以及要删除的标签的标签键。例如，在名为 My-Workspace 的工作区中删除具有标签键 *Status* 的标签：

```
aws amp untag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tag-keys Status
```

如果成功，该命令不返回任何内容。要验证与工作区关联的标签，请运行 list-tags-for-resource 命令。

标记规则组命名空间

使用本部分中的过程处理 Amazon Managed Service for Prometheus 规则组命名空间的标签。

主题

向规则组命名空间添加标签

向 Amazon Managed Service for Prometheus 规则组命名空间添加标签有助于您标识和组织您的 AWS 资源并管理对这些资源的访问。首先，向规则组命名空间添加一个或多个标签（键值对）。有了标签后，您可以创建 IAM 策略来根据这些标签管理对该命名空间的访问。您可以使用控制台或 AWS CLI 向 Amazon Managed Service for Prometheus 规则组命名空间添加标签。

> **Important**
>
> 向规则组命名空间添加标签可能会影响对该规则组命名空间的访问。在添加标签之前，请务必查看是否存在任何 IAM 策略可能使用标签来控制对资源的访问。

有关在创建规则组命名空间时为其添加标签的更多信息，请参阅 创建规则文件。

主题

向规则组命名空间添加标签（控制台）

您可以使用控制台向 Amazon Managed Service for Prometheus 规则组命名空间添加一个或多个标签。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择规则管理选项卡。
6. 选择命名空间名称旁边的按钮，然后选择编辑。
7. 请选择创建标签、添加新标签。
8. 在 Key (键) 中，输入标签的名称。您可以在 Value (值) 中添加可选的标签值。
9. （可选）要添加其它标签，请再次选择添加新标签。
10. 添加完标签后，选择保存更改。

向规则组命名空间添加标签（AWS CLI）

按照以下步骤使用 AWS CLI 向 Amazon Managed Service for Prometheus 规则组命名空间添加标签。要在创建规则组命名空间时向其添加标签，请参阅 将规则配置文件上传到 Amazon Managed Service for Prometheus。

在这些步骤中，我们假设您已安装最新版本的 AWS CLI 或已更新到当前版本。有关更多信息，请参阅安装 AWS Command Line Interface。

在终端或命令行运行 tag-resource 命令，指定要为其添加标签的规则组命名空间的 Amazon 资源名称（ARN），以及要添加的标签的键和值。您可以向规则组命名空间添加多个标签。*############ My-Workspace # Amazon Managed Service for Prometheus ########### Status##### Secret######### Team###### My-Team*：

```
aws amp tag-resource \
    --resource-arn arn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name \
    --tags Status=Secret,Team=My-Team
```

如果成功，该命令不返回任何内容。

查看规则组命名空间的标签

标签可以帮助您标识和组织您的 AWS 资源并管理对其的访问。有关标记策略的更多信息，请参阅标记 AWS 资源。

查看 Amazon Managed Service for Prometheus 规则组命名空间的标签（控制台）

您可以使用控制台查看与 Amazon Managed Service for Prometheus 规则组命名空间关联的标签。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择规则管理选项卡。
6. 选择命名空间名称。

查看 Amazon Managed Service for Prometheus 工作区的标签（AWS CLI）

按照以下步骤使用 AWS CLI 来查看规则组命名空间的 AWS 标签。如果尚未添加标签，则返回的列表为空。

在终端或命令行中，运行 list-tags-for-resource 命令。例如，要查看规则组命名空间的标签键和标签值列表，请执行以下操作：

```
aws amp list-tags-for-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name
```

