



用户指南

Amazon Managed Service for Prometheus



Amazon Managed Service for Prometheus: 用户指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Amazon Managed Service for Prometheus ?	1
支持的区域 :	1
定价	12
高级支持	12
开始使用	13
设置 AWS	13
注册获取 AWS 账户	13
创建工作区	14
摄取指标	15
步骤 1 : 添加新的 Helm 图表存储库	15
步骤 2 : 创建 Prometheus 命名空间	16
步骤 3 : 设置服务账户的 IAM 角色。	16
步骤 4 : 设置新服务器并开始摄取指标	16
查询指标	17
管理工作区	19
创建工作区	19
配置工作区	21
编辑工作区别名	22
查找工作区详细信息	23
删除工作区	25
摄取指标	26
AWS 托管收集器	26
集成 Amazon EKS	27
集成 Amazon MSK	45
与 Prometheus 兼容的指标	61
监控收集器	61
客户托管收集器	66
保护指标的摄取	67
ADOT 收集器	68
Prometheus 收集器	83
高可用性数据	91
查询指标	99
PromQL 备忘单	99
基本选择器	100

范围向量选择器	100
聚合运算符	100
常见的函数	101
二元运算符	101
实用查询示例	102
确保指标查询安全	102
使用 AWS PrivateLink 使用适用于 Prometheus 的亚马逊托管服务	67
身份验证和授权	67
使用 Amazon Managed Grafana	103
在私有 VPC 中连接到 Amazon Managed Grafana	104
使用 Grafana 开源	104
先决条件	104
步骤 1：设置 AWS SigV4	105
步骤 2：在 Grafana 中添加 Prometheus 数据来源	106
步骤 3：(可选) 如果 “保存&测试” 不起作用进行故障排除	108
在 Amazon EKS 中使用 Grafana	109
设置 AWS SigV4	109
设置服务账户的 IAM 角色	110
使用 Helm 升级 Grafana 服务器	111
在 Grafana 中添加 Prometheus 数据来源	111
使用直接查询	112
使用 awscurl 进行查询	113
查询统计数据	115
异常检测	119
异常检测的工作原理	119
异常检测入门	119
PreviewAnomalyDetector	120
查询参数格式	121
API 请求和响应	121
记录和警报规则	124
必要的 IAM 权限	125
创建规则文件	126
上传规则文件	127
编辑规则文件	129
对规则评估进行故障排除	130
验证警报触发状态	130

解决警报通知缺失	131
检查规则运行状况	131
在查询中使用偏移量来处理摄取延迟	133
常见问题和解决方案	134
规则评估的最佳实践	134
规则器故障排除	135
警报管理器	136
必要的 IAM 权限	137
创建配置文件	137
设置警报接收器	139
Amazon SNS	139
PagerDuty	149
上传配置文件	154
将警报与 Grafana 集成	156
先决条件	156
设置 Amazon Managed Grafana	157
警报管理器故障排除	158
活动警报警告	159
警报聚合组大小警告	159
警报大小过大警告	160
空内容警告	160
key/value 警告无效	161
消息限制警告	161
没有基于资源的策略错误	162
非 ASCII 警告	162
未授权调用 KMS	163
模板错误	163
监控工作区	165
CloudWatch 指标	165
设置 CloudWatch 闹铃	176
CloudWatch 日志	177
配置 CloudWatch 日志	177
查询见解和控制	179
配置查询日志记录	180
配置查询节流阈值	182
日志内容	182

限制	183
了解和优化成本	184
哪些因素会增加我的成本？	184
降低成本的最佳方法是什么？ 如何降低摄取成本？	184
降低我的查询成本的最佳方法是什么？	184
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？	184
如何降低警报查询成本？	185
我可以随时查看账单吗？	186
我可以使用的哪些指标来监控我的成本？	186
如何查看我的费用 AWS Cost Explorer？	187
如何计算一个月内摄入的样本数量？	188
哪些数据粒度可用于历史成本分析？	189
监控亚马逊托管服务 Prometheus 成本的最佳做法是什么？	190
为什么我月初的账单高于月末？	190
我删除了我所有的 Amazon Managed Service for Prometheus 工作区，但似乎仍在收费。这可能是怎么回事？	190
集成	191
Amazon EKS 成本监控	191
AWS 可观测性加速器	192
先决条件	192
使用托管指标（无代理）示例	193
备选方案：自行管理 OpenTelemetry 的收集器	194
查看控制面板	195
AWS 适用于 Kubernetes 的控制器	196
先决条件	196
部署工作区	197
配置集群以进行远程写入	200
Amazon CloudWatch 指标与 Firehose	202
基础设施	203
创建 Amazon CloudWatch 流	205
清理	206
安全性	207
数据保护	207
Amazon Managed Service for Prometheus 收集的数据	208
静态加密	209
身份和访问管理	221

受众	221
使用身份进行身份验证	222
使用策略管理访问	223
Amazon Managed Service for Prometheus 如何与 IAM 配合使用	224
基于身份的策略示例	229
问题排查	231
IAM 权限和策略	233
Amazon Managed Service for Prometheus 权限	233
示例 IAM 策略	233
合规性验证	234
恢复能力	234
基础设施安全性	234
使用服务关联角色	235
指标抓取角色	235
CloudTrail 日志	237
适用于 Prometheus 的亚马逊托管服务管理活动 CloudTrail	238
Amazon Managed Service for Prometheus 事件示例	238
设置服务账户的 IAM 角色	243
设置服务角色从 Amazon EKS 集群中摄取指标	243
设置服务账户的 IAM 角色以查询指标	246
接口 VPC 端点	249
为 Amazon Managed Service for Prometheus 创建接口 VPC 终端节点	250
问题排查	253
429 或超出限制错误	253
我看到重复的样本	254
我看到关于样本时间戳的错误	255
我看到一条与限制相关的错误消息	255
您的本地 Prometheus 服务器输出超出了限制。	256
我的一些数据没有出现	257
标签	258
标记工作区	259
向工作区添加标签	259
查看工作区的标签	261
编辑工作区的标签	262
从工作区中删除标签	263
标记规则组命名空间	264

向规则组命名空间添加标签	264
查看规则组命名空间的标签	266
编辑规则组命名空间的标签	267
从规则组命名空间中删除标签	268
服务配额	270
服务配额	270
活跃系列默认配额	275
扩展到超出默认配额	276
摄取节流	276
摄取数据的额外限制	277
API 参考	278
Amazon Managed Service for Prometheus API	278
将 Amazon Managed Service for Prometheus 与 AWS SDK 配合使用	278
兼容普罗米修斯 APIs	279
CreateAlertManagerAlerts	279
DeleteAlertManagerSilence	281
GetAlertManagerStatus	282
GetAlertManagerSilence	283
GetLabels	284
GetMetricMetadata	286
GetSeries	288
ListAlerts	290
ListAlertManagerAlerts	291
ListAlertManagerAlertGroups	292
ListAlertManagerReceivers	294
ListAlertManagerSilences	295
ListRules	296
PutAlertManagerSilences	297
QueryMetrics	299
RemoteWrite	301
文档历史记录	303
.....	cccviii

什么是 Amazon Managed Service for Prometheus ?

适用于 Prometheus 的 Amazon 托管服务是一项针对容器指标的无服务器监控服务 Prometheus-compatible，可以更轻松地大规模安全地监控容器环境。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。

适用于 Prometheus 的亚马逊托管服务旨在使用多个可用区 () 部署实现高可用性。Multi-AZ 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 Prometheus，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。

摄取到工作区的指标默认存储 150 天，然后自动删除。您可以通过将工作区配置为最长 1095 天 (三年) 来调整保留期。有关更多信息，请参阅[配置您的工作区](#)。

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	HTTPS
		aps-workspaces.us-east-2.amazonaws.com	HTTPS
		aps-workspaces-fips.us-east-2.amazonaws.com	HTTPS
		aps-workspaces-fips.us-east-2.api.aws	HTTPS

区域名称	区域	端点	协议
		aps-workspaces.us-east-2.api.aws	HTTPS
		aps-fips.us-east-2.amazonaws.com	HTTPS
		aps.us-east-2.api.aws	HTTPS
		aps-fips.us-east-2.api.aws	
美国东部 (弗吉尼亚州北部)	us-east-1	aps.us-east-1.amazonaws.com	HTTPS
		aps-workspaces.us-east-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-east-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-east-1.api.aws	HTTPS
		aps-workspaces.us-east-1.api.aws	HTTPS
		aps-fips.us-east-1.amazonaws.com	HTTPS
		aps.us-east-1.api.aws	HTTPS
		aps-fips.us-east-1.api.aws	HTTPS
美国西部 (北加利福尼亚)	us-west-1	aps.us-west-1.amazonaws.com	HTTPS
		aps-workspaces.us-west-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-west-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-west-1.api.aws	HTTPS
		aps-workspaces.us-west-1.api.aws	HTTPS
		aps-fips.us-west-1.amazonaws.com	HTTPS
		aps.us-west-1.api.aws	HTTPS
		aps-fips.us-west-1.api.aws	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
		aps-workspaces-fips.us-west-2.amazonaws.com	HTTPS
		aps-workspaces-fips.us-west-2.api.aws	HTTPS
		aps-workspaces.us-west-2.api.aws	HTTPS
		aps-fips.us-west-2.amazonaws.com	HTTPS
		aps.us-west-2.api.aws	HTTPS
		aps-fips.us-west-2.api.aws	HTTPS
非洲 (开普敦)	af-south-1	aps.af-south-1.amazonaws.com	HTTPS
		aps-workspaces.af-south-1.amazonaws.com	HTTPS
		aps-workspaces.af-south-1.api.aws	HTTPS
		aps.af-south-1.api.aws	HTTPS
亚太地区 (香港)	ap-east-1	aps.ap-east-1.amazonaws.com	HTTPS
		aps-workspaces.ap-east-1.amazonaws.com	HTTPS
		aps-workspaces.ap-east-1.api.aws	HTTPS
		aps.ap-east-1.api.aws	HTTPS
亚太地区 (海得拉巴)	ap-south-2	aps.ap-south-2.amazonaws.com	HTTPS
		aps-workspaces.ap-south-2.amazonaws.com	HTTPS
		aps-workspaces.ap-south-2.api.aws	HTTPS
		aps.ap-south-2.api.aws	HTTPS

区域名称	区域	端点	协议
亚太地区 (雅加达)	ap-southeast-3	aps.ap-southeast-3.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-3.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-3.api.aws	HTTPS
		aps.ap-southeast-3.api.aws	HTTPS
亚太地区 (马来西亚)	ap-southeast-5	aps.ap-southeast-5.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-5.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-5.api.aws	HTTPS
		aps.ap-southeast-5.api.aws	HTTPS
亚太地区 (墨尔本)	ap-southeast-4	aps.ap-southeast-4.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-4.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-4.api.aws	HTTPS
		aps.ap-southeast-4.api.aws	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
		aps-workspaces.ap-south-1.api.aws	HTTPS
		aps.ap-south-1.api.aws	HTTPS

区域名称	区域	端点	协议
亚太地区 (大阪)	ap-northeast-3	aps.ap-northeast-3.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-3.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-3.api.aws	HTTPS
		aps.ap-northeast-3.api.aws	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
		aps-workspaces.ap-northeast-2.api.aws	HTTPS
		aps.ap-northeast-2.api.aws	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
		aps-workspaces.ap-southeast-1.api.aws	HTTPS
		aps.ap-southeast-1.api.aws	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
		aps-workspaces.ap-southeast-2.api.aws	HTTPS
		aps.ap-southeast-2.api.aws	HTTPS

区域名称	区域	端点	协议
亚太地区 (台北)	ap-east-2	aps.ap-east-2.amazonaws.com	HTTPS
		aps-workspaces.ap-east-2.amazonaws.com	HTTPS
		aps-workspaces.ap-east-2.api.aws	HTTPS
		aps.ap-east-2.api.aws	HTTPS
亚太地区 (泰国)	ap-southeast-7	aps.ap-southeast-7.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-7.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-7.api.aws	HTTPS
		aps.ap-southeast-7.api.aws	HTTPS
亚太地区 (东京)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-northeast-1.api.aws	HTTPS
		aps.ap-northeast-1.api.aws	HTTPS

区域名称	区域	端点	协议
加拿大 (中部)	ca-central-1	aps.ca-central-1.amazonaws.com	HTTPS
		aps-workspaces.ca-central-1.amazonaws.com	HTTPS
		aps-workspaces-fips.ca-central-1.amazonaws.com	HTTPS
		aps-workspaces-fips.ca-central-1.api.aws	HTTPS
		aps-workspaces.ca-central-1.api.aws	HTTPS
		aps-fips.ca-central-1.amazonaws.com	HTTPS
		aps.ca-central-1.api.aws	HTTPS
		aps-fips.ca-central-1.api.aws	HTTPS
加拿大西部 (卡尔加里)	ca-west-1	aps.ca-west-1.amazonaws.com	HTTPS
		aps-workspaces.ca-west-1.amazonaws.com	HTTPS
		aps-workspaces-fips.ca-west-1.amazonaws.com	HTTPS
		aps-workspaces-fips.ca-west-1.api.aws	HTTPS
		aps-workspaces.ca-west-1.api.aws	HTTPS
		aps-fips.ca-west-1.amazonaws.com	HTTPS
		aps.ca-west-1.api.aws	HTTPS
		aps-fips.ca-west-1.api.aws	HTTPS

区域名称	区域	端点	协议
欧洲地区 (法兰克福)	eu-centra l-1	aps.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.api.aws	HTTPS
		aps.eu-central-1.api.aws	HTTPS
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
		aps-workspaces.eu-west-1.api.aws	HTTPS
		aps.eu-west-1.api.aws	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
		aps-workspaces.eu-west-2.api.aws	HTTPS
		aps.eu-west-2.api.aws	HTTPS
欧洲地区 (米兰)	eu-south- 1	aps.eu-south-1.amazonaws.com	HTTPS
		aps-workspaces.eu-south-1.amazonaws.com	HTTPS
		aps-workspaces.eu-south-1.api.aws	HTTPS
		aps.eu-south-1.api.aws	HTTPS
欧洲地区 (巴黎)	eu- west-3	aps.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.api.aws	HTTPS
		aps.eu-west-3.api.aws	HTTPS

区域名称	区域	端点	协议
欧洲 (西班牙)	eu-south-2	aps.eu-south-2.amazonaws.com	HTTPS
		aps-workspaces.eu-south-2.amazonaws.com	HTTPS
		aps-workspaces.eu-south-2.api.aws	HTTPS
		aps.eu-south-2.api.aws	HTTPS
欧洲地区 (斯德哥尔摩)	eu-north-1	aps.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.api.aws	HTTPS
		aps.eu-north-1.api.aws	HTTPS
欧洲 (苏黎世)	eu-central-2	aps.eu-central-2.amazonaws.com	HTTPS
		aps-workspaces.eu-central-2.amazonaws.com	HTTPS
		aps-workspaces.eu-central-2.api.aws	HTTPS
		aps.eu-central-2.api.aws	HTTPS
以色列 (特拉维夫)	il-central-1	aps.il-central-1.amazonaws.com	HTTPS
		aps-workspaces.il-central-1.amazonaws.com	HTTPS
		aps-workspaces.il-central-1.api.aws	HTTPS
		aps.il-central-1.api.aws	HTTPS
墨西哥 (中部)	mx-central-1	aps.mx-central-1.amazonaws.com	HTTPS
		aps-workspaces.mx-central-1.amazonaws.com	HTTPS
		aps-workspaces.mx-central-1.api.aws	HTTPS
		aps.mx-central-1.api.aws	HTTPS

区域名称	区域	端点	协议
中东 (巴林)	me-south-1	aps.me-south-1.amazonaws.com	HTTPS
		aps-workspaces.me-south-1.amazonaws.com	HTTPS
		aps-workspaces.me-south-1.api.aws	HTTPS
		aps.me-south-1.api.aws	HTTPS
中东 (阿联酋)	me-central-1	aps.me-central-1.amazonaws.com	HTTPS
		aps-workspaces.me-central-1.amazonaws.com	HTTPS
		aps-workspaces.me-central-1.api.aws	HTTPS
		aps.me-central-1.api.aws	HTTPS
南美洲 (圣保罗)	sa-east-1	aps.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.api.aws	HTTPS
		aps.sa-east-1.api.aws	HTTPS