如果成功，该命令返回类似以下内容的信息：

```
{
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

编辑规则组命名空间的标签

您可以更改与规则组命名空间关联的标签值。您也可以更改标签键的名称，这相当于删除当前的标签并使用新名称和相同的值添加一个不同的标签。

> **Important**
>
> 编辑规则组命名空间的标签可能会影响对该命名空间的访问。编辑资源的标签名称（键）或值之前，请务必查看是否存在任何 IAM 策略可能使用标签的键或值来控制对资源的访问。

编辑 Amazon Managed Service for Prometheus 规则组命名空间的标签（控制台）

您可以使用控制台编辑与 Amazon Managed Service for Prometheus 规则组命名空间关联的标签。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择规则管理选项卡。
6. 选择命名空间的名称。
7. 选择管理标签和添加新标签。
8. 要更改现有标签的值，请为值输入新值。
9. 要添加其它标签，请选择添加新标签。
10. 添加和编辑完标签后，选择保存更改。

编辑 Amazon Managed Service for Prometheus 规则组命名空间的标签（AWS CLI）

按照以下步骤使用 AWS CLI 来更新规则组命名空间的标签。您可以更改现有键的值或添加另一个键。

在终端或命令行中运行 tag-resource 命令，并指定要更新标签的资源的 Amazon 资源名称（ARN）以及标签键和标签值：

```
aws amp tag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tags Team=New-Team
```

从规则组命名空间中删除标签

您可以删除与规则组命名空间关联的一个或多个标签。删除标签不会从与该标签关联的其他 AWS 资源中删除该标签。

> **Important**
>
> 删除资源的标签可能会影响对该资源的访问。从资源中删除标签之前，请务必查看是否存在任何 IAM 策略可能使用标签的键或值来控制对资源（如存储库）的访问。

删除 Amazon Managed Service for Prometheus 规则组命名空间中的标签（控制台）

您可以使用控制台删除标签和规则组命名空间之间的关联。

1. 打开 Amazon Managed Service for Prometheus 控制台，网址为 https://console.aws.amazon.com/prometheus/。
2. 在导航窗格中，选择菜单图标。
3. 选择所有工作区。
4. 选择要管理的工作区的工作区 ID。
5. 选择规则管理选项卡。
6. 选择命名空间的名称。
7. 选择 Manage tags（管理标签）。
8. 在要删除的标签的旁边，选择删除。
9. 完成后，请选择保存更改。

删除 Amazon Managed Service for Prometheus 规则组命名空间中的标签（AWS CLI）

可以按照以下步骤使用 AWS CLI 从规则组命名空间中删除标签。删除标签并不会删除标签，而只是删除它和规则组命名空间之间的关联。

> **Note**
>
> 如果您删除 Amazon Managed Service for Prometheus 规则组命名空间，则所有标签关联都将从已删除的命名空间中删除。您无需在删除命名空间之前删除标签。

在终端或命令行中运行 untag-resource 命令，并指定要移除标签的规则组命名空间的 Amazon 资源名称（ARN）以及要移除的标签的标签键。例如，在名为 My-Workspace 的工作区中删除具有标签键 *Status* 的标签：