区域名称	区域	端点	协议
AWS GovCloud (US-East)	us-gov-east-1	aps.us-gov-east-1.amazonaws.com	HTTPS
		aps-workspaces.us-gov-east-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-gov-east-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-gov-east-1.api.aws	HTTPS
		aps-workspaces.us-gov-east-1.api.aws	HTTPS
		aps-fips.us-gov-east-1.amazonaws.com	HTTPS
		aps.us-gov-east-1.api.aws	HTTPS
		aps-fips.us-gov-east-1.api.aws	HTTPS
AWS GovCloud (US-West)	us-gov-west-1	aps.us-gov-west-1.amazonaws.com	HTTPS
		aps-workspaces.us-gov-west-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-gov-west-1.amazonaws.com	HTTPS
		aps-workspaces-fips.us-gov-west-1.api.aws	HTTPS
		aps-workspaces.us-gov-west-1.api.aws	HTTPS
		aps-fips.us-gov-west-1.amazonaws.com	HTTPS
		aps.us-gov-west-1.api.aws	HTTPS
		aps-fips.us-gov-west-1.api.aws	HTTPS

适用于 Prometheus 的 Amazon 托管服务包括控制平面终端节点（用于执行工作空间管理任务）和数据平面终端节点（用于 Prometheus-compatible 处理工作空间实例中的数据）。控制面板端点以

aps.* 开头，数据面板端点以 aps-workspaces.* 开头。以 .amazonaws.com 结尾的端点支持 IPv4，以 .api.aws 结尾的端点同时支持 IPv4 和 IPv6。

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 AWS 成本 AWS Cost Explorer 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 Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus 是一种无服务器、与 Prometheus 兼容的服务，用于监控容器指标，可轻松安全地大规模监控容器环境。本节将介绍使用 Amazon Managed Service for Prometheus 的三个关键领域：

- [创建工作区](#) – 创建 Amazon Managed Service for Prometheus 工作区来存储和监控您的指标。
- [摄取指标](#) - 在将指标输入工作区之前，工作区是空的。您可以向 Amazon Managed Service for Prometheus 发送指标，或让 Amazon Managed Service for Prometheus 自动抓取指标。
- [查询指标](#) - 将指标作为数据存入工作区后，就可以随时查询数据，以探索或监控这些指标。

如果您不熟悉 AWS，本节还包括[有关设置的详细信息 AWS 账户](#)。

主题

- [设置 AWS](#)
- [创建 Amazon Managed Service for Prometheus 工作区](#)
- [将 Prometheus 指标摄取到工作区](#)
- [查询 Prometheus 指标](#)

设置 AWS

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 Amazon Managed Service for Prometheus 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](#)

注册获取 AWS 账户

首先 AWS，你需要一个 AWS 账户。有关创建的信息 AWS 账户，请参阅《AWS 账户管理 参考指南》AWS 账户中的[入门](#)指南。

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。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

有关创建工作区和可用选项的更多详细信息，请参阅[创建 Amazon Managed Service for Prometheus 工作区](#)。

创建 Amazon Managed Service for Prometheus 工作区

1. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。两个工作空间可以具有相同的别名，但所有工作空间都将具有唯一的工作空间，这些工作空间由适用于 Prometheus 的 Amazon 托管服务生成。

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。这将显示包括远程写入和查询的此工作空间的状态、ARN、工作空间 ID 和端点 URLs 等信息。

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 URLs 显示的终端节点 (远程写入 URL) 和端点-查询 URL。当您 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。

要在 Amazon EKS 中生成指标，例如 Kubernetes 或节点级指标，您可以使用 Amazon EKS 社区附加组件。有关更多信息，请参阅《Amazon EKS 用户指南》中的[可用的社区附加组件](#)。

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[将指标摄取到 Amazon Managed Service for Prometheus 工作区](#)。

Note

摄取到工作区的指标默认存储 150 天，然后自动删除。您可以通过将工作区配置为最长 1095 天（三年）来调整保留期。有关更多信息，请参阅[配置您的工作区](#)。

本部分中的说明有助于您快速启动并运行 Amazon Managed Service for Prometheus。假定您已经[创建了工作区](#)。在本节中，您将在 Amazon EKS 集群中设置新的 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 服务器将从该集群收集指标。
- Amazon EKS 集群必须安装 [Amazon EBS CSI 驱动程序](#)（Helm 要求）。
- 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。*prometheus-agent-namespace* 替换为您想要的这个命名空间的名称。

```
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，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。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。
 - *IAM_PROXY_PROMETHEUS_ROLE_ARN* 替换为您在中创建 `iam-proxy-ingest-role` 的 ARN。[设置服务角色从 Amazon EKS 集群中摄取指标](#)
 - *WORKSPACE_ID* 替换为适用于 Prometheus 的亚马逊托管服务工作空间的 ID。

- **REGION** 替换为适用于 Prometheus 的亚马逊托管服务工作区的区域。

```
## 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。
- **prometheus-agent-namespace** 替换为您的 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

要了解查询亚马逊托管服务的 Prometheus 指标或使用适用于 Prometheus 的亚马逊托管服务的其他方法，请参阅。 APIs [查询 Prometheus 指标](#)

本节假定您已经[创建了一个工作区](#)，并正在[将指标摄取到](#)其中。

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 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。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 Amazon Managed Service for Prometheus 工作区](#)
- [配置工作区](#)
- [编辑工作区别名](#)
- [查找您的 Amazon Managed Service for Prometheus 工作区详细信息，包括 ARN](#)
- [删除 Amazon Managed Service for Prometheus 工作区](#)

创建 Amazon Managed Service for Prometheus 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 Managed Service for Prometheus 控制台。

Note

如果您运行的是 Amazon EKS 集群，还可以使用 [AWS Controllers for Kubernetes](#) 创建新的工作区。

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。两个工作空间可以具有相同的别名，但所有工作空间都具有唯一的工作空间，这些工作空间由适用于 Prometheus 的 Amazon 托管服务生成。

（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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。两个工作空间可以具有相同的别名，但所有工作空间都具有唯一的工作空间，这些工作空间由适用于 Prometheus 的 Amazon 托管服务生成。

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。这将显示包括远程写入和查询的此工作空间的状态、ARN、工作空间 ID 和端点 URLs 等信息。

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。

记下显示的 URLs 终端节点（远程写入 URL）和 Endpoint-查询 URL。当您 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](#)。

配置工作区

您可以针对以下内容配置工作区：

- 定义标签集，并定义与您定义的标签集相匹配的活跃时间序列的限制。标签集是由一个或多个标签组成的集合，这些标签 name/value 对有助于为时间序列指标提供背景信息。

通过定义标签集和设置活跃时间序列限制，您可以将一个租户或源中的峰值限制为仅影响该租户或源。例如，如果您对标签集 `team=A env=prod` 设置 100 万个活跃时间序列限制，则如果与该标签集匹配的已摄取时间序列数量超过该限制，则只有与标签集匹配的时间序列才会受到节流。这样，其他租户或指标源不会受到影响。

有关 Prometheus 中的标签的更多信息，请参阅 [Data Model](#)。

- 设置保留期以定义数据在工作区中保留的天数。

配置工作区

1. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。
4. 选择工作区配置选项卡。
5. 要设置工作区的保留期，请在保留期部分中选择编辑。然后，以天为单位指定新的保留期。最长为 1095 天（三年）。
6. 要添加或修改标签集及其活跃序列限制，请在标签集部分中选择编辑。然后执行以下操作：
 - a. （可选）在默认存储桶限制中输入值，以便对可在工作区中摄取的最大活跃时间序列数设置限制，仅计入与任何已定义的标签集不匹配的时间序列。
 - b. 要定义标签集，请在活跃系列限制下输入新标签集的活跃时间序列限制。

然后，为将在标签集中使用的一个标签输入标签和值，并选择添加标签。

- c. （可选）要定义其他标签集，请选择添加另一个标签集并重复前面的步骤。
7. 在完成后，选择保存更改。

编辑工作区别名

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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。

查找您的 Amazon Managed Service for Prometheus 工作区详细信息，包括 ARN

您可以使用 AWS 管理控制台或 AWS CLI 查找 Amazon Managed Service for Prometheus 工作区的详细信息。

Console

使用 Amazon Managed Service for Prometheus 控制台查找工作区详细信息

1. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。这将显示有关工作区的详细信息，包括：
 - 当前状态 - 工作区的状态（例如活动）显示在状态下方。
 - ARN - 工作区 ARN 显示在 ARN 下方。
 - ID - 工作区 ID 显示在工作区 ID 下方。
 - URLs— 控制台显示多个 URLs 工作区的内容，包括 URLs 用于写入工作区或从工作区查询数据的。

Note

默认情况下，URLs 给定的是 IPv4 URLs。您也可以使用双堆栈（IPv4 并且 IPv6 受支持）。URLs 它们是一样的，但使用的是 `api.aws` 域，而不是默认的 `amazonaws.com`。例如，如果你看到以下内容（一个 IPv4 URL）：

```
https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-abcd1234-ef56-7890-ab12-example/api/v1/remote_write
```

你可以创建一个双堆栈（包括对的支持 IPv6），URL 如下：

```
https://aps-workspaces.us-east-1.api.aws/workspaces/ws-abcd1234-ef56-7890-ab12-example/api/v1/remote_write
```

本部分下方是一些选项卡，其中包含有关规则、警报管理器、日志、配置和标签的信息。

AWS CLI

要查找您的工作空间详细信息，请使用 AWS CLI

以下命令返回工作区的详细信息。必须将 *my-workspace-id* 替换为要获取详细信息的工作区的工作区 ID。

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

这将返回有关工作区的详细信息，包括：

- 当前状态 - 在 `statusCode` 属性中返回工作区的状态（如 ACTIVE）。
- ARN - 在 `arn` 属性中返回工作区 ARN。
- URLs— AWS CLI 返回 `prometheusEndpoint` 属性中工作空间的基本 URL。

Note

默认情况下，返回的网址是 IPv4 网址。您还可以在域中使用双栈网址（IPv4 且 IPv6 支持的）网址，`api.aws` 而不是默认网址 `amazonaws.com` 例如，如果你看到以下内容（一个 IPv4 URL）：

```
https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-abcd1234-ef56-7890-ab12-example/
```

你可以创建一个双堆栈（包括对的支持 IPv6），URL 如下：

```
https://aps-workspaces.us-east-1.api.aws/workspaces/ws-abcd1234-ef56-7890-ab12-example/
```

您也可以分别通过添加 `/api/v1/remote_write` 或 `/api/v1/query`，URLs 为工作区创建远程写入和查询。

删除 Amazon Managed Service for Prometheus 工作区

删除工作区会删除已摄取到其中的数据。

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](#)。

要使用删除工作区 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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。

将指标摄取到 Amazon Managed Service for Prometheus 工作区

必须先将指标摄取到 Amazon Managed Service for Prometheus 工作区，然后才能对这些指标进行查询或发出警报。本部分介绍如何设置将指标摄取到工作区。

Note

摄取到工作区的指标默认存储 150 天，然后自动删除。您可以通过将工作区配置为最长 1095 天（三年）来调整保留期。有关更多信息，请参阅[配置您的工作区](#)。

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 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

- [使用 AWS 托管收集器摄取指标](#)
- [客户托管收集器](#)

使用 AWS 托管收集器摄取指标

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

Amazon EKS 在集群中公开 API 服务器指标、`kube-controller-manager` 指标和 `kube-scheduler` 指标。在 Kubernetes 环境中运行的许多其他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 提供完全托管式无代理抓取器或收集器，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。Amazon Managed Service for Prometheus 托管收集器可与 Amazon EKS 集群（包括 EC2 和 Fargate）配合使用。

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）。收集器通过这些指标抓取指标 ENIs，然后使用 VPC 终端节点将数据推送 `remote_write` 到适用于 Prometheus 的亚马逊托管服务工作空间。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

- [为 Amazon EKS 设置托管式收集器](#)
- [为 Amazon MSK 设置托管式 Prometheus 收集器](#)
- [与 Prometheus 兼容的指标有哪些？](#)
- [使用已出售日志监控收集器](#)

为 Amazon EKS 设置托管式收集器

要使用 Amazon Managed Service for Prometheus 收集器，应创建一个抓取器来发现和提取 Amazon EKS 集群中的指标。还可以创建与 Amazon Managed Streaming for Apache Kafka 集成的抓取器。有关更多信息，请参阅 [与 Amazon MSK 集成](#)。

- 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

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？](#)。Amazon EKS 集群会公开 API 服务器的指标。Kubernetes 版本 1.28 或更高版本的 Amazon EKS 集群还公开 kube-scheduler 和 kube-controller-manager 的指标。有关更多信息，请参阅《Amazon EKS 用户指南》中的[获取 Prometheus 格式的控制面板原始指标](#)。

Note

从集群中抓取指标可能会产生网络使用费。优化这些成本的方法之一是配置您的 /metrics 端点，对提供的指标进行压缩（例如使用 gzip），从而减少必须在网络上传输的数据。如何做到这一点，取决于提供指标的应用程序或库。有些库默认使用 gzip 压缩。

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。

可通过三种方式创建抓取器：

- 当您[通过 Amazon EKS 控制台创建 Amazon EKS 集群](#)并选择开启 Prometheus 指标时，系统会自动为您创建抓取器。
- 您可以从 Amazon EKS 控制台为现有集群创建抓取器。在 [Amazon EKS 控制台](#) 中打开集群，然后在可观测性选项卡上，选择添加抓取器。

有关可用设置的更多详细信息，请参阅《Amazon EKS 用户指南》中的[开启 Prometheus 指标](#)。

- 您可以使用 AWS API 或 创建抓取工具 AWS CLI。

以下过程中描述了这些选项。

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。
- Amazon EKS 集群所在的 Amazon VPC 必须[启用了 DNS](#)。

Note

集群将通过其 Amazon 资源名称 (ARN) 与抓取器相关联。如果删除一个集群，然后创建一个同名的新集群，新集群将重新使用 ARN。因此，抓取器将尝试收集新集群的指标。[删除抓取器](#)与删除集群是分开的。

AWS API

使用 AWS API 创建抓取程序

使用 CreateScraper API 操作使用 AP AWS I 创建抓取工具。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信息 IDs，并提供用于抓取器的配置。

Note

安全组和子网应设置为所连接集群的安全组和子网。
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 scrapeConfiguration 格式的更多信息，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

使用 AWS CLI 创建抓取程序

通过 AWS CLI 使用 `create-scraper` 命令创建抓取器。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信息 IDs，并提供用于抓取器的配置。

Note

安全组和子网应设置为所连接集群的安全组和子网。
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 `scrape-configuration` 格式的更多信息，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 [CreateScraper](#) API 操作创建抓取程序。
- 使用 [ListScrapers](#) API 操作列出您现有的抓取程序。
- 使用 [UpdateScraper](#) API 操作更新抓取器的别名、配置或目的地。
- 使用 [DeleteScraper](#) API 操作删除抓取程序。
- 通过 [DescribeScraper](#) API 操作获取有关抓取程序的更多详细信息。
- 通过 [GetDefaultScraperConfiguration](#) API 操作获取抓取程序的通用配置。

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。

跨账户设置

要在 Amazon EKS 集群和 Amazon Managed Service for Prometheus 工作区位于不同的账户中时创建跨账户抓取器，请使用以下过程。例如，您有一个包含 Amazon EKS 集群的源账户 `account_id_source` 和一个包含 Amazon Managed Service for Prometheus 工作区的目标账户 `account_id_target`。

在跨账户设置中创建抓取器

1. 在源账户中，创建角色 `arn:aws:iam::account_id_source:role/Source` 并添加以下信任策略。

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

```

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

```

2. 在源 (Amazon EKS 集群) 和目标 (适用于 Prometheus 的亚马逊托管服务工作空间) 的每种组合上，您需要创建一个 `arn:aws:iam::account_id:target:role/Target` 角色并添加以下具有权限的信任策略。 [AmazonPrometheusRemoteWriteAccess](#)

```

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