```
aws amp untag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tag-keys Status
```

如果成功，该命令不返回任何内容。要验证与资源关联的标签，请运行 list-tags-for-resource 命令。

Amazon Managed Service for Prometheus 服务配额

以下两个部分介绍了与 Amazon Managed Service for Prometheus 相关的配额和限制。

服务限额

Amazon Managed Service for Prometheus 的限额如下。适用于 Prometheus 的亚马逊托管服务提供使用率指标来监控 Prometheu CloudWatch s 的资源使用情况。使用 CloudWatch 使用情况指标警报功能，您可以监控 Prometheus 的资源和使用情况，以防止出现限制错误。

随着项目和工作区的增长，可能需要监控或请求增加的最常见配额是：每个工作区的活跃系列、每个工作区的摄取率和每个工作区的摄取突增大小。

对于所有可调整配额，您可以通过选择可调整列中的链接或通过请求增加限额来请求增加限额。

每个工作区的活跃系列限制是动态应用的。有关更多信息，请参阅默认活跃系列。每个工作空间的摄取速率和每个工作空间的摄取突发大小共同控制了将数据采集到工作空间的速度。有关更多信息，请参阅限制摄入量。

> **Note**
>
> 除非另有说明，否则这些限额适用于每个工作区。

名称	默认值	可调整	描述
每个工作区包含元数据的活动指标数	每个受支持的区域：2 万个	否	每个工作区具有元数据的活动指标的数量。
每个工作区的活跃系列数	每个支持的区域：每 2 小时 1000 万个	是	每个工作区的唯一活跃系列数量。如果在过去 2 小时内报告了样本，则该系列处于活跃状态。容量从 2M 到 10M，会根据最近 30 分钟的使用情况自动调整。

名称	默认值	可调整	描述
警报管理器定义文件中的警报聚合组大小	每个受支持的区域：1000 个	是	警报管理器定义文件中警报聚合组的最大大小。group_by 的每个标签值组合都将创建一个聚合组。
警报管理器定义文件大小	每个受支持的区域：1MB	否	警报管理器定义文件的最大大小。
警报管理器中的警报有效载荷大小	每个受支持的区域：20 MB	否	每个工作区所有警报管理器警报的最大警报负载大小。警报大小取决于标签和注释。
警报管理器中的警报	每个受支持的区域：1000 个	是	每个工作区并发警报管理器警报的最大数量。
HA 追踪器集群	每个受支持的区域：500 个	否	HA 追踪器将针对每个工作区摄取的样本跟踪的最大集群数。
每个工作区的摄取突增大小	每个受支持的区域：100 万个	是	每个工作区在每秒一次突增中可以摄取的最大样本数。
每个工作区的摄取率	每个支持的区域：17 万个	是	每个工作区每秒的指标样本摄取率。
警报管理器定义文件中的抑制规则数	每个受支持的区域：100 个	是	警报管理器定义文件中抑制规则的最大数量。
标签大小	每个支持的区域：7KB	否	一个系列接受的所有标签和标签值的最大组合大小。

名称	默认值	可调整	描述
每个指标系列的标签数	每个受支持的区域：70 个	是	每个指标系列的标签数。
元数据长度	每个受支持的区域：1 KB	否	指标元数据接受的最大长度。元数据是指指标名称、HELP 和 UNIT。
每个指标的元数据	每个受支持的区域：10 个	否	每个指标的最大元数据数。
警报管理器路由树中的节点数	每个受支持的区域：100 个	是	警报管理器路由树的最大节点数。
每秒事务中的 API 操作数	每个受支持的区域：10 个	是	每个区域每秒 API 操作最大数量。这包括工作区 CRUD API、标记 API、规则组命名空间 CRUD API 和警报管理器定义 CRUD API。
即时查询的查询字节数	每个受支持的区域：5 GB	否	单个即时查询可以扫描的最大字节数。
范围查询的查询字节数	每个受支持的区域：5 GB	否	单个范围查询中每 24 小时可以扫描的最大字节数。
已提取的查询区块	每个受支持的区域：2000 万个	否	单次查询期间可以扫描的最大组块数。
查询样本	每个受支持的区域：5000 万个	否	单次查询期间可以扫描的最大样本数。
已提取的查询序列	每个支持的区域：1200 万个	否	单次查询期间可以扫描的最大系列数。

名称	默认值	可调整	描述
查询时间范围（以天为单位）	每个受支持的区域：32 个	否	任何 PromQL 查询的最大时间范围。
请求大小	每个受支持的区域：1MB	否	摄取或查询的最大请求大小。
摄取数据的保留时间（以天为单位）	每个受支持的区域：150 个	是	数据在工作区的保留天数。早于此期限的数据将被删除。您可以请求更改配额以增加或减少此值。
规则评估间隔	每个受支持的区域：30 秒	是	每个工作区中规则组的最小规则评估间隔。
规则组命名空间定义文件大小	每个受支持的区域：1MB	否	规则组命名空间定义文件的最大大小。
每个工作区的规则数	每个受支持的区域：2000 个	是	每个工作区的最大规则数。
警报管理器定义文件中的模板数	每个受支持的区域：100 个	是	警报管理器定义文件中的最大模板数。
每个账户每个区域的工作区数	每个受支持的区域：25 个	是	每个区域的工作区最大数。

默认活跃系列

默认情况下，Amazon Managed Service for Prometheus 允许您最多使用活跃时间序列的配额。

Amazon Managed Service for Prometheus 工作区会自动根据您的摄取量进行调整。随着使用量的增加，Amazon Managed Service for Prometheus 将自动增加您的时间序列容量，使您的基准使用量翻一番，直至达到默认配额。例如，如果过去 30 分钟的平均活跃时间序列为 350 万，则可以使用多达 700 万个时间序列而不受限制。

如果您需要的容量超过之前基准的两倍，Amazon Managed Service for Prometheus 会随着您摄取量的增加自动分配更多容量，以便确保您的工作负载不会经历持续限制，不超过您的配额。但是，如果您过去 30 分钟超出先前基准值的两倍，则可能发生限制。为避免限制，Amazon Managed Service for Prometheus 建议在增加到之前活跃时间序列的两倍以上时，逐渐增加摄取量。

> **Note**
>
> 活动时间序列的最小容量为 200 万，当您的序列少于 200 万时没有限制。
> 要超出其默认配额，您可以请求增加限额。

限制摄入量

适用于 Prometheus 的亚马逊托管服务会根据您当前的限制限制每个工作空间的摄取量。这有助于保持工作空间的性能。如果你超过了限制，你将在 CloudWatch 指标`DiscardedSamples`中看到（并附上`rate_limited`原因）。您可以使用 Amazon CloudWatch 监控您的摄取量，并创建警报，在接近限制限制时向您发出警报。有关更多信息，请参阅CloudWatch 指标。

适用于 Prometheus 的亚马逊托管服务使用令牌存储桶算法来实现摄取限制。使用此算法，您的账户拥有一个持有特定数量的令牌的存储桶。存储桶中的代币数量代表您在任何给定秒钟的摄取限制。

采集的每个数据样本都会从存储桶中移除一个令牌。如果您的存储桶大小（每个工作空间的摄取突发大小）为 1,000,000，则您的工作空间可以在一秒钟内采集 100 万个数据样本。如果要摄取的样本超过一百万，它将受到限制，并且不会再摄取任何记录。其他数据样本将被丢弃。

存储桶会按设定的速率自动填充。如果存储桶低于其最大容量，则每秒向其添加一定数量的令牌，直到其达到最大容量。如果充值令牌到达时桶已满，则它们将被丢弃。存储桶容纳的代币数量不能超过其最大数量。样本摄取的补充速率由每个工作空间的摄取速率限制来设置。如果将每个工作空间的摄取率设置为 170,000，则存储桶的充值速率为每秒 170,000 个代币。

如果您的工作空间在一秒钟内提取了 1,000,000 个数据样本，则您的存储桶会立即减少到零令牌。然后，该存储桶每秒充满 170,000 个代币，直到其最大容量达到 1,000,000 个代币。如果不再进行摄取，则之前空的存储桶将在 6 秒钟内恢复到其最大容量。

> **Note**
>
> 摄取发生在批处理请求中。如果您有 100 个可用代币，并且发送了包含 101 个样本的请求，则整个请求都将被拒绝。亚马逊 Prometheus 托管服务不接受部分请求。如果您正在编写收集器，则可以管理重试（使用较小的批次或经过一段时间后）。

您无需等到存储桶已满之后您的工作空间就可以采集更多数据样本。您可以在令牌被添加到存储桶时使用这些令牌。如果您立即使用充值令牌，则存储桶无法达到其最大容量。例如，如果您耗尽存储桶，则可以继续每秒采集 170,000 个数据样本。只有当您每秒采集的数据样本少于 170,000 时，存储桶才能重新填充到最大容量。

摄取数据的额外限制

Amazon Managed Service for Prometheus 对摄取到工作区的数据有以下额外要求。这些不可调整。

- 超过 1 小时的指标样本会拒绝摄取。
- 每个样本和元数据都必须有一个指标名称。

API 参考

本部分列出了 Amazon Managed Service for Prometheus 支持的 API 操作和数据结构。

有关这些 API 操作及其系列、标签和 API 请求配额的信息，请参阅《Amazon Managed Service for Prometheus 用户指南》中的 Amazon Managed Service for Prometheus 服务配额。

主题

- Amazon Managed Service for Prometheus API
- 与 Prometheus 兼容的 API

Amazon Managed Service for Prometheus API

适用于 Prometheus 的亚马逊托管服务提供 API 操作，用于创建和维护您的 Prometheus 工作空间的亚马逊托管服务。这包括工作空间、抓取器、警报管理器定义、规则组、命名空间和日志记录的 API。

有关适用于 Prometheus 的亚马逊托管服务 API 的详细信息，请参阅适用于 Prometheus 的亚马逊托管服务 API 参考。

将适用于 Prometheus 的亚马逊托管服务与 SDK 配合使用 AWS

AWS 软件开发套件 (SDK) 可用于许多流行的编程语言。每个 SDK 都提供一个 API、代码示例和文档，便于开发人员使用自己的首选语言构建 AWS 应用程序。如需按语言列出的软件开发工具包和工具列表，请参阅 AWS 开发人员中心 AWS中的开发工具。

> **开发工具包版本**
>
> 我们建议您使用项目中使用的最新版本的 SD AWS K 以及任何其他 SDK，并使 SDK 保持最新。 AWS 开发工具包为您提供最新的特性和功能以及安全更新。

与 Prometheus 兼容的 API

Amazon Managed Service for Prometheus 支持以下与 Prometheus 兼容的 API。

有关使用与 Prometheus 兼容的 API 的更多信息，请参阅。使用与 Prometheus 兼容的 API 进行查询

主题

CreateAlertManagerAlerts

`CreateAlertManagerAlerts` 操作在工作区中创建警报。

有效的 HTTP 动词：

`POST`

有效的 URI：

`/workspaces/workspaceId/alertmanager/api/v2/alerts`

URL 查询参数：

`alerts` 对象数组，其中每个对象代表一个警报。以下是警报对象的示例。