```

3. 使用 `--role-configuration` 选项创建抓取器。

```

aws amp create-scraper \
  --source eksConfiguration="{clusterArn='arn:aws:eks:us-west-2:account-
  id_source:cluster/xarw,subnetIds=[subnet-subnet-id]}" \
  --scrape-configuration configurationBlob=<base64-encoded-blob> \

```

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

4. 验证抓取器创建。

```
aws amp list-scrapers
{
  "scrapers": [
    {
      "scrapersId": "scrapers-id",
      "arn": "arn:aws:aps:us-west-2:account-id_source:scrapers/scrapers-id",
      "roleArn": "arn:aws:iam::account-id_source:role/aws-service-role/scrapers.aps.amazonaws.com/AWSServiceRoleForAmazonPrometheusScrapersInternal_cc319052-41a3-4",
      "status": {
        "statusCode": "ACTIVE"
      },
      "createdAt": "2024-10-29T16:37:58.789000+00:00",
      "lastModifiedAt": "2024-10-29T16:55:17.085000+00:00",
      "tags": {},
      "source": {
        "eksConfiguration": {
          "clusterArn": "arn:aws:eks:us-west-2:account-id_source:cluster/xarw",
          "securityGroupIds": [
            "sg-security-group-id",
            "sg-security-group-id"
          ],
          "subnetIds": [
            "subnet-subnet-id"
          ]
        }
      },
      "destination": {
        "ampConfiguration": {
          "workspaceArn": "arn:aws:aps:us-west-2:account-id_target:workspace/ws-workspace-id"
        }
      }
    }
  ]
}
```

在 RoleConfiguration 和服务相关角色之间切换

当您想要切换回服务相关角色而不是 RoleConfiguration 以写入 Amazon Managed Service for Prometheus 工作区时，必须更新 UpdateScraper 并提供一个与抓取器位于相同账户中的工作区（不需要 RoleConfiguration）。系统将从抓取器中移除 RoleConfiguration，并将使用服务相关角色。

当您更改与抓取器位于相同账户中的工作区并且想要继续使用 RoleConfiguration 时，必须再次在 UpdateScraper 上提供 RoleConfiguration。

为启用了客户自主管理型密钥的工作区创建抓取器

要创建抓取器以便将指标摄取到具有[客户自主管理型密钥](#)的 Amazon Managed Service for Prometheus 工作区，请使用 `--role-configuration` 并将源和目标都设置为同一个账户。

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

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。在传入 CreateScraper API 的每个子网中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 访问条目自动提供 Amazon Managed Service for Prometheus 收集器对集群的访问权限。
- 手动配置 Amazon EKS 集群，以便进行托管指标抓取。

以下主题将对每个主题进行详细介绍。

将 Amazon EKS 配置为使用访问条目进行抓取器访问

使用 Amazon EKS 的访问条目是让 Amazon Managed Service for Prometheus 从集群中抓取指标的最简单方法。

必须将您要抓取的 Amazon EKS 集群配置为支持 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《Amazon EKS 用户指南》中的[支持 IAM 角色或用户访问 Amazon EKS 集群上的 Kubernetes 对象](#)。

您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 Amazon EKS 控制台创建 Amazon EKS 集群](#)时配置此访问权限（按照说明创建作为集群一部分的抓取器），然后将自动创建访问条目策略，使 Amazon Managed Service for Prometheus 能够访问集群指标。
- 创建集群后添加：如果 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，您[通过 Amazon Managed Service for Prometheus API 或 CLI](#)或通过 Amazon EKS 控制台创建的任何抓取器会自动为您创建正确的访问条目策略，而抓取器将有权访问您的集群。

已创建访问条目策略

当您创建抓取器并让 Amazon Managed Service for Prometheus 为您生成访问条目策略时，它会生成以下策略。有关访问条目的更多信息，请参阅《Amazon EKS 用户指南》中的[支持 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",
    "apiGroups": [
      "metrics.eks.amazonaws.com"
    ],
    "resources": [
      "kcm/metrics",
      "ksh/metrics"
    ],
    "verbs": [
      "get"
    ]
  }
]
```

```
    },
    {
      "effect": "allow",
      "nonResourceURLs": [
        "/metrics"
      ],
      "verbs": [
        "get"
      ]
    }
  ]
}
```

手动配置 Amazon EKS 以进行抓取器访问

如果您更喜欢使用 `aws-auth` ConfigMap 来控制对 kubernetes 集群的访问，您仍然可以让 Amazon Managed Service for Prometheus 抓取器访问您的指标。以下步骤将支持 Amazon Managed Service for Prometheus 从您的 Amazon EKS 集群抓取指标。

Note

有关 ConfigMap 和访问条目的更多信息，请参阅《Amazon EKS 用户指南》中的[支持 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 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"]
- apiGroups: ["metrics.eks.amazonaws.com"]
  resources: ["kcm/metrics", "ksh/metrics"]
  verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aps-collector-user-role-binding
subjects:
- kind: User
  name: aps-collector-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aps-collector-role
  apiGroup: rbac.authorization.k8s.io

```

2. 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带有 ID 的抓取器的信息 *scraper-id*。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/AWSServiceRoleForAmazonPrometheusScrapper_unique-id
```

例如，此 ARN：

```
arn:aws:iam::111122223333:role/aws-service-role/scrapper.aps.amazonaws.com/  
AWSServiceRoleForAmazonPrometheusScrapper_1234abcd-56ef-7
```

必须重新编写为：

```
arn:aws:iam::111122223333:role/  
AWSServiceRoleForAmazonPrometheusScrapper_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征和功能，以及安全更新。或者，使用 [AWS CloudShell](#)，它可以自动提供始终使用 up-to-date 命令行体验。

要列出您账户中的所有抓取程序，请使用 [ListScrapers](#) API 操作。

或者，使用 AWS CLI，拨打：

```
aws amp list-scrapers --region aws-region
```

`ListScrapers` 返回您账户中的所有抓取程序，例如：

```
{
```

```
"scrapers": [
  {
    "scrapeId": "s-1234abcd-56ef-7890-abcd-1234ef567890",
    "arn": "arn:aws:aps:us-west-2:123456789012:scrape/s-1234abcd-56ef-7890-
abcd-1234ef567890",
    "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
AWSServiceRoleForAmazonPrometheusScrape_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"
      }
    }
  }
]
```

要删除抓取程序，请使用 `ListScrapers` 操作查找要删除的抓取程序的 `scrapeId`，然后使用 [DeleteScrape](#) 操作将其删除。

或者，使用 AWS CLI，拨打：

```
aws amp delete-scrape --scrape-id scrapeId
```

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 `GetDefaultScrapeConfiguration` 操作下载通用配置文件。

要修改抓取器的配置，您可以使用 `UpdateScrape` 操作。如果您需要更新指标的源（例如，更新到不同的 Amazon EKS 集群），则必须删除抓取器，然后使用新的源重新创建它。

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 [Configuration](#)。全局配置选项和 `<scrape_config>` 选项描述了最常用的选项。

由于 Amazon EKS 是唯一受支持的服务，因此唯一受支持的服务发现配置（`<*_sd_config>`）是 `<kubernetes_sd_config>`。

支持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。此示例包括对 kube API 服务器指标的支持，`kube-controller-manager` 以及对 `kube-scheduler` 指标的支持。有关更多信息，请参阅《Amazon EKS 用户指南》中的 [获取 Prometheus 格式的控制面板原始指标](#)。

```
global:
  scrape_interval: 30s
```

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

```
source_labels:
- __meta_kubernetes_namespace
- __meta_kubernetes_pod_name
separator: '/'
regex: 'kube-system/kube-proxy.+'
```

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

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

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

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。
- DNS 解析 – 与目标名称相关，此配置中唯一可识别的服务器名称是 Kubernetes api 服务器 `kubernetes.default.svc`。所有其他计算机名称必须通过 IP 地址指定。
- 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？

Important

验证已为 Amazon EKS 集群启用私有访问。有关更多信息，请参阅《Amazon EKS 用户指南》中的[集群私有端点](#)。

`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。

在确定端点存在的前提下，有若干原因可能导致收集器无法找到端点。

- 可能需要调整抓取配置。可能需要调整发现 `relabel_config`。
- 用于发现的 `role` 可能存在问题。
- Amazon EKS 集群使用的 Amazon VPC 可能未[启用 DNS](#)，这会使收集器无法找到端点。

- 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。您可以验证详细信息是否正确，例如使用的协议 (http 或 https)、端点或端口。您还可以检查端点是否以有效的 200 响应进行响应，并遵循 Prometheus 格式。最后，响应正文不能大于支持的最大大小。(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 count(up)，通过检查 up 指标计数来检查正在抓取的指标端点的数量。

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。
- 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。

为 Amazon MSK 设置托管式 Prometheus 收集器

要使用 Amazon Managed Service for Prometheus 收集器，您必须创建一个抓取器，用于发现和提取 Amazon Managed Streaming for Apache Kafka 集群中的指标。还可以创建与 Amazon Elastic Kubernetes Service 集成的抓取器。有关更多信息，请参阅[集成 Amazon EKS](#)。

创建抓取程序

Amazon Managed Service for Prometheus 收集器由一个抓取器组成，该抓取器用于发现和收集 Amazon MSK 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。

您可以使用 AWS API 或按以下步骤 AWS CLI 所述创建抓取器。

创建您自己的抓取程序时有以下几个先决条件：

- 您必须先创建 Amazon MSK 集群。
- 将 Amazon MSK 集群的安全组配置为支持 Amazon VPC 内的端口 11001 (JMX 导出程序) 和 11002 (节点导出程序) 上的入站流量，因为抓取器需要访问这些 DNS 记录才能收集 Prometheus 指标。
- Amazon MSK 集群所在的 Amazon VPC 必须[启用了 DNS](#)。

Note

集群将通过其 Amazon 资源名称 (ARN) 与抓取器相关联。如果删除一个集群，然后创建一个同名的新集群，新集群将重新使用 ARN。因此，抓取器将尝试收集新集群的指标。[删除抓取器](#)与删除集群是分开的。

To create a scraper using the AWS API

使用 CreateScraper API 操作使用 AP AWS I 创建抓取工具。以下示例在美国东部 (弗吉尼亚州北部) 区域中创建抓取器。将 *example* 内容替换为您的 Amazon MSK 集群信息，并提供您的抓取器配置。

Note

配置安全组和子网以匹配您的目标集群。至少包含跨两个可用区的两个子网。

```
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-east-1:123456789012:workspace/ws-
workspace-id"
    }
  },
  "source": {
    "vpcConfiguration": {
      "securityGroupIds": ["sg-security-group-id"],
      "subnetIds": ["subnet-subnet-id-1", "subnet-subnet-id-2"]
    }
  },
  "scrapeConfiguration": {
    "configurationBlob": base64-encoded-blob
  }
}
```

在示例中，`scrapeConfiguration` 参数需要一个 base64 编码的 Prometheus 配置 YAML 文件，该文件指定 MSK 集群的 DNS 记录。

每个 DNS 记录都代表特定可用区域中的代理终端节点，允许客户端连接到分布在您选择 AZs 的代理以实现高可用性。

MSK 集群属性中的 DNS 记录数量与集群配置中代理节点和可用区的数量相对应：

- 默认配置-3 个 DNS 记录中的 3 个代理节点 AZs = 3 个 DNS 记录
- 自定义配置 — 2 个 Broker 节点，跨越 AZs 2 个 DNS 记录

[要获取 MSK 集群的 DNS 记录，请在家中打开 MSK 控制台？https://console.aws.amazon.com/msk/region=us-east-1#/home/](https://console.aws.amazon.com/msk/region=us-east-1#/home/)。转到 MSK 集群。选择属性、代理和端点。

您可以通过两个选项将 Prometheus 配置为从 MSK 集群中抓取指标：

1. 集群级 DNS 解析（推荐）：使用集群的基本 DNS 名称来自动发现所有代理。如果代理端点为 `b-1.clusterName.xxx.xxx.xxx`，请将 `clusterName.xxx.xxx.xxx` 用作 DNS 记录。这可使 Prometheus 自动抓取集群中的所有代理。

各个代理端点：单独指定每个代理端点以进行精细控制。在配置中使用完整的代理标识符（b-1、b-2）。例如：

```
dns_sd_configs:
  - names:
    - b-1.clusterName.xxx.xxx.xxx
    - b-2.clusterName.xxx.xxx.xxx
    - b-3.clusterName.xxx.xxx.xxx
```

Note

使用 AWS 控制 `clusterName.xxx.xxx.xxx` 台中的实际 MSK 集群终端节点替换。

有关更多信息，请参阅 Prometheus 文档中的 [<dns_sd_config>](#)。

下面是抓取器配置文件的示例：

```
global:
  scrape_interval: 30s
  external_labels:
    clusterArn: msk-test-1

scrape_configs:
  - job_name: msk-jmx
    scheme: http
    metrics_path: /metrics
    scrape_timeout: 10s
    dns_sd_configs:
      - names:
        - dns-record-1
        - dns-record-2
        - dns-record-3
        type: A
        port: 11001
    relabel_configs:
      - source_labels: [__meta_dns_name]
        target_label: broker_dns
      - source_labels: [__address__]
        target_label: instance
```

```

    regex: '(.*)'
    replacement: '${1}'

- job_name: msk-node
  scheme: http
  metrics_path: /metrics
  scrape_timeout: 10s
  dns_sd_configs:
    - names:
      - dns-record-1
      - dns-record-2
      - dns-record-3
      type: A
      port: 11002
  relabel_configs:
    - source_labels: [__meta_dns_name]
      target_label: broker_dns
    - source_labels: [__address__]
      target_label: instance
      regex: '(.*)'
      replacement: '${1}'

```

运行以下命令之一来将 YAML 文件转换为 base64。也可以使用任何在线 base64 转换器来转换文件。

Example Linux/macOS

```
echo -n scraper config updated with dns records | base64
```

Example 窗戶 PowerShell

```
[Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes(scraper config updated with dns records))
```

To create a scraper using the AWS CLI

通过 AWS Command Line Interface 使用 `create-scraper` 命令创建抓取器。以下示例在美国东部（弗吉尼亚州北部）区域中创建抓取器。将 *example* 内容替换为您的 Amazon MSK 集群信息，并提供您的抓取器配置。

Note

配置安全组和子网以匹配您的目标集群。至少包含跨两个可用区的两个子网。

```
aws amp create-scraper \
  --source vpcConfiguration="{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-2:123456789012:workspace/ws-workspace-id'}"
```

- 以下是您可以与 AWS API 一起使用的抓取器操作的完整列表：

使用 [CreateScrapper](#) API 操作创建抓取程序。

- 使用 [ListScrapers](#) API 操作列出您现有的抓取程序。
- 使用 [UpdateScrapper](#) API 操作更新抓取器的别名、配置或目的地。
- 使用 [DeleteScrapper](#) API 操作删除抓取程序。
- 通过 [DescribeScrapper](#) API 操作获取有关抓取程序的更多详细信息。

跨账户设置

要在跨账户设置中创建抓取器，而您要从中收集指标的 Amazon MSK 集群与 Amazon Managed Service for Prometheus 收集器位于不同的账户中，请使用以下过程。

例如，当您有两个账户时，第一个账户是 Amazon MSK 所在的源账户 `account_id_source`，另一个是 Amazon Managed Service for Prometheus 工作区所在的目标账户 `account_id_target`。

在跨账户设置中创建抓取器

1. 在源账户中，创建角色 `arn:aws:iam::111122223333:role/Source` 并添加以下信任策略。

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "scraper.aps.amazonaws.com"
    ]
  }
}
```

```

    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnEquals": {
        "aws:SourceArn": "arn:aws:aps:aws-region:111122223333:scraper/scraper-id"
      },
      "StringEquals": {
        "AWS:SourceAccount": "111122223333"
      }
    }
  }
}

```

2. 在源 (Amazon MSK 集群) 和目标 (适用于 Prometheus 的亚马逊托管服务工作空间) 的每种组合上，您都需要创建一个 `arn:aws:iam::444455556666:role/Target` 角色并添加以下具有权限的信任策略。[AmazonPrometheusRemoteWriteAccess](#)

```

{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Source"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "StringEquals": {
      "sts:ExternalId": "arn:aws:aps:aws-region:111122223333:scraper/scraper-id"
    }
  }
}

```

3. 使用 `--role-configuration` 选项创建抓取器。