```
[
```

```
  {
    "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"
  }
]
```

示例请求

```
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/"
  }
]
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
```

```
Content-Length: 0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

DeleteAlertManagerSilence

DeleteSilence 删除一个警报静默。

有效的 HTTP 动词：

DELETE

有效的 URI：

/workspaces/*workspaceId*/alertmanager/api/v2/silence/*silenceID*

URL 查询参数：无

示例请求

```
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
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

GetAlertManagerStatus

GetAlertManagerStatus 检索有关警报管理器状态的信息。

有效的 HTTP 动词：

```
GET
```

有效的 URI：

```
/workspaces/workspaceId/alertmanager/api/v2/status
```

URL 查询参数：无

示例请求

```
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
```

示例响应

```
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

GetAlertManagerSilence 检索有关一个警报静默的信息。

有效的 HTTP 动词：

```
GET
```

有效的 URI：

```
/workspaces/workspaceId/alertmanager/api/v2/silence/silenceID
```

URL 查询参数：无

示例请求

```
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
```

示例响应

```
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 操作检索与时间序列关联的标签。

有效的 HTTP 动词：

GET, POST

有效的 URI：

/workspaces/*workspaceId*/api/v1/labels

/workspaces/*workspaceId*/api/v1/label/*label-name*/values 此 URI 仅支持 GET 请求。

URL 查询参数：

match[]=<series_selector> 重复的序列选择器参数，用于选择要从中读取标签名称的序列。可选。

start=<rfc3339 | unix_timestamp> 开始时间戳。可选。

end=<rfc3339 | unix_timestamp> 结束时间戳。可选。

/workspaces/workspaceId/api/v1/labels 的示例请求