```

aws amp create-scraper \
  --source vpcConfiguration="{subnetIds=[subnet-subnet-id], "securityGroupIds": ["sg-security-group-id"]}" \
  --scrape-configuration configurationBlob=<base64-encoded-blob> \
  --destination ampConfiguration="{workspaceArn='arn:aws:aps:aws-region:444455556666:workspace/ws-workspace-id'}" \
  --role-configuration

```

```
'{"sourceRoleArn":"arn:aws:iam::111122223333:role/Source",  
"targetRoleArn":"arn:aws:iam::444455556666:role/Target"}'
```

4. 验证抓取器创建。

```
aws amp list-scrapers  
{  
  "scrapers": [  
    {  
      "scrapersId": "s-example123456789abcdef0",  
      "arn": "arn:aws:aps:aws-region:111122223333:scraper/s-  
example123456789abcdef0": "arn:aws:iam::111122223333:role/Source",  
      "status": "ACTIVE",  
      "creationTime": "2025-10-27T18:45:00.000Z",  
      "lastModificationTime": "2025-10-27T18:50:00.000Z",  
      "tags": {},  
      "statusReason": "Scraper is running successfully",  
      "source": {  
        "vpcConfiguration": {  
          "subnetIds": ["subnet-subnet-id"],  
          "securityGroupIds": ["sg-security-group-id"]  
        }  
      },  
      "destination": {  
        "ampConfiguration": {  
          "workspaceArn": "arn:aws:aps:aws-region:444455556666:workspace/  
ws-workspace-id"  
        }  
      },  
      "scrapeConfiguration": {  
        "configurationBlob": "<base64-encoded-blob>"  
      }  
    }  
  ]  
}
```

在 RoleConfiguration 和服务相关角色之间切换

当您想要切换回服务相关角色而不是 RoleConfiguration 以写入 Amazon Managed Service for Prometheus 工作区时，必须更新 UpdateScraper 并提供一个与抓取器位于相同账户中的工作区（不需要 RoleConfiguration）。系统将从抓取器中移除 RoleConfiguration，并将使用服务相关角色。

当您更改与抓取器位于相同账户中的工作区并且想要继续使用 RoleConfiguration 时，必须再次提供在 UpdateScraper 上提供 RoleConfiguration。

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征和功能，以及安全更新。或者，使用 [AWS CloudShell](#)，它可以自动提供始终使用 up-to-date 命令行体验。

要列出您账户中的所有抓取程序，请使用 [ListScrapers](#) API 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:aws-region: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": {
      "vpcConfiguration": {
        "securityGroupIds": [
          "sg-1234abcd5678ef90"
        ],
        "subnetIds": [
          "subnet-abcd1234ef567890",
          "subnet-1234abcd5678ab90"
        ]
      }
    },
    "destination": {
      "ampConfiguration": {
        "workspaceArn": "arn:aws:aps:aws-region:123456789012:workspace/
ws-1234abcd-5678-ef90-ab12-cdef3456a78"
      }
    }
  }
]
}

```

要删除抓取程序，请使用 `ListScrapers` 操作查找要删除的抓取程序的 `scraperId`，然后使用 [DeleteScraper](#) 操作将其删除。

或者，使用 AWS CLI，拨打：

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

从 Amazon MSK 收集的指标

当与 Amazon MSK 集成时，Amazon Managed Service for Prometheus 收集器会自动抓取以下指标：

指标：`jmx_exporter` 和 `pod_exporter` 作业

指标	描述/用途
<code>jmx_config_reload_failure_total</code>	JMX 导出程序未能重新加载其配置文件的总次数。
<code>jmx_scrape_duration_seconds</code>	在当前收集周期内抓取 JMX 指标所花费的时间（以秒为单位）。

指标	描述/用途
jmx_scrape_error	指示在 JMX 指标抓取期间是否出现错误 (1 = 错误, 0 = 成功)。
java_lang_memory_ _used HeapMemoryUsage	JVM 当前使用的堆内存量 (以字节为单位)。
java_lang_memory_ _max HeapMemoryUsage	可用于内存管理的最大堆内存量 (以字节为单位)。
java_lang_memory_ _used NonHeapMemoryUsage	JVM 当前使用的非堆内存量 (以字节为单位)。
kafka_cluster_Partition_Value	与 Kafka 集群分区相关的当前状态或值, 按分区 ID 和主题细分。
kafka_consumer_consumer_coordinator_metrics_assigned_partitions	当前分配给该使用者的分区数。
kafka_consumer_consumer_coordinator_metrics_commit_latency_avg	提交偏移量所花费的平均时间 (以毫秒为单位)。
kafka_consumer_consumer_coordinator_metrics_commit_rate	每秒的偏移量提交次数。
kafka_consumer_consumer_coordinator_metrics_failed_rebalance_total	失败的使用者组重新平衡总数。
kafka_consumer_consumer_coordinator_metrics_last_heartbeat_seconds_ago	自上次向协调器发送检测信号以来的秒数。
kafka_consumer_consumer_coordinator_metrics_rebalance_latency_avg	使用者组重新平衡所花费的平均时间 (以毫秒为单位)。
kafka_consumer_consumer_coordinator_metrics_rebalance_total	使用者组重新平衡总数。
kafka_consumer_consumer_fetch_manager_metrics_bytes_consumed_rate	使用者每秒使用的平均字节数。

指标	描述/用途
kafka_consumer_consumer_fetch_manager_metrics_fetch_latency_avg	提取请求所花费的平均时间（以毫秒为单位）。
kafka_consumer_consumer_fetch_manager_metrics_fetch_rate	每秒的提取请求数。
kafka_consumer_consumer_fetch_manager_metrics_records_consumed_rate	每秒使用的平均记录数。
kafka_consumer_consumer_fetch_manager_metrics_records_lag_max	最大滞后，以该使用者中的任何分区的记录数表示。
kafka_consumer_consumer_metrics_connection_count	当前活动的连接数。
kafka_consumer_consumer_metrics_incoming_byte_rate	每秒从所有服务器接收的平均字节数。
kafka_consumer_consumer_metrics_last_poll_seconds_ago	自上次使用者 poll() 调用以来的秒数。
kafka_consumer_consumer_metrics_request_rate	每秒发送的请求数。
kafka_consumer_consumer_metrics_response_rate	每秒接收的响应数。
kafka_consumer_consumer_group_consumer_lag_metrics_val	使用者组的当前使用者滞后值，表示使用者落后的程度。
kafka_consumer_controller_val	Kafka 控制器的当前状态或值（1 = 活动控制器，0 = 不活动）。
kafka_consumer_controller_event_manager_val	已处理的控制器事件总数。
kafka_consumer_controller_event_manager_val_mean	处理控制器事件所花费的平均时间。

指标	描述/用途
kafka_controller_StatsController_MeanRate	每秒控制器统计操作的平均速率。
kafka_coordinator_group_metadata_manager_Value	使用者组的组元数据管理器的当前状态或值。
kafka_log_flush_stats_log_count	日志刷新操作总数。
kafka_log_flush_stats_log_mean	日志刷新操作所花费的平均时间。
kafka_log_flush_stats_log_mean_rate	每秒日志刷新操作的平均速率。
kafka_network_request_metrics_count	已处理的网络请求总数。
kafka_network_request_metrics_mean	处理网络请求所花费的平均时间。
kafka_network_request_metrics_mean_rate	每秒网络请求的平均速率。
kafka_network_accept_mean_rate	每秒接受的连接的平均速率。
kafka_server_fetch_queue_size	提取请求队列的当前大小。
kafka_server_produce_queue_size	生产请求队列的当前大小。
kafka_server_request_queue_size	常规请求队列的当前大小。
kafka_broker_topic_metrics_server_count	代理主题操作（消息in/out, bytes in/out）的总数。
kafka_broker_topic_metrics_server_mean_rate	每秒代理主题操作的平均速率。
kafka_broker_topic_metrics_server_one_minute_rate	代理主题操作的一分钟移动平均速率。
kafka_delayed_operation_purgatory_server_value	处于等待状态（等待完成）的延迟操作的当前数量。
kafka_delayed_fetch_metrics_server_mean_rate	每秒延迟提取操作的平均速率。

指标	描述/用途
kafka_FetcherLagMetrics server__Valu	副本提取器线程的当前滞后值（落后于领导者线程的程度）。
kafka_FetcherStats server__ MeanRate	每秒提取器操作的平均速率。
kafka_ReplicaManager server__Valu	副本管理器的当前状态或值。
kafka_ReplicaManager server__ MeanRate	每秒副本管理器操作的平均速率。
kafka_LeaderReplication server__byte_rate	对于此代理作为领导者的分区，每秒复制的字节速率。
kafka_server_group_coordinator_metrics_group_completed_rebalance_count	完成的使用者组重新平衡的总数。
kafka_server_group_coordinator_metrics_offset_commit_count	偏移量提交操作的总数。
kafka_server_group_coordinator_metrics_offset_commit_rate	每秒偏移量提交操作的速率。
kafka_server_socket_server_metrics_connection_count	当前活动的连接数。
kafka_server_socket_server_metrics_connection_creation_rate	每秒新建连接的速率。
kafka_server_socket_server_metrics_connection_close_rate	每秒连接关闭的速率。
kafka_server_socket_server_metrics_failed_authentication_total	失败的身份验证尝试总数。
kafka_server_socket_server_metrics_incoming_byte_rate	每秒传入字节数。
kafka_server_socket_server_metrics_outgoing_byte_rate	每秒传出字节数。

指标	描述/用途
kafka_server_socket_server_metrics_request_rate	每秒请求速率。
kafka_server_socket_server_metrics_response_rate	每秒响应速率。
kafka_server_socket_server_metrics_network_io_rate	每秒网络 I/O 操作的速率。
kafka_server_socket_server_metrics_io_ratio	花在 I/O 操作上的时间的一小部分。
kafka_server_controller_channel_metrics_connection_count	控制器通道的当前活动连接数。
kafka_server_controller_channel_metrics_incoming_byte_rate	控制器通道每秒传入字节的速率。
kafka_server_controller_channel_metrics_outgoing_byte_rate	控制器通道每秒传出字节的速率。
kafka_server_controller_channel_metrics_request_rate	控制器通道每秒的请求速率。
kafka_server_replica_fetcher_metrics_connection_count	副本提取器的当前活动连接数。
kafka_server_replica_fetcher_metrics_incoming_byte_rate	副本提取器的每秒传入字节的速率。
kafka_server_replica_fetcher_metrics_request_rate	副本提取器的每秒请求速率。
kafka_server_replica_fetcher_metrics_failed_authentication_total	副本提取器的失败的身份验证尝试总数。
kafka_ZooKeeperClientMetrics_server__count	ZooKeeper 客户机操作总数。
kafka_ZooKeeperClientMetrics_server__Mean	ZooKeeper 客户端操作的平均延迟。

指标	描述/用途
kafka_KafkaServer server_ _Valu	Kafka 服务器的当前状态或值 (通常表示服务器正在运行) 。
node_cpu_seconds_total	按照 CPU 和模式细分的每种模式 (用户、系统、空闲等) 所 CPUs 花费的总秒数。
node_disk_read_bytes_total	成功从磁盘读取的总字节数，按设备细分。
node_disk_reads_completed_total	成功完成的磁盘读取总数，按设备细分。
node_disk_writes_completed_total	成功完成的磁盘写入总数，按设备细分。
node_disk_written_bytes_total	成功写入磁盘的总字节数，按设备细分。
node_filesystem_avail_bytes	非根用户的可用文件系统空间 (以字节为单位) ，按设备和挂载点细分。
node_filesystem_size_bytes	文件系统的总大小 (以字节为单位) ，按设备和挂载点细分。
node_filesystem_free_bytes	可用文件系统空间 (以字节为单位) ，按设备和挂载点细分。
node_filesystem_files	文件系统上的文件节点 (inode) 总数，按设备和挂载点细分。
node_filesystem_files_free	文件系统上空闲文件节点 (inode) 的数量，按设备和挂载点细分。
node_filesystem_readonly	表示文件系统是否以只读方式挂载 (1 = 只读，0 = 读写) 。
node_filesystem_device_error	表示在获取文件系统统计信息时是否出现错误 (1 = 错误，0 = 成功) 。

限制

当前 Amazon MSK 与 Amazon Managed Service for Prometheus 的集成存在以下限制：

- 仅对于 Amazon MSK 预置集群才支持（不适用于 Amazon MSK Serverless）
- 不支持同时启用公共访问和 KRaft 元数据模式的 Amazon MSK 集群
- 对于 Amazon MSK Express 代理不支持
- 目前支持 Amazon MSK 集群与 Amazon Managed Service for Prometheus 收集器/工作区之间的 1:1 映射

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 [为 Amazon EKS 设置托管式收集器](#)。

使用已出售日志监控收集器

Amazon Managed Service for Prometheus 收集器提供已出售日志，有助于您监控指标收集过程并对其进行故障排除。这些日志会自动发送到 Amazon CloudWatch Logs，并提供对服务发现、指标收集和数据导出操作的可见性。收集器会针对指标收集管道的三个主要组件出售日志：

主题

- [服务发现日志](#)
- [收集器日志](#)
- [导出程序日志](#)
- [了解和使用收集器已出售日志](#)

服务发现日志

服务发现日志提供有关目标发现过程的信息，包括：

- 访问 Kubernetes API 资源时的身份验证或权限问题。
- 服务发现设置中的配置错误。

以下示例演示了在服务发现过程中可能遇到的常见身份验证和权限错误：

Amazon EKS 集群不存在

当指定的 Amazon EKS 集群不存在时，您会收到以下错误：

```
{
  "component": "SERVICE_DISCOVERY",
  "timestamp": "2025-04-30T17:25:41.946Z",
  "message": {
    "log": "Failed to watch Service - Verify your scraper source exists."
  },
  "scrapeConfigId": "s-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

服务的权限无效

当收集器缺少适当的基于角色的访问控制 (RBAC) 权限以监视服务时，您会收到以下错误：

```
{
  "component": "SERVICE_DISCOVERY",
  "timestamp": "2025-04-30T17:25:41.946Z",
  "message": {
    "log": "Failed to watch Service - Verify your scraper source permissions are valid."
  },
  "scrapeConfigId": "s-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

端点的权限无效

当收集器缺少适当的基于角色的访问控制 (RBAC) 权限以监视端点时，您会收到以下错误：

```
{
```

```
"component": "SERVICE_DISCOVERY",
"timestamp": "2025-04-30T17:25:41.946Z",
"message": {
  "log": "Failed to watch Endpoints - Verify your scraper source permissions are
valid."
},
"scrapeConfigId": "s-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

收集器日志

收集器日志提供有关指标抓取过程的信息，包括：

- 由于端点不可用而导致抓取失败。
- 尝试抓取目标时出现连接问题。
- 抓取操作期间超时。
- 抓取目标返回的 HTTP 状态错误。

以下示例演示了在指标抓取过程中可能遇到的常见收集器错误：

缺少指标端点

当 `/metrics` 端点在目标实例上不可用时，您会收到以下错误：

```
{
  "component": "COLLECTOR",
  "message": {
    "log": "Failed to scrape Prometheus endpoint - verify /metrics endpoint is
available",
    "job": "pod_exporter",
    "targetLabels": "{\"__name__=\\"up\\", instance=\10.24.34.0\\", job=
\\\"pod_exporter\\\"}"
  },
  "timestamp": "1752787969551",
  "scraperId": "s-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

连接被拒绝

当收集器无法与目标端点建立连接时，您会收到以下错误：

```
{
  "scrapeConfigId": "s-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "timestamp": "2025-04-30T17:25:41.946Z",
  "message": {
    "message": "Scrape failed",
    "scrape_pool": "pod_exporter",
    "target": "http://10.24.34.0:80/metrics",
    "error": "Get \"http://10.24.34.0:80/metrics\": dial tcp 10.24.34.0:80: connect:
connection refused"
  },
  "component": "COLLECTOR"
}
```