```
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/labels 的示例响应

```
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/values 的示例请求

```
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/values 的示例响应

```
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 操作检索有关当前正在从目标中抓取的指标的元数据。它不提供任何目标信息。

查询结果的数据部分由一个对象组成，其中每个键是一个指标名称，每个值是唯一的元数据对象的列表，在所有目标中针对该指标名称公开。

有效的 HTTP 动词：

GET

有效的 URI：

/workspaces/*workspaceId*/api/v1/metadata

URL 查询参数：

`limit=<number>` 要返回的最大指标数。

`metric=<string>` 要筛选元数据的指标名称。如果将其保留为空，则会检索所有指标元数据。

示例请求

```
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
```

示例响应

```
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` 操作检索与特定标签集匹配的时间序列列表。

有效的 HTTP 动词：

GET, POST

有效的 URI：

/workspaces/*workspaceId*/api/v1/series

URL 查询参数：

match[]=<series_selector> 重复的序列选择器参数，用于选择要返回的序列。必须至少提供一个 match[] 参数。

start=<rfc3339 | unix_timestamp> 开始时间戳。可选

end=<rfc3339 | unix_timestamp> 结束时间戳。可选

示例请求

```
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
```

示例响应

```
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` 操作检索工作区中当前处于活动状态的警报。

有效的 HTTP 动词：

`GET`

有效的 URI：

```
/workspaces/workspaceId/api/v1/alerts
```

示例请求

```
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
```

示例响应

```
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

`ListAlertManagerAlerts` 检索有关工作区警报管理器中当前触发的警报的信息。

有效的 HTTP 动词：

```
GET
```

有效的 URI：

```
/workspaces/workspaceId/alertmanager/api/v2/alerts
```

示例请求

```
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
```

示例响应

```
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` 操作检索工作区警报管理器中配置的警报组列表。

有效的 HTTP 动词：

`GET`

有效的 URI：

`/workspaces/workspaceId/alertmanager/api/v2/alerts/groups`

URL 查询参数：

`active` 布尔值。如果为 true，则返回的列表包括活动警报。默认值为 true。可选

`silenced` 布尔值。如果为 true，则返回的列表包括静默警报。默认值为 true。可选

`inhibited` 布尔值。如果为 true，则返回的列表包括抑制的警报。默认值为 true。可选

`filter` 字符串数组。用于筛选警报的匹配器列表。可选

`receiver` 字符串。匹配接收方以筛选警报的正则表达式。可选

示例请求

```
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
```

示例响应

```
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` 操作检索有关警报管理器中配置的接收方的信息。

有效的 HTTP 动词：

`GET`

有效的 URI：

`/workspaces/workspaceId/alertmanager/api/v2/receivers`

URL 查询参数：无

示例请求

```
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
```

示例响应

```
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 操作检索有关在工作区中配置的警报静默的信息。

有效的 HTTP 动词：

GET

有效的 URI：

/workspaces/*workspaceId*/alertmanager/api/v2/silences

示例请求

```
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
```

示例响应

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 312
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "id": "d29d9df3-9125-4441-912c-70b05f86f973",
        "status": {
```

```
            "state": "active"
        },
        "updatedAt": "2021-10-22T19:32:11.763Z",
        "comment": "hello-world",
        "createdBy": "test-person",
        "endsAt": "2023-07-24T01:05:36.000Z",
        "matchers": [
            {
                "isEqual": true,
                "isRegex": true,
                "name": "job",
                "value": "hello"
            }
        ],
        "startsAt": "2021-10-22T19:32:11.763Z"
    }
]
```

ListRules

`ListRules` 检索有关工作区中配置的规则的信息。

有效的 HTTP 动词：

```
GET
```

有效的 URI：

```
/workspaces/workspaceId/api/v1/rules
```

示例请求

```
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
```

示例响应

```
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 操作创建新的警报静默或更新现有的警报静默。

有效的 HTTP 动词：

POST

有效的 URI：

/workspaces/*workspaceId*/alertmanager/api/v2/silences

URL 查询参数：

silence 表示静默的对象。格式如下：