导出程序日志

导出程序日志提供有关将收集到的指标发送到 Amazon Managed Service for Prometheus 工作区的过程的信息，包括：

- 处理的指标和数据点数量。
- 由于工作区问题而导致导出失败。
- 尝试写入指标时出现权限错误。
- 导出管道中的依赖关系失败。

以下示例演示了在指标导出过程中可能遇到的常见导出程序错误：

找不到工作区

当找不到指标导出的目标工作区时，您会收到以下错误：

```
{
  "component": "EXPORTER",
  "message": {
    "log": "Failed to export to the target workspace - Verify your scraper
destination.",
    "samplesDropped": 5
  },
  "timestamp": "1752787969664",
  "scraperId": "s-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

了解和使用收集器已出售日志

日志结构

所有收集器已出售日志都遵循一致的结构，其中包含以下字段：

scrapeConfigId

生成日志的抓取配置的唯一标识符。

timestamp

生成日志条目的时间。

message

日志消息内容，其中可能包括其他结构化字段。

组件

生成日志的组件 (SERVICE_DISCOVERY、COLLECTOR 或 EXPORTER)

使用已出售日志进行故障排除

收集器已出售日志有助于您解决指标收集中的常见问题：

1. 服务发现问题

- 检查 SERVICE_DISCOVERY 日志中是否存在身份验证或权限错误。
- 验证收集器是否拥有访问 Kubernetes 资源的必要权限。

2. 指标抓取问题

- 检查 COLLECTOR 日志中是否存在抓取失败。
- 验证目标端点是否可访问并返回指标。
- 确保防火墙规则支持收集器连接到目标端点。

3. 指标导出问题

- 检查 EXPORTER 日志是否存在导出失败。
- 确认工作区存在并已正确配置。
- 确保收集器具有写入工作区的必要权限。

访问收集器已出售日志

收集器出售的日志会自动发送到 Amazon CloudWatch 日志。访问这些日志：

1. 打开 CloudWatch 控制台，网址为 <https://console.aws.amazon.com/cloudwatch/>。
2. 在导航窗格中，选择日志组。
3. 查找并选择收集器的日志组：/aws/prometheus/workspace_id/collector/collector_id。
4. 浏览或搜索日志事件以查找相关信息。

您还可以使用 CloudWatch Logs Insights 来查询和分析您的收集器日志。例如，查找所有服务发现错误：

```
fields @timestamp, message.message
| filter component = "SERVICE_DISCOVERY" and message.message like /Failed/
| sort @timestamp desc
```

用于监控收集器的最佳实践

有效地监控 Amazon Managed Service for Prometheus 收集器：

1. 为收集器的关键问题设置 CloudWatch 警报，例如持续的抓取失败或导出错误。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的 [警报](#)。
2. 创建 CloudWatch 仪表板以可视化收集器性能指标以及销售的日志数据。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的 [控制面板](#)。
3. 定期查看服务发现日志，以确保正确地发现目标。
4. 监控已丢弃目标的数量，以确定潜在的配置问题。
5. 跟踪导出失败情况，以确保指标成功发送到工作区。

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](#)
- [使用 AWS Distro OpenTelemetry 作为收藏家](#)
- [使用 Prometheus 实例作为收集器](#)
- [设置 Amazon Managed Service for Prometheus 以获取高可用性数据](#)

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 可确保来自您的网络流量在 AWS 网络中 VPCs 受到保护，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 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（具有相应权限）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

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

以下主题介绍了将 ADOT 设置为指标收集器的三种不同方法，具体取决于指标是来自 Amazon EKS、Amazon ECS 还是 Amazon EC2 实例。

主题

- [使用 AWS Distro 在亚马逊 Elastic Kubernetes Service 集群 OpenTelemetry 上设置指标提取](#)
- [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](#)

使用 AWS Distro 在亚马逊 Elastic Kubernetes Service 集群 OpenTelemetry 上设置指标提取

您可以使用 AWS Distro for OpenTelemetry (ADOT) 收集器从装有 Prometheus 工具的应用程序中获取指标，然后将这些指标发送到适用于 Prometheus 的亚马逊托管服务。

Note

有关 ADOT 收集器的更多信息，请参阅[AWS 发行版](#)。OpenTelemetry

有关 Prometheus 分析的应用程序的更多信息，请参阅[与 Prometheus 兼容的指标有哪些？](#)。

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。使用 <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:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
```

```

    "Condition": {
      "StringEquals": {
        "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:adot-
col:amp-iamproxy-ingest-service-account",
        "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
      }
    }
  }
]
}

```

6. 选择权限选项卡，并确保将以下权限策略附加到该角色。

JSON

```

{
  "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](https://github.com/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。

您还需要 `YOUR_ACCOUNT_ID` 在 Kubernetes 配置的服务账户部分更改为您的 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。如果您想抓取其他示例应用程序，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`。](#)

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 适用于 Go 的 AWS SDK](#)。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适用于 Go 的 AWS SDK 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管 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](#)。

步骤 1：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` 替换为你的 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]

```

步骤 2：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。*my-region* 用你的 region 价值替换。

```

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

```

5. 推送容器映像。

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

步骤 3：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"
}
```

步骤 4：授予访问 Amazon Managed Service for Prometheus 的任务权限

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。

步骤 5：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. 使用 <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。

- 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。
- (可选) 为您的 IAM 角色创建 IAM 标签。选择下一步：审核。
- 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，您可以运行一个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

- 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)
}
```

- 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
```

- 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`，请 *my-region* 替换为您的“区域”值和 *my-workspace-id* 亚马逊 Prometheus 托管服务为您生成的工作空间 ID。对于 `region`，请 *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

您可以使用在代理模式下运行的 Prometheus 实例（称为 Prometheus 代理）来抓取指标并将其发送到您的 Amazon Managed Service for Prometheus 工作区。

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

创建 Prometheus 代理时，您需要负责其配置和维护。通过[启用安全特征](#)，避免将 Prometheus 抓取端点暴露给公共互联网。

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 服务器将从该集群收集指标。
- Amazon EKS 集群必须安装 [Amazon EBS CSI 驱动程序](#)（Helm 要求）。
- 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。*prometheus-namespace* 替换为您想要的这个命名空间的名称。

```
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。
 - **IAM_PROXY_PROMETHEUS_ROLE_ARN** 替换为您在中创建 amp-iamproxy-ingest-role 的 ARN。[设置服务角色从 Amazon EKS 集群中摄取指标](#)
 - **WORKSPACE_ID** 替换为适用于 Prometheus 的亚马逊托管服务工作空间的 ID。
 - **REGION** 替换为适用于 Prometheus 的亚马逊托管服务工作区的区域。

```
## 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。
- *prometheus-namespace* 替换为您的 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。](#)

• [步骤 2：使用 Helm 升级现有的 Prometheus 服务器](#)

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 头盔图名称。将 *prometheus-namespace* 替换为命名空间的名称。

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

当您向 Amazon Managed Service for Prometheus 发送数据时，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 [利用跨区域工作区在 Amazon Managed Service for Prometheus 中增加高可用性](#)。

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](#)
- [使用 Prometheus 将高可用性数据发送到 Amazon Managed Service for Prometheus](#)
- [使用 Prometheus Operator Helm 图表为 Amazon Managed Service for Prometheus 设置高可用性数据](#)
- [使用 Distro 向适用于 Prometheus 的亚马逊托管服务发送高可用性数据 AWS OpenTelemetry](#)
- [使用 Prometheus 社区 Helm 图表向 Amazon Managed Service for Prometheus 发送高可用性数据](#)
- [有关 Amazon Managed Service for Prometheus 中高可用性配置常见问题的解答](#)
- [利用跨区域工作区在 Amazon Managed Service for Prometheus 中增加高可用性](#)

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，这样您就可以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）可能会自行发布带有 `cluster` 标签的指标。这会导致 Amazon Managed Service for Prometheus 重复数据删除功能出现问题。有关更多信息，请参阅 [高可用性 FAQ](#)。

以下主题介绍如何发送数据以及如何包含 `cluster` 和 `__replica__` 标签，以便 Amazon Managed Service for Prometheus 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) 可能会自行发布带有 `cluster` 标签的指标。这会导致 Amazon Managed Service for Prometheus 重复数据删除功能出现问题。有关更多信息，请参阅 [高可用性 FAQ](#)。

使用 Prometheus Operator Helm 图表为 Amazon Managed Service for Prometheus 设置高可用性数据

要使用 Helm 中的 Prometheus Operator 设置高可用性配置，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 可以识别它们。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) 可能会自行发布带有 `cluster` 标签的指标。这会导致 Amazon Managed Service for Prometheus 重复数据删除功能出现问题。有关更多信息，请参阅 [高可用性 FAQ](#)。

使用 Distro 向适用于 Prometheus 的亚马逊托管服务发送高可用性数据 AWS OpenTelemetry

AWS Distro for OpenTelemetry (ADOT) 是该项目的安全且可用于生产的发行版。OpenTelemetry ADOT 为您提供源代码 APIs、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 Prometheus 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) 可能会自行发布带有 `cluster` 标签的指标。这会导致 Amazon Managed Service for Prometheus 重复数据删除功能出现问题。有关更多信息，请参阅[高可用性 FAQ](#)。

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) 可能会自行发布带有 `cluster` 标签的指标。这会导致 Amazon Managed Service for Prometheus 重复数据删除功能出现问题。有关更多信息，请参阅[高可用性 FAQ](#)。

有关 Amazon Managed Service for Prometheus 中高可用性配置常见问题的解答

我是否应该将值 `__replica__` 包含在另一个标签中以跟踪采样点？

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 `cluster` 标签，而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 中引入了一个带有 `cluster` 标签的新指标 `apiserver_storage_size_bytes`。这会导致 Amazon Managed Service for Prometheus 中的重复数据删除功能出现问题，这取决于 `cluster` 标签。在 Kubernetes 1.3 中，该标签重命名为 `storage-cluster_id`（在 1.28 和 1.29 的后续补丁中也进行了重命名）。如果您的集群发出带有 `cluster` 标签的指标，则 Amazon Managed Service for Prometheus 无法对关联的时间序列进行重复数据删除。我们建议您将 Kubernetes 集群升级到最新的补丁版本，以避免出现此问题。或者，您也可以将 `apiserver_storage_size_bytes` 指标上重新标记 `cluster` 标签，然后再将其摄取到 Amazon Managed Service for Prometheus。

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

利用跨区域工作区在 Amazon Managed Service for Prometheus 中增加高可用性

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。

要创建可视化表示指标的控制面板，可以使用 Amazon Managed Grafana 等服务。Amazon Managed Grafana（或 Grafana 的独立实例）可以构建一个图形界面，以多种显示演示样式显示您的指标。有关 Amazon Managed Grafana 的更多信息，请参阅 [《Amazon Managed Grafana 用户指南》](#)。

您还可以创建一次性查询，探索您的数据，或使用直接查询编写自己的应用程序来使用您的指标。直接查询使用 Amazon Managed Service for Prometheus API 和标准 Prometheus 查询语言 PromQL 从 Prometheus 工作区获取数据。有关 PromQL 及其语法的更多信息，请参阅 Prometheus 文档中的 [Querying Prometheus](#)。

主题

- [PromQL 备忘单](#)
- [基本选择器](#)
- [范围向量选择器](#)
- [聚合运算符](#)
- [常见的函数](#)
- [二元运算符](#)
- [实用查询示例](#)
- [确保指标查询安全](#)
- [设置 Amazon Managed Grafana 以与 Amazon Managed Service for Prometheus 配合使用](#)
- [设置 Grafana 开源或 Grafana Enterprise 以与 Amazon Managed Service for Prometheus 配合使用](#)
- [使用在 Amazon EKS 集群中运行的 Grafana 进行查询](#)
- [使用 Prometheus-compatible API 进行查询](#)
- [获取每次查询的查询使用统计数据](#)

PromQL 备忘单

在 Amazon Managed Service for Prometheus 工作区中查询指标时，使用此 PromQL（Prometheus 查询语言）备忘单作为快速参考。借助 PromQL，您可以通过其功能查询语言实时选择和聚合时间序列数据。

有关 PromQL 的更多详细信息，请参阅网站上的 [PromQL 备忘单](#)。PromLabs

基本选择器

按指标名称和标签匹配程序选择时间序列：

```
# Select all time series with the metric name http_requests_total
http_requests_total

# Select time series with specific label values
http_requests_total{job="prometheus", method="GET"}

# Use label matchers
http_requests_total{status_code!="200"}           # Not equal
http_requests_total{status_code=~"2.."}         # Regex match
http_requests_total{status_code!~"4.."}         # Negative regex match
```

范围向量选择器

选择随时间推移而变化的样本范围：

```
# Select 5 minutes of data
http_requests_total[5m]

# Time units: s (seconds), m (minutes), h (hours), d (days), w (weeks), y (years)
cpu_usage[1h]
memory_usage[30s]
```

聚合运算符

跨多个时间序列聚合数据：

```
# Sum all values
sum(http_requests_total)

# Sum by specific labels
sum by (job) (http_requests_total)
sum without (instance) (http_requests_total)
```

```
# Other aggregation operators
avg(cpu_usage)           # Average
min(response_time)      # Minimum
max(response_time)      # Maximum
count(up)                # Count of series
stddev(cpu_usage)       # Standard deviation
```

常见的函数

应用函数来转换数据：

```
# Rate of increase per second (for counters)
rate(http_requests_total[5m])

# Increase over time range
increase(http_requests_total[1h])

# Derivative (for gauges)
deriv(cpu_temperature[5m])

# Mathematical functions
abs(cpu_usage - 50)      # Absolute value
round(cpu_usage, 0.1)   # Round to nearest 0.1
sqrt(memory_usage)      # Square root

# Time functions
time()                  # Current Unix timestamp
hour()                  # Hour of day (0-23)
day_of_week()           # Day of week (0-6, Sunday=0)
```

二元运算符

执行算术和逻辑运算：

```
# Arithmetic operators
cpu_usage + 10
```

```
memory_total - memory_available
disk_usage / disk_total * 100

# Comparison operators (return 0 or 1)
cpu_usage > 80
memory_usage < 1000
response_time >= 0.5

# Logical operators
(cpu_usage > 80) and (memory_usage > 1000)
(status_code == 200) or (status_code == 201)
```

实用查询示例

您可以在 Amazon Managed Service for Prometheus 工作区中使用的常见监控查询：

```
# CPU usage percentage
100 - (avg by (instance) (rate(node_cpu_seconds_total{mode="idle"}[5m]))) * 100

# Memory usage percentage
(1 - (node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes)) * 100

# Request rate per second
sum(rate(http_requests_total[5m])) by (job)

# Error rate percentage
sum(rate(http_requests_total{status_code=~"5.."}[5m])) /
sum(rate(http_requests_total[5m])) * 100

# 95th percentile response time
histogram_quantile(0.95, sum(rate(http_request_duration_seconds_bucket[5m])) by (le))

# Top 5 instances by CPU usage
topk(5, avg by (instance) (cpu_usage))
```

确保指标查询安全

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

使用 AWS PrivateLink 使用适用于 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（具有相应权限）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

您可以使用 Grafana 实例在 Amazon Managed Service for Prometheus 中查询指标。本主题将向您介绍如何使用 Grafana 的独立实例从 Amazon Managed Service for Prometheus 查询指标。

先决条件

Grafana 实例 – 您必须拥有一个能够向 Amazon Managed Service for Prometheus 进行身份验证的 Grafana 实例。

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。

要检查你的 Grafana 版本，请输入以下命令，*grafana_install_directory* 替换为你的 Grafana 安装路径：

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

如果您还没有独立的 Grafana，或者需要更新的版本，可以安装一个新的实例。有关设置独立 Grafana 的说明，请参阅 Grafana 文档中的[安装 Grafana](#)。有关 Grafana 入门的信息，请参阅 Grafana 文档中的[Grafana 入门](#)。

AWS 账户 – 您必须拥有具有正确权限的 AWS 账户，才能访问 Amazon Managed Service for Prometheus 指标。

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](#)。

下一节将详细介绍如何通过 Grafana 设置身份验证。

步骤 1：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
```

步骤 2：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 应类似于 `https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`。

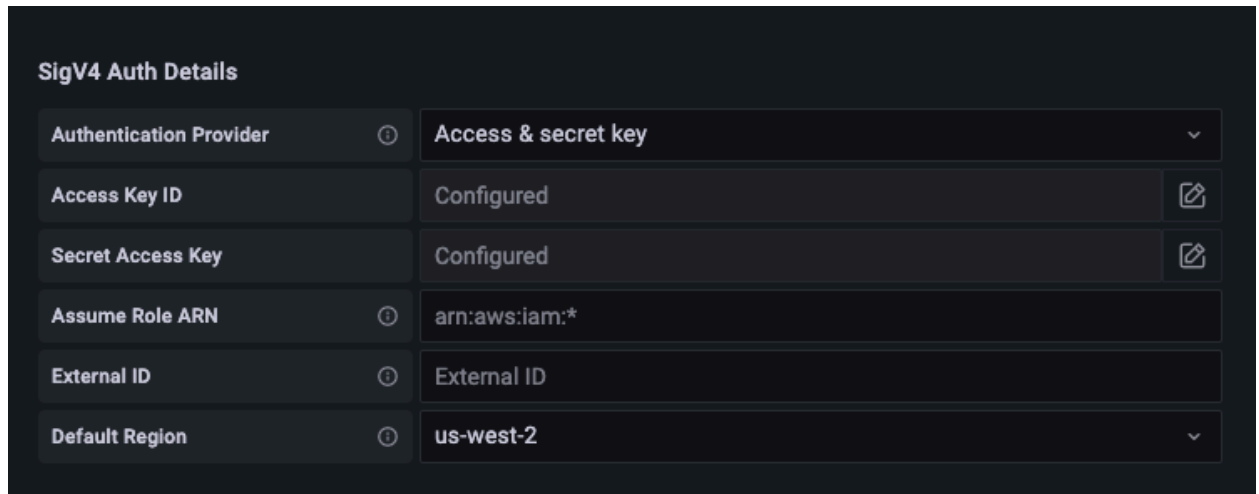
7. 在身份验证下，选择 Sigv4 身份验证的开关将其启用。
8. 您可以通过直接在 Grafana 中指定您的长期凭证来配置 Sigv4 授权，也可以使用默认的提供商链。直接指定您的长期凭证可以让您更快地启动，以下步骤首先给出了这些说明。在您更加熟悉将 Grafana 与 Amazon Managed Service for Prometheus 一起使用后，我们建议您使用默认的提供商链，因为它提供了更好的灵活性和安全性。有关设置默认提供商链的更多信息，请参阅[指定凭证](#)。

- 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. 对于秘密访问密钥，输入您的 AWS 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

如果您没有看到该消息，下一节将提供连接时的故障排除提示。

以下屏幕截图显示了 SDK 默认 SigV4 身份认证详细信息设置。

SigV4 Auth Details	
Authentication Provider	AWS SDK Default
Assume Role ARN	arn:aws:iam:*
External ID	External ID
Default Region	us-west-2

9. 针对新的数据来源测试 PromQL 查询：

- 选择探索。
- 运行示例 PromQL 查询，例如：

```
prometheus_tsdb_head_series
```

步骤 3：（可选）如果“保存&测试”不起作用进行故障排除

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 Amazon Prometheus Query Access 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 AWS SigV4

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。`account-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。当您这样做是为了将 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：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
      }
    }
  ],
```

```

    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": [
          "system:serviceaccount:aws-amp:amp-iamproxy-query-service-
account",
          "system:serviceaccount:grafana-namespace:grafana-service-account-
name"
        ],
        "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
      }
    }
  }
]
}

```

5. 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 Prometheus-compatible API 进行查询

尽管使用诸如[亚马逊托管 Grafana](#)之类的工具是查看和查询指标的最简单方法，但适用于 Prometheus 的亚马逊托管服务也支持 Prometheus-compatible 多个 API，您可以使用这些 API 来查询您的指标。有关所有可用 Prometheus-compatible API 的更多信息，请参阅[兼容普罗米修斯 APIs](#)。

这些 Prometheus-compatible 些 API 使用 Prometheus 查询语言 PromQL 来指定您要返回的数据。有关 PromQL 及其语法的详细信息，请参阅 Prometheus 文档中的[查询 Prometheus](#)。

使用这些 API 查询指标时，必须使用 AWS 签名版本 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。awscli 以下主题[使用 awscli 查询 Prometheus-compatible API](#) 将引导您完成使用设置 Sigv4 awscli 的过程。AWS

主题

- [使用 awscurl 通过 API 进行查询 Prometheus-compatible](#)

使用 awscurl 通过 API 进行查询 Prometheus-compatible

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
```

下面的示例是一个 `awscurl` 查询示例。用适合您的用例的值替换 *Region*、*Workspace-id* 和 *QUERY* 输入：

```
# 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 inferred 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

查询字符串必须使用 url 编码。

对于类似 `query=up` 的查询，您可能会得到如下结果：

```
{
  "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。您可以在中找到该角色的访问密钥和密钥<https://console.aws.amazon.com/iam/>。

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
```

```
$ awscli -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 awscli 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 awscli 应用程序及其依赖项。以下示例使用 Docker 运行时部署 awscli，但任何符合 OCI 的运行时和映像都可正常工作。

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

获取每次查询的查询使用统计数据

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。您可以获得每次查询的统计数据，以跟踪已处理的样本。通过在请求中包括查询参数 stats=all，query 或 queryRange API 的查询响应可包括已处理查询样本的统计数据。在 stats 对象中创建 samples 对象，并在响应中返回 stats 数据。

samples 对象包含以下属性：

属性	说明
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
```

异常检测

Amazon Managed Service for Prometheus 提供异常检测功能，该功能使用机器学习算法自动识别指标数据中的异常模式。此功能有助于您主动检测潜在问题，减少警报疲劳，并通过关注真正的异常行为而不是静态阈值来提高监控有效性。

Amazon Managed Service for Prometheus 中的异常检测使用随机森林砍伐 (RCF) 算法，该算法分析时间序列数据以建立正常行为模式，并识别与这些模式的偏差。该算法可适应季节性趋势，从容地处理缺失的数据，并为检测到的异常提供置信度分数。

异常检测的工作原理

Amazon Managed Service for Prometheus 异常检测使用机器学习来识别指标数据中的异常模式，而无需手动配置阈值。该系统学习正常行为模式和季节性变化，从而减少误报并实现早期问题检测。它不断地适应应用程序变化，使其适用于动态云环境。

异常检测可监控应用程序性能指标（例如响应时间和错误率），通过 CPU 和内存使用情况跟踪基础设施运行状况，检测异常用户行为，通过流量分析确定容量规划需求，并监控业务指标以了解意外变化。它最适合可预测的模式、季节性变化或渐进的增长趋势。

使用随机森林砍伐 (RCF) 算法来分析时间序列数据。RCF 创建决策树，用于对数据空间进行分区并识别远离正态分布的孤立点。该算法从传入的数据中学习，来为每个指标建立正常行为的动态模型。

启用后，它会分析历史数据以确定基准模式和季节性趋势，然后生成对预期值的预测并识别偏差。该算法产生四个关键输出：

- upper_band：预期正常值的上限
- lower_band：预期正常值的下限
- score：表示数据点异常程度的数值异常分数
- value：实际观测到的指标值

异常检测入门

要开始对 Prometheus 指标使用异常检测，您需要足够的历史数据让算法学习正常模式。我们建议在启用异常检测之前至少保持 14 天的一致指标数据，以获得最佳结果。

您可以使用 `PreviewAnomalyDetector` API 预览异常检测将如何与您的指标结合使用。使用 `PreviewAnomalyDetector` 根据历史数据测试算法并评估其有效性，然后再将其实施到生产监控中。有关更多信息，请参阅 [PreviewAnomalyDetector API](#)。

在实施异常检测时，请考虑以下最佳实践：

- 从稳定的指标开始：从具有一致模式的指标开始，最初避免高度不稳定或稀疏的数据。
- 使用聚合数据：将异常检测应用于聚合指标（例如平均值或总和），而不是原始的高基数数据，以提高性能和准确性。
- 调整灵敏度：根据您的特定使用案例和对误报与漏报异常的容忍度调整算法参数。
- 监控算法性能：定期检查检测到的异常，以确保算法随着系统的发展持续提供有价值的见解。

PreviewAnomalyDetector API

使用 `PreviewAnomalyDetector` 操作创建端点，用于演示异常检测算法在指定时间段内如何分析您的指标数据。此端点有助于您在实施之前评估和验证检测器的性能。

有效的 HTTP 动词

GET, POST

支持的有效载荷类型

URL 编码的参数

POST 的 `application/x-www-form-urlencoded`

支持的参数

`query=<string>` Prometheus 表达式查询字符串。

`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。

查询参数格式

在查询参数中用 `RandomCutForest (RCF)` 伪函数包装原始 PromQL 表达式。有关更多信息，请参阅适用于 Prometheus 的亚马逊托管服务 API 参考 [RandomCutForestConfiguration](#) 中。

RCF 函数使用此格式：

```
RCF(<query>
[,shingle size
[,sample size
[,ignore near expected from above
[,ignore near expected from below
[,ignore near expected from above ratio
[,ignore near expected from below ratio]]]])
```

除查询之外的所有参数都是可选的，省略时使用默认值。最小语法为：

```
RCF(<query>)
```

必须使用聚合函数包装查询。要使用特定的可选参数，同时忽略其他可选参数，请在函数中保留空位置：

```
RCF(<query>,,,,,1.0,1.0)
```

此示例仅根据预期值和观测值之间的比率来设置忽略异常检测峰值和下降的比率参数。

API 请求和响应

成功的调用返回的格式与 [QueryMetrics API](#) 相同。除了原始时间序列外，当有足够的样本可用时，API 还会返回以下新的时间序列：

- `anomaly_detector_preview:lower_band` : PromQL 表达式结果的预期值的下限
- `anomaly_detector_preview:score` : 异常分数介于 0 和 1 之间，其中 1 表示该数据点的异常可信度较高
- `anomaly_detector_preview:upper_band` : PromQL 表达式结果的预期值的上限

示例请求

```
POST /workspaces/workspace-id/anomalydetectors/preview
Content-Type: application/x-www-form-urlencoded
```

```
query=RCF%28avg%28vector%28time%28%29%29%29%2C%208%2C%20256%29&start=1735689600&end=1735695000&step=1m
```

示例响应

```
200 OK
...
{
  "status": "success",
  "data": {
    "result": [
      {
        "metric": {},
        "values": [
          [
            1735689600,
            "1735689600"
          ],
          [
            1735689660,
            "1735689660"
          ],
          .....
        ]
      },
      {
        "metric": {
          "anomaly_detector_preview": "upper_band"
        },
        "values": [
          [
            1735693500,
            "1.7356943E9"
          ],
          [
            1735693560,
            "1.7356945E9"
          ]
        ]
      }
    ]
  }
}
```

```
    .....
  ]
},
{
  "metric": {
    "anomaly_detector_preview": "lower_band"
  },
  "values": [
    [
      1735693500,
      "1.7356928E9"
    ],
    [
      1735693560,
      "1.7356929E9"
    ],
    .....
  ]
},
{
  "metric": {
    "anomaly_detector_preview": "score"
  },
  "values": [
    [
      1735693500,
      "0.0"
    ],
    [
      1735695000,
      "0.0"
    ],
    .....
  ]
}
],
"resultType": "matrix"
}
}
```

使用规则修改或监控收到的指标

您可以设置规则，以便在 Amazon Managed Service for Prometheus 收到指标时对指标采取操作。这些规则可以监控指标，甚至可以根据收到的指标创建新的计算指标。

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，警报管理器可配置为管理规则，或将其转发给通知下游接收器，如 Amazon Simple Notification Service。

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条规则文件。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他规则组或其他规则文件中的规则。

主题

- [了解使用规则所需的 IAM 权限](#)
- [创建规则文件](#)
- [将规则配置文件上传到 Amazon Managed Service for Prometheus](#)
- [编辑或替换规则配置文件](#)
- [对规则评估进行故障排除](#)
- [规则器故障排除](#)

了解使用规则所需的 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。

JSON

```
{
  "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。要使用此政策，请将 `<account>`、`<region><workspace-id>`、和 `<namespace-name>` 替换为 `<namespace-name>` 与您的账户对应的值。

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 规则文件是一个 YAML 文本文件，其格式与独立 Prometheus 中的规则文件相同。有关更多信息，请参阅 Prometheus 文档中的 [Defining Recording rules](#) 和 [Alerting rules](#)。

以下是规则文件的基本示例：

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

此示例创建了规则组 `cpu_metrics`，该组每 60 秒评估一次。此规则组使用记录规则创建一个名为 `avg_cpu_usage` 的新指标，然后在警报中使用该指标。下面描述一些使用的属性。有关警报规则和其他属性的更多信息，请参阅 Prometheus 文档中的 [警报规则](#)。

- `record: avg_cpu_usage` – 此记录规则创建一个名为 `avg_cpu_usage` 的新指标。
- 如果未指定 `interval` 属性，则规则组的默认评估间隔为 60 秒。
- `expr: avg(rate(node_cpu_seconds_total[5m])) by (instance)` – 记录规则的此表达式按 `instance` 标签分组，计算每个节点过去 5 分钟 CPU 的平均使用率。
- `alert: HighAverageCPU` – 此警报规则创建一个名为 `HighAverageCPU` 的新警报
- `expr: avg_cpu_usage > 0.8` – 此表达式要求警报查找 CPU 平均使用率超过 80% 的样本。
- `for: 10m`：只有当平均 CPU 使用率在至少 10 分钟内超过 80% 时，才会触发警报。

在这种情况下，该指标的计算值为过去 5 分钟的平均值。因此，只有当至少有两个连续的 5 分钟样本（总计 10 分钟）且其中平均 CPU 使用率高于 80% 时，才会触发警报。

- `keep_firing_for`: 20m – 此警报将继续触发，直到样本低于阈值至少 20 分钟。这对避免警报连续反复升降很有帮助。

Note

您可以在本地创建规则定义文件，然后将其上传到 Amazon Managed Service for Prometheus，也可以直接在 Amazon Managed Service for Prometheus 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

一旦您知道规则配置文件中需要哪些规则，就可以在控制台中创建和编辑该文件，也可以使用控制台或 AWS CLI 上传文件。

Note

如果您运行的是 Amazon EKS 集群，还可以使用 [AWS Controllers for Kubernetes](#) 上传规则配置文件。

使用 Amazon Managed Service for Prometheus 控制台编辑或替换规则配置并创建命名空间

1. 打开适用于 Prometheus 的亚马逊托管服务控制台，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，您也可以直接在 Amazon Managed Service for Prometheus 控制台中创建和编辑规则定义文件，方法是选择定义配置。这将创建一个默认定义文件样本，您可以在上传前对其进行编辑。

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。

要添加其他标签，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。

编辑或替换规则配置文件

如果要更改已经上传到 Amazon Managed Service for Prometheus 的规则文件中的规则，可以上传一个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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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）如果要下载当前的规则配置文件，请选择下载或复制。
6. 选择修改可直接在控制台内编辑配置。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。如果是这样选择，请选择新的规则定义文件，然后选择继续上传文件。

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。

对规则评估进行故障排除

本指南提供了有关亚马逊 Prometheus 托管服务 (AMP) 中规则评估常见问题的 step-by-step 故障排除程序。按照下列过程来诊断和解决警报和记录规则问题。

主题

- [验证警报触发状态](#)
- [解决警报通知缺失](#)
- [检查规则运行状况](#)
- [在查询中使用偏移量来处理摄取延迟](#)
- [常见问题和解决方案](#)
- [规则评估的最佳实践](#)

验证警报触发状态

在对规则评估问题进行故障排除时，首先要通过查询合成时间序列 ALERTS 来验证警报是否已触发。ALERTS 时间序列包括以下标签：

- 警报名称：警报的名称。
- 警报状态：待处理或触发。
 - 待处理：警报正在等待在 for 子句中指定的持续时间。
 - 触发：警报在指定的持续时间内满足条件。其他标签在您的警报规则中定义。