```
{
  "id": "string",
  "matchers": [
    {
      "name": "string",
      "value": "string",
      "isRegex": Boolean,
      "isEqual": Boolean
    }
  ],
  "startsAt": "timestamp",
  "endsAt": "timestamp",
  "createdBy": "string",
  "comment": "string"
}
```

示例请求

```
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"
}
```

示例响应

```
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 操作评估单个时间点或一段时间内的即时查询。

有效的 HTTP 动词：

GET, POST

有效的 URI：

/workspaces/*workspaceId*/api/v1/query 此 URI 评估单个时间点的即时查询。

/workspaces/*workspaceId*/api/v1/query_range 此 URI 评估一段时间内的即时查询。

URL 查询参数：

query=<string> Prometheus 表达式查询字符串。在 query 和 query_range 中使用。

time=<rfc3339 | unix_timestamp>（可选）如果您对单个时间点的即时查询使用 query，则为评估时间戳。

timeout=<duration>（可选）评估超时。默认为以 -query.timeout 标志的值为上限。在 query 和 query_range 中使用。

start=<rfc3339 | unix_timestamp> 如果您使用 query_range 查询一段时间范围，则为开始时间戳。

end=<rfc3339 | unix_timestamp> 如果您使用 query_range 查询一段时间范围，则为结束时间戳。

step=<duration | float> duration 格式的查询解析步长，或者是 float 秒数。仅在您使用 query_range 查询一段时间范围时使用，并且对于此类查询是必需的。

Duration

与 Prometheus 兼容的 API 中的 duration 是一个数值，后面紧跟以下单位之一：

- ms 毫秒
- s 秒
- m 分钟
- h 小时
- d 天，假设一天始终是 24 小时
- w 周，假设一周总始终是 7 天
- y 年，假设一年始终是 365 天

示例请求

```
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
```

示例响应

```
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 操作以标准格式将指标从 Prometheus 服务器写入远程 URL。通常，您将使用现有客户端（例如 Prometheus 服务器）来调用此操作。

有效的 HTTP 动词：

POST

有效的 URI：

/workspaces/*workspaceId*/api/v1/remote_write

URL 查询参数：

无

RemoteWrite 摄取率为每秒 7 万个样本，摄入突增大小为 100 万个样本。

示例请求

```
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**
>
> 有关请求正文语法，请参阅协议缓冲区定义，网址为： https://github.com/prometheus/prometheus/blob/1c624c58ca934f618be737b4995e22051f5724c1/prompb/remote.pb.go#L64。