Note

当警报为触发或待处理时，样本值为 1。当警报处于空闲状态时，不会生成任何样本。

解决警报通知缺失

如果警报已触发但通知未到达，请验证以下 Alertmanager 设置：

1. 验证 Alertmanager 配置：检查路由、接收器和设置是否配置正确。查看路径屏蔽设置，包括等待时间、时间间隔和所需标签，这些设置可能会影响警报触发。将警报规则与其对应的路由和接收器进行比较，以确认匹配是否正确。对于带有 `time_interval` 的路由，请验证时间戳是否在指定的间隔内。
2. 检查警报接收器权限：使用 Amazon SNS 主题时，请验证 AMP 是否具有发布通知所需的权限。有关更多信息，请参阅 [授予 Amazon Managed Service for Prometheus 向 Amazon SNS 主题发送警报消息的权限](#)。
3. 验证接收器有效载荷兼容性：确认警报接收器接受 Alertmanager 的有效载荷格式。有关 Amazon SNS 要求，请参阅 [了解 Amazon SNS 消息验证规则](#)。
4. 查看 Alertmanager 日志：AMP 提供 Alertmanager 中的已出售日志，以协助调试通知问题。有关更多信息，请参阅 [使用日志监控亚马逊托管服务的 Prometheus 事件 CloudWatch](#)。

有关 Alertmanager 的更多信息，请参阅 [使用警报管理器在 Amazon Managed Service for Prometheus 中管理和转发警报](#)。

检查规则运行状况

格式错误的规则可能导致评估失败。使用以下方法来确定规则评估失败的原因：

Example

使用 ListRules API

[ListRules](#) API 提供有关规则运行状况的信息。检查 `health` 和 `lastError` 字段以诊断问题。

示例响应：

```
{
```

```
"status": "success",
"data": {
  "groups": [
    {
      "name": "my_rule_group",
      "file": "my_namespace",
      "rules": [
        {
          "state": "firing",
          "name": "broken_alerting_rule",
          "query": "...",
          "duration": 0,
          "keepFiringFor": 0,
          "labels": {},
          "annotations": {},
          "alerts": [],
          "health": "err",
          "lastError": "vector contains metrics with the same labelset after applying
alert labels",
          "type": "alerting",
          "lastEvaluation": "1970-01-01T00:00:00.000000000Z",
          "evaluationTime": 0.08
        }
      ]
    }
  ]
}
```

Example

使用已出售日志

ListRules API 仅显示最新信息。有关更详细的历史记录，请在工作区中启用 [已出售日志](#) 以访问：

- 评估失败的时间戳
- 详细错误消息
- 历史评估数据

已出售日志消息示例：

```
{
```

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

有关来自 Ruler 或 Alertmanager 的日志的更多示例，请参阅[规则器故障排除](#)和[使用警报管理器在 Amazon Managed Service for Prometheus 中管理和转发警报](#)。

在查询中使用偏移量来处理摄取延迟

默认情况下，表达式是在没有偏移量（即时查询）的情况下使用评估时的值进行评估的。如果指标摄取延迟，则记录规则所代表的值可能与您在摄取所有指标后手动评估表达式时的值不同。

Tip

使用偏移量修改器可以减少因摄取延迟而导致的问题。有关更多信息，请参阅《Prometheus 文档》中的[偏移量修改器](#)。

示例：处理延迟的指标

如果规则在 12:00 进行评估，但由于摄取延迟，该指标的最新样本来自 11:45，则该规则将找不到在 12:00 时间戳处的样本。要减轻这一影响，请添加偏移量，例如：**`my_metric_name offset 15m`**。

示例：处理来自多个来源的指标

当指标源自不同的来源（例如两台服务器）时，系统可能会在不同的时间摄取这些指标。为减轻这一影响，请构建一个表达式，例如：**`metric_from_server_A / metric_from_server_B`**

如果规则在服务器 A 和服务器 B 的摄取时间之间进行评估，您将得到预期之外的结果。使用偏移量有助于使评估时间保持一致。

常见问题和解决方案

记录规则数据存在差距

如果您发现记录规则数据与手动评估（当您通过查询 API 或 UI 直接执行记录规则的原始 PromQL 表达式时）相比存在差距，则可能是由于以下原因之一：

1. 评估时间长：一个规则组不能同时进行多个评估。如果评估时间超过配置的间隔，则可能会错过后续评估。超过所配置间隔的多次连续错过评估可能会导致记录规则变得过时。有关更多信息，请参阅《Prometheus 文档》中的 [Staleness](#)。您可以使用 CloudWatch 指标来监控评估持续时间 `RuleGroupLastEvaluationDuration`，以识别评估时间过长的规则组。
2. 监控错过的评估 — AMP 提供用于跟踪何时跳过评估的 `RuleGroupIterationsMissed` CloudWatch 指标。ListRules API 显示每个规则/组的评估时间和上次评估时间，这有助于识别错过评估的模式。有关更多信息，请参阅 [ListRules](#)。

建议：将规则拆分为不同的组

要缩短评估持续时间，请将规则拆分为不同的规则组。组内的规则按顺序执行，而规则组可以并行执行。将相互依赖的相关规则保留在同一个组中。通常，较小的规则组可确保评估更一致和差距更少。

规则评估的最佳实践

1. 优化规则组大小：使规则组保持较小，以确保评估一致性。将相关的规则分组在一起，但避免使用大的规则组。
2. 设置适当的评估间隔：在及时警报与系统负载之间取得平衡。查看受监控指标的稳定性模式，以了解其正常波动范围。
3. 对延迟的指标使用偏移量修改器：添加偏移量以补偿摄取延迟。根据观测到的摄取模式调整偏移量持续时间。
4. 监控评估性能：跟踪 `RuleGroupIterationsMissed` 指标。在 ListRules API 中查看评估时间。
5. 验证规则表达式：确保规则定义和手动查询之间的表达式完全匹配。测试具有不同时间范围的表达式以了解行为。
6. 定期查看规则运行状况：检查规则评估中是否存在错误。监控已出售日志，以了解反复出现的问题。

通过遵循这些故障排除步骤和最佳实践，您可以识别和解决 Amazon Managed Service for Prometheus 中规则评估的常见问题。

规则器故障排除

使用 [使用日志监控亚马逊托管服务的 Prometheus 事件 CloudWatch](#)，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_v1_namespace, namespace=canary_long_running_v1_namespace,
err=found duplicate series for the match group {dimension1=\\\\"1\\"} on the right
hand-side of the operation: [{__name__=\\\\"fake_metric2\\"}, {__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_v1_namespace",
    "namespace": "canary_long_running_v1_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。

使用警报管理器在 Amazon Managed Service for Prometheus 中管理和转发警报

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。警报管理器可对警报进行去重、分组并路由到下游接收器。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 - 如果某些其他警报已经触发，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他警报静音。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 支持模板化，使用单独的模板文件。模板可以使用条件语句和格式化数据等。

在 Amazon Managed Service for Prometheus 中，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

- [了解使用警报管理器所需的 IAM 权限](#)

- [在 Amazon Managed Service for Prometheus 中创建警报管理器配置以管理和路由警报](#)
- [使用 Amazon Managed Service for Prometheus 中的警报管理器将警报转发给警报接收器](#)
- [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](#)
- [使用 CloudWatch 日志对警报管理器进行故障排除](#)

了解使用警报管理器所需的 IAM 权限

必须向用户授予使用 Amazon Managed Service for Prometheus 中警报管理器的权限。创建具有以下权限的 AWS Identity and Access Management (IAM) 策略，并将此策略分配给您的用户、组或角色。

在 Amazon Managed Service for Prometheus 中创建警报管理器配置以管理和路由警报

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 Amazon Managed Service for Prometheus 控制台中编辑配置文件，但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他类型的接收方，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 }}
```

使用 Amazon Managed Service for Prometheus 中的警报管理器将警报转发给警报接收器

当警报规则发出警报时，会将其发送到警报管理器。警报管理器可对警报执行去重、在维护期间抑制警报或根据需要对警报进行分组等功能。然后，警报管理器会将警报作为消息转发给警报接收器。您可以设置一个警报接收器，该接收器可以通知操作人员、自动响应或以其他方式响应警报。

您可以在 Prometheus 的亚马逊托管服务中将亚马逊简单通知服务 (Amazon SNS) PagerDuty 配置为警报接收器。以下主题介绍如何创建和配置警报接收器。

主题

- [使用 Amazon SNS 作为警报接收器](#)
- [PagerDuty 用作警报接收器](#)

使用 Amazon SNS 作为警报接收器

您可以使用现有的 Amazon SNS 主题作为 Amazon Managed Service for Prometheus 的警报接收器，也可以创建一个新的主题。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。

无论使用新的还是现有的 Amazon SNS 主题，您都需要 Amazon SNS 主题的 Amazon 资源名称 (ARN) 来完成以下任务。

主题

- [授予 Amazon Managed Service for Prometheus 向 Amazon SNS 主题发送警报消息的权限](#)
- [配置警报管理器以向 Amazon SNS 主题发送消息](#)
- [将警报管理器配置为以 JSON 格式向 Amazon SNS 发送消息](#)
- [将 Amazon SNS 配置为向其他目的地发送警报消息](#)
- [了解 Amazon SNS 消息验证规则](#)

授予 Amazon Managed Service for Prometheus 向 Amazon SNS 主题发送警报消息的权限

您必须授予 Amazon Managed Service for Prometheus 向您的 Amazon SNS 主题发送消息的权限。以下策略声明将授予该权限。其中包括一项 Condition 声明，以防止出现被称为“混淆代理”问题的安全问题。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. [在 v3/home 上打开亚马逊 SNS 控制台。https://console.aws.amazon.com/sns/](https://console.aws.amazon.com/sns/)
2. 在导航窗格中，选择主题。
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 Amazon SNS 主题启用了服务端加密 (SSE)，则需要在用于加密主题的密 AWS KMS 策略中添加 `kms:GenerateDataKey*` 和 `kms:Decrypt` 权限，从而允许适用于 Prometheus 的亚马逊托管服务向该加密主题发送消息。

例如，您可以在策略中添加以下内容：

```

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

```

有关更多信息，请参阅 [SNS 主题的 AWS KMS 权限](#)。

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](#)。

选择加入区域的 SNS 主题配置

您可以使用配置 `aps.amazonaws.com` 与适用于 Prometheus 的亚马逊托管服务工作区 AWS 区域相同的亚马逊 SNS 主题。要将来自 non-opt-in 区域（例如 `us-east-1`）的 SNS 主题与可选区域（例如 `af-south-1`）一起使用，您需要使用区域服务主体格式。在区域服务原则中，`us-east-1` 替换为您要使用的 non-opt-in 区域：`aps.us-east-1.amazonaws.com`。

下表列出了选择加入区域及其相应的区域服务主体：

选择加入区域及其区域服务主体

区域名称	Region	区域服务主体
非洲（开普敦）	af-south-1	af-south-1.aps.amazonaws.com
亚太地区（香港）	ap-east-1	ap-east-1.aps.amazonaws.com
亚太地区（泰国）	ap-southeast-7	ap-southeast-7.aps.amazonaws.com
欧洲地区（米兰）	eu-south-1	eu-south-1.aps.amazonaws.com
欧洲（苏黎世）	eu-central-2	eu-central-2.aps.amazonaws.com

区域名称	Region	区域服务主体
中东 (阿联酋) :	me-central-1	me-central-1.aps.amazonaws.com
亚太地区 (马来西亚)	ap-southeast-5	ap-southeast-5.aps.amazonaws.com

有关启用选择加入区域的信息，请参阅 Amazon Web Services 一般参考中的《IAM 用户指南》中的[管理 AWS 区域](#)。

在为这些选择加入区域配置 Amazon SNS 主题时，请确保使用正确的区域服务主体来启用跨区域发送警报。

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，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况，AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

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，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 Amazon SNS 主题发送警报消息的权限](#) 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。

配置警报管理器以向 Amazon SNS 主题发送消息

在您拥有 (新的或现有的) 标准类型的 Amazon SNS 主题后，您可以将其作为警报接收器添加到警报管理器配置中。警报管理器可将警报转发给已配置的警报接收器。要完成此操作，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: AWS ##
        topic_arn: ARN_of_SNS_topic
        subject: yoursubject
        attributes:
          key: yourkey
          value: yourvalue
```

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 [将警报管理器配置为以 JSON 格式向 Amazon SNS 发送消息](#)。

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](#)。

将警报管理器配置为以 JSON 格式向 Amazon SNS 发送消息

默认情况下，Amazon Managed Service for Prometheus 警报管理器以纯文本列表格式输出消息。这可能会增加其他服务的解析难度。您可以将警报管理器配置为以 JSON 格式发送警报。JSON 可以更轻松地在接收网络挂钩的终端节点中 AWS Lambda 或接收网络挂钩的终端节点中处理来自 Amazon SNS 的下游消息。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 [在 Amazon Managed Service for Prometheus 中创建警报管理器配置以管理和路由警报](#)。
- 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](#)。

将 Amazon SNS 配置为向其他目的地发送警报消息

Amazon Managed Service for Prometheus 只能向 Amazon Simple Notification Service (Amazon SNS) 发送警报消息。要将这些消息发送到其他目的地，例如电子邮件、webhook、Slack 或 OpsGenie，您必须将 Amazon SNS 配置为将消息转发到这些终端节点。

以下各节将介绍如何配置 Amazon SNS 以将警报转发到其他目的地。

主题

- [电子邮件](#)
- [Webhook](#)
- [Slack](#)
- [OpsGenie](#)

电子邮件

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

了解 Amazon SNS 消息验证规则

Amazon Simple Notification Service (Amazon SNS) 要求消息符合特定标准。不符合这些标准的消息将在收到后予以修改。如有必要，Amazon SNS 接收器将根据以下规则验证、截断或修改警报消息：

- 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 Message: 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。
- 将添加一个带有“已修改”键的消息属性，并且“消息：错误-消息”的值已从 X KB 中截断，因为它超过 256 KB 的大小限制。
- 主题包含控制字符或非 ASCII 字符。
- 如果主题包含控制字符或非 ASCII 字符，则 SNS 会将主题替换为 Error - contains control- or non-ASCII characters。
- 对于 SNS 电子邮件主题，请移除控制字符，例如换行符：\n。
- 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。
- 将添加一个带有 modified 键的消息属性，并且“主题：错误-主题”的值已从 X 字符中截断，因为它超过了 100 个字符的大小限制。
- 消息属性的键/值无效。
- 无效的消息属性将被删除。
- 将添加一个带有 modified 键的消息属性，且由于无效 MessageAttributeKey 或，消息属性的值为 MessageAttribute:Error-X 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 modified 键的消息属性，并删除消息属性的值 MessageAttribute : Error-X，因为它超过了 256KB 的大小限制。

PagerDuty 用作警报接收器

您可以将 Prometheus 的亚马逊托管服务配置为直接向其发送提醒。PagerDuty 此集成要求您将 PagerDuty 集成密钥存储在中，AWS Secrets Manager 并授予适用于 Prometheus 的亚马逊托管服务读取密钥的权限。

PagerDuty 集成可实现事件响应工作流程的自动化，并确保关键警报在正确的时间到达正确的团队成员。当您 PagerDuty 用作警报接收器时，您可以利用 PagerDuty 的上报政策、待命调度和事件管理功能来确保警报得到快速确认和解决。这种集成对于生产环境特别有价值，在此类环境中，快速响应系统问题对于维护服务可用性和满足 SLA 要求至关重要。有关更多信息，请参阅 PagerDuty 网站上的 [PagerDuty 知识库](#)。

PagerDuty 配置选项

选项	描述	必填
routing_key	服务集成的 PagerDuty 路由密钥。您必须将其指定为 Secrets Manager ARN	是
service_key	PagerDuty 服务集成的服务密钥。您必须将其指定为 Secrets Manager ARN	是 (对于事件 API v1)
client	通知程序的客户端标识	否
client_url	指向通知发送者的反向链接	否
description	事件的描述	否
details	一组任意 key/value 配对，可提供有关事件的更多细节	否
severity	事件的严重性	否
class	事件的类别或类型	否
component	源计算机中负责事件的组件	否

选项	描述	必填
group	组件的逻辑分组	否
source	受影响系统的唯一位置	否

Note

不支持 `url`、`service_key_file`、`routing_key_file` 和 `http_config` 选项。

以下主题介绍如何在亚马逊 Prometheus PagerDuty 托管服务中配置为警报接收者。

主题

- [配置 AWS Secrets Manager 和权限](#)
- [配置警报管理器以向其发送警报 PagerDuty](#)