示例响应

```
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 用户指南》的文档历史记录

下表介绍了《Amazon Managed Service for Prometheus 用户指南》中的重要文档更新。要获得本文档的更新通知，您可以订阅 RSS 源。

变更	说明	日期
增加了在控制台中编辑规则定义文件和警报管理器配置文件的功能	适用于 Prometheus 的亚马逊托管服务增加了对从适用于 Prometheus 的亚马逊托管服务控制台中编辑警报管理器配置文件和规则定义文件的支持。	2024 年 5 月 16 日
添加了更简单的 AWS 托管收集器设置，其中包含了 Amazon EKS 的访问条目	适用于 Prometheus 的亚马逊托管服务增加了对亚马逊 EKS 访问条目的支持，以简化托管收集器的设置。AWSAmazonPrometheusScraperServiceRolePolicy托管收集器的 AWS 托管策略已更新，允许删除不再使用的访问条目。	2024年5月2日
将 AWS API 移至单独的 API 参考指南	适用于 Prometheus 的亚马逊托管服务 API 现已在其自己的参考文献中提供，即《适用于 AWS Prometheus 的亚马逊托管服务 API 参考》。与 Prometheus 兼容的 API 继续记录在《适用于 Prometheus 的亚马逊托管服务用户指南》中。	2024 年 2 月 7 日
增加了用于工作区加密的客户托管密钥	Amazon Managed Service for Prometheus 增加了对用于工	2023 年 12 月 21 日

	作区加密的客户托管密钥的支持。有关更多信息，请参阅静态加密。	
向添加了新权限 AmazonPrometheusFullAccess	为AmazonPrometheusFullAccess托管策略添加了新的权限，以支持为 Amazon EKS 集群创建 AWS 托管收集器。	2023 年 11 月 26 日
添加了新的托管策略，AmazonPrometheusScraperServiceLinkedRolePolicy	添加了新的托管策略，AmazonPrometheusScraperServiceLinkedRolePolicy允许 AWS 托管收集器从 Amazon EKS 集群收集指标。	2023 年 11 月 26 日
添加了 AWS 托管收集器作为摄取方法	Amazon Managed Service for Prometheus 增加了对 AWS 托管收集器的支持。	2023 年 11 月 26 日
增加了对与 Amazon Managed Grafana 集成的支持	Amazon Managed Service for Prometheus 增加了对与 Amazon Managed Grafana 警报集成的支持。	2022 年 11 月 23 日
向添加了新权限 AmazonPrometheusConsoleFullAccess	为AmazonPrometheusConsoleFullAccess托管策略添加了新的权限，以支持在 CloudWatch 日志中记录警报管理器和标尺事件。	2022 年 10 月 24 日
增加了 Amazon EKS 可观察性解决方案。	Amazon Prometheus 托管服务使用可观测性加速器添加了新的解决方案。 AWS 有关更多信息，请参阅使用 AWS Observability Accelerator。	2022 年 10 月 14 日

增加了对集成到 Amazon EKS 成本监控的支持。	Amazon Managed Service for Prometheus 增加了对集成到 Amazon EKS 成本监控的支持。有关更多信息，请参阅与 Amazon EKS 成本监控集成。	2022 年 9 月 22 日
在 Amazon 日志中启动了对警报管理器和标尺 CloudWatch 日志的支持。	适用于 Prometheus 的亚马逊托管服务开始支持亚马逊日志中的警报管理器和标尺错误日志。 CloudWatch 有关更多信息，请参阅 Amazon CloudWatch 日志。	2022 年 9 月 1 日
增加了自定义存储保留支持。	Amazon Managed Service for Prometheus 通过修改工作区的配额为每个工作区增加了自定义存储保留支持。有关 Amazon Managed Service for Prometheus 中的配额的更多信息，请参阅服务配额。	2022 年 8 月 12 日
向 Amazon 添加了使用量指标 CloudWatch。	适用于 Prometheus 的亚马逊托管服务增加了对向亚马逊发送使用量指标的支持。 CloudWatch有关更多信息，请参阅 Amazon CloudWatch 指标。	2022 年 5 月 6 日
增加了对欧洲（伦敦）区域的支持。	Amazon Managed Service for Prometheus 增加了对欧洲（伦敦）区域的支持。	2022 年 5 月 4 日

Amazon Managed Service for Prometheus 现已正式推出，增加了对规则和警报管理器的支持。	Amazon Managed Service for Prometheus 现已正式推出。它还支持规则和警报管理器。有关更多信息，请参阅记录规则和警报规则和警报管理器和模板。	2021 年 9 月 29 日
增加了标记支持。	Amazon Managed Service for Prometheus 支持为 Amazon Managed Service for Prometheus 工作区添加标签。	2021 年 9 月 7 日
增加了活跃系列和摄取率配额。	活跃系列配额增加到 100 万个，摄取率配额增加到每秒 7 万个样本。	2021 年 2 月 22 日
Amazon Managed Service for Prometheus 预览版本。	Amazon Managed Service for Prometheus 预览版已发布。	2020 年 12 月 15 日

AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。