配置 AWS Secrets Manager 和权限

在向发送警报之前 PagerDuty，必须安全地存储您的 PagerDuty 集成密钥并配置必要的权限。此过程包括在中创建密钥 AWS Secrets Manager，使用客户托管 AWS Key Management Service (AWS KMS) 密钥对其进行加密，以及向适用于 Prometheus 的 Amazon 托管服务授予访问该密钥及其加密密钥所需的权限。以下过程将指导您完成此配置过程的每个步骤。

在 Secrets Manager 中为以下内容创建密钥 PagerDuty

要 PagerDuty 用作警报接收器，必须将 PagerDuty 集成密钥存储在 Secrets Manager 中。按照以下步骤进行操作：

1. 打开 [Secrets Manager 控制台](#)。
2. 选择存储新密钥。
3. 对于密钥类型，请选择其他密钥类型。
4. 对于密钥/值对，请输入您的 PagerDuty 集成密钥作为秘密值。这要么是您的 PagerDuty 集成中的路由密钥，要么是服务密钥。
5. 选择下一步。
6. 输入密钥的名称和描述，然后选择下一步。

7. 根据需要配置轮换设置，然后选择下一步。
8. 检查您的设置，然后选择存储。
9. 创建密钥后，记下它的 ARN。在配置警报管理器时，需要此信息。

使用客户管理 AWS KMS 的密钥加密您的密钥

您必须向 Amazon Managed Service for Prometheus 授予访问您的密钥及其加密密钥的权限：

1. 密钥资源策略：在 [Secrets Manager 控制台](#) 中打开您的密钥。
 - a. 选择资源权限。
 - b. 选择编辑权限。
 - c. 添加以下策略语句。在语句中，*highlighted values* 用您的特定值替换。

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "aps.amazonaws.com"
  },
  "Action": "secretsmanager:GetSecretValue",
  "Resource": "*",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:aps:aws-
region:123456789012:workspace/WORKSPACE_ID"
    },
    "StringEquals": {
      "aws:SourceAccount": "123456789012"
    }
  }
}
```

- d. 选择保存。
2. KMS 密钥策略：在 [AWS KMS 控制台](#) 中打开您的 AWS KMS 密钥。
 - a. 选择密钥策略。
 - b. 选择编辑。
 - c. 添加以下策略语句。在语句中，*highlighted values* 用您的特定值替换。

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "aps.amazonaws.com"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:aps:aws-region:123456789012:workspace/WORKSPACE_ID"
    },
    "StringEquals": {
      "aws:SourceAccount": "123456789012"
    }
  }
}
```

d. 选择保存。

后续步骤：继续探讨下一个主题，即[配置警报管理器以向其发送警报 PagerDuty](#)。

配置警报管理器以向其发送警报 PagerDuty

要将警报管理器配置为向其发送警报 PagerDuty，您需要更新警报管理器定义。您可以使用 AWS 管理控制台 AWS CLI、或来执行此操作 AWS SDKs。

Example 警报管理器配置

以下是向发送警报的警报管理器配置示例 PagerDuty。在示例中，*highlighted values*用您的特定值替换。

```
alertmanager_config: |
  route:
    receiver: 'pagerduty-receiver'
    group_by: ['alertname']
    group_wait: 30s
    group_interval: 5m
    repeat_interval: 1h
  receivers:
    - name: 'pagerduty-receiver'
      pagerduty_configs:
```

```

- routing_key:
  aws_secrets_manager:
    secret_arn: 'arn:aws:secretsmanager:aws-
region:123456789012:secret:YOUR_SECRET_NAME'
    secret_key: 'YOUR_SECRET_KEY'
    refresh_interval: 5m
  description: '{{ .CommonLabels.alertname }}'
  severity: 'critical'
  details:
    firing: '{{ .Alerts.Firing | len }}'
    status: '{{ .Status }}'
    instance: '{{ .CommonLabels.instance }}'

```

Example AWS CLI

以下是用于更新警报管理器定义的 AWS CLI 命令。在示例中，*highlighted values* 用您的特定值替换。

```

aws amp put-alert-manager-definition \
  --workspace-id WORKSPACE_ID \
  --data file://alertmanager-config.yaml

```

故障排除 PagerDuty 集成

如果未向发送警报 PagerDuty，请检查以下项目：

- 确认您的密钥存在且包含正确的 PagerDuty 集成密钥。
- 确认您的密钥已使用客户自主管理型 KMS 密钥进行加密。
- 确保密钥和 KMS 密钥的资源策略向 Amazon Managed Service for Prometheus 授予必要的权限。
- 检查警报管理器配置中的 ARN 是否正确地引用您的密钥。
- 确认您的 PagerDuty 集成密钥在您的 PagerDuty 账户中有效且有效。

适用于 Prometheus 的亚马逊托管服务支持 CloudWatch 亚马逊日志和以下指标，CloudWatch 以帮助进行故障排除。有关更多信息，请参阅[使用日志监控亚马逊托管服务的 Prometheus 事件 CloudWatch](#)和[使用 CloudWatch 指标监控亚马逊托管服务的 Prometheus 资源](#)。

- SecretFetchFailure
- AlertManagerNotificationsThrottledByIntegration

- AlertManagerNotificationsFailedByIntegration

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

一旦您知道自己想要在警报管理器配置文件中添加什么内容，就可以在控制台中创建和编辑该文件，也可以使用 Amazon Managed Service for Prometheus 控制台或 AWS CLI 上传现有文件。

Note

如果运行的是 Amazon EKS 集群，还可以使用 [AWS Controllers for Kubernetes](#) 上传警报管理器配置文件。

使用 Amazon Managed Service for Prometheus 控制台编辑或替换警报管理器配置

1. 打开适用于 Prometheus 的亚马逊托管服务控制台，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，请改选修改。

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，您也可以选择创建定义选项，创建一个新文件并直接在控制台中进行编辑。这将创建一个默认配置示例，您可以在上传前对其进行编辑。

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 file://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 证书，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。

有关创建 AWS 账户 和 IAM 证书的更多信息，请参阅[设置 AWS](#)。

- 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 Amazon Managed Service for Prometheus 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[使用警报管理器](#)

在 [Amazon Managed Service for Prometheus 中管理和转发警报](#)。有关规则的更多信息，请参阅 [使用规则修改或监控收到的指标](#)。

- 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[步骤 2：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。在以下示例中，*variables* 用您自己的（特定于账户的）信息替换：

```
https://aps-workspaces.US East (N. Virginia).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。

使用 CloudWatch 日志对警报管理器进行故障排除

使用 [使用日志监控亚马逊托管服务的 Prometheus 事件 CloudWatch](#)，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

- [活动警报警告](#)
- [警报聚合组大小警告](#)
- [警报大小过大警告](#)
- [空内容警告](#)
- [key/value 警告无效](#)

- [消息限制警告](#)
- [没有基于资源的策略错误](#)
- [非 ASCII 警告](#)
- [未授权调用 KMS](#)
- [模板错误](#)

活动警报警告

当日志包含以下警告时

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

这表示已超出警报管理器活动警报配额。

要采取的操作

请求提高限额。登录 AWS 管理控制台 并打开 Service Quotas 控制台，网址为<https://console.aws.amazon.com/servicequotas/>。

警报聚合组大小警告

当日志包含以下警告时

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

这表示已超出警报管理器警报聚合组大小配额。

要采取的操作

可以使用 `group_by` 参数减小警报聚合组大小。有关更多信息，请参阅《Prometheus 文档》中的[路由相关设置](#)。

您也可以请求提高限额。登录 AWS 管理控制台 并打开 Service Quotas 控制台，网址为<https://console.aws.amazon.com/servicequotas/>。

警报大小过大警告

当日志包含以下警告时

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

这意味着已超出警报管理器每个工作区的警报大小配额。

要采取的操作

移除不必要的注释和标签以缩小警报大小。

空内容警告

当日志包含以下警告时

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

key/value 警告无效

当日志包含以下警告时

```
{
  "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
  "message": {
    "log": "MessageAttributes has been removed because of invalid key/value,
    numberOfRemovedAttributes=1"
    "level": "WARN"
  },
  "component": "alertmanager"
}
```

这意味着某些消息属性因无效而 keys/values 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-abcd1234-ef56-78ab-cd90-1234abcd0000",
  "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-abcd1234-ef56-78ab-cd90-1234abcd0000",
  "message": {
    "log": "Notify for alerts failed, AMP is not authorized to perform: SNS:Publish
on resource: arn:aws:sns:us-west-2:12345:testSnsReceiver because no resource-based
policy allows the SNS:Publish action"
    "level": "ERROR"
  },
  "component": "alertmanager"
}
```

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` (Amazon Managed Service for Prometheus) 访问 Amazon SNS 主题的权限。有关 SNS 访问策略的更多信息，请参阅《Amazon Simple Notification Service 开发人员指南》中的[使用访问策略语言](#)和 [Amazon SNS 访问控制示例案例](#)。

非 ASCII 警告

当日志包含以下警告时

```
{
  "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
  "message": {
    "log": "Subject has been modified because it contains control or non-ASCII
characters."
    "level": "WARN"
  },
  "component": "alertmanager"
}
```

```
}
```

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。

未授权调用 KMS

当日志包含以下 AWS KMS 错误时

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

要采取的操作

验证用于加密 Amazon SNS 主题的密钥的密钥政策是否支持 Amazon Managed Service for Prometheus 服务主体 `aps.amazonaws.com` 执行以下操作：`kms:GenerateDataKey*` 和 `kms:Decrypt`。有关更多信息，请参阅 [SNS 主题的 AWS KMS 权限](#)。

模板错误

当日志包含以下错误时

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

```
}
```

这意味着 AlertManager 定义中使用的模板存在错误。错误条目包含有关接收器、sns_configs 中的位置以及包含错误的属性的说明。

要采取的操作

验证警报管理器定义。确保语法正确，并且您引用的模板变量和函数存在。有关更多信息，请参阅 Prometheus 开源文档中的[通知模板参考](#)。

记录和监控 Amazon Managed Service for Prometheus 工作区

亚马逊 Prometheus 托管服务使用 CloudWatch 亚马逊来提供有关其运营的数据。您可以使用 CloudWatch 指标来了解资源使用情况以及对适用于 Prometheus 的亚马逊托管服务工作空间的请求。您可以开启 CloudWatch 日志支持以获取工作空间中发生的事件的日志。

以下主题更详细地描述 CloudWatch 了使用。

使用 CloudWatch 指标监控亚马逊托管服务的 Prometheus 资源

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。有关使用率指标的更多信息，请参阅 [CloudWatch 使用率指标](#)。

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ResourceCount*	CreateAlertManagerAlertsTPS	AWS/Usage	每个工作区、每秒可执行的 CreateAlertManagerAlerts API 操作的最大数量。
ResourceCount*	DeleteAlertManagerSilencesTPS	AWS/Usage	每个工作区、每秒可执行的 DeleteAlertManagerSilences API 操作的最大数量。
ResourceCount*	GetAlertManagerSilenceTPS	AWS/Usage	每个工作区、每秒可执行的 GetAlertManagerSilence API 操作的最大数量。
ResourceCount*	GetAlertManagerStatusTPS	AWS/Usage	每个工作区、每秒可执行的 GetAlertManagerSta

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ResourceCount*	GetLabelsTPS	AWS/Usage	tus API 操作的最大数量。 每个工作区、每秒可执行的 GetLabels API 操作的最大数量。
ResourceCount*	GetMetricMetadataTPS	AWS/Usage	每个工作区、每秒可执行的 GetMetricMetadata API 操作的最大数量。
ResourceCount*	GetSeriesTPS	AWS/Usage	每个工作区、每秒可执行的 GetSeries API 操作的最大数量。
ResourceCount	InhibitionRulesInAlertManagerDefinition	AWS/Usage	警报管理器定义文件中最大的抑制规则数。
ResourceCount*	ListAlertManagerAlertGroupInfosTPS	AWS/Usage	每个工作区、每秒可执行的 ListAlertManagerAlertGroupInfos API 操作的最大数量。
ResourceCount*	ListAlertManagerAlertGroupsTPS	AWS/Usage	每个工作区、每秒可执行的 ListAlertManagerAlertGroups API 操作的最大数量。
ResourceCount*	ListAlertManagerAlertsTPS	AWS/Usage	每个工作区、每秒可执行的 ListAlertManagerAlerts API 操作的最大数量。

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ResourceCount*	ListAlertManagerReceiversTPS	AWS/Usage	每个工作区、每秒可执行的 ListAlertManagerReceivers API 操作的最大数量。
ResourceCount*	ListAlertManagerSilencesTPS	AWS/Usage	每个工作区、每秒可执行的 ListAlertManagerSilences API 操作的最大数量。
ResourceCount*	ListAlertsTPS	AWS/Usage	每个工作区、每秒可执行的 ListAlerts API 操作的最大数量。
ResourceCount*	ListRulesTPS	AWS/Usage	每个工作区、每秒可执行的 ListRules API 操作的最大数量。
ResourceCount*	PutAlertManagerSilencesTPS	AWS/Usage	每个工作区、每秒可执行的 PutAlertManagerSilences API 操作的最大数量。
ResourceCount	HAReplicaGroupCount	AWS/Usage	高可用性副本组的数量
ResourceCount*	QueryMetricsTPS	AWS/Usage	每秒查询操作数
ResourceCount*	RemoteWriteTPS	AWS/Usage	每秒远程写入操作数

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ResourceCount	ActiveAlerts	AWS/Usage	每个工作区的活动警报数 单位：计数 有效统计数据：平均值、最小值、最大值
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：平均值、最小值、最大值
ResourceCount	AlertAggregationGroupSize	AWS/Usage	警报管理器定义文件中的警报聚合组的最大大小。group_by 的每个标签值组合都会创建一个聚合组。
ResourceCount	AlertManagerDefinitionSizeBytes	AWS/Usage	警报管理器定义文件的最大大小（以字节为单位）。
ResourceCount	AllSilences	AWS/Usage	每个工作区的最大静默数，包括已过期、活动和待处理的静默。
ResourceCount	IngestionRate	AWS/Usage	样本摄取率 单位：每秒计数 有效统计数据：平均值、最小值、最大值
ResourceCount	RuleEvaluationInterval	AWS/Usage	最小规则评估间隔

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ResourceCount	RuleGroup Namespace Definition nSizeBytes	AWS/Usage	一个规则组命名空间定义文件的最大大小（以字节为单位）。
ResourceCount	Templates InAlertMa nagerDefinition	AWS/Usage	警报管理器定义文件中的最大模板数。
ResourceCount	WorkspaceCount	AWS/Usage	每个区域、每个账户的最大工作区数量。
ResourceCount	SizeOfAlerts	AWS/Usage	工作区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：平均值、最小值、最大值
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：平均值、最小值、最大值

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ResourceCount	UnprocessedAlerts	AWS/Usage	<p>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。</p> <p>单位：计数</p> <p>有效统计数据：平均值、最小值、最大值</p>
ResourceCount	AllAlerts	AWS/Usage	<p>每个工作区处于任何状态的警报数量</p> <p>单位：计数</p> <p>有效统计数据：平均值、最小值、最大值</p>
ResourceCount	AllRules	AWS/Usage	<p>每个工作区处于任何状态的规则数量</p> <p>单位：计数</p> <p>有效统计数据：平均值、最小值、最大值</p>
ActiveSeriesPerLabelSet	-	AWS/Prometheus	<p>每个用户定义的标签集的当前活动系列使用情况</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
ActiveSeriesLimitPerLabelSet	-	AWS/Prometheus	<p>每个用户定义的标签集的当前活动系列限制值</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>
AlertManagerAlertsReceived	-	AWS/Prometheus	<p>警报管理器收到的成功警报总数</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>
AlertManagerNotificationsFailed	-	AWS/Prometheus	<p>发送失败的警报数量</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>
AlertManagerNotificationsThrottled	-	AWS/Prometheus	<p>限制的警报数量</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
AnomalyDetectors	WorkspaceId	AWS/Prometheus	给定工作区的异常检测器总数 单位：计数 有效统计数据：平均值、最小值、最大值
AnomalyDetectorEvaluations	WorkspaceId, AnomalyDetectorId	AWS/Prometheus	异常检测器评估总数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
AnomalyDetectorEvaluationFailures	WorkspaceId, AnomalyDetectorId	AWS/Prometheus	间隔内异常检测器失败的次数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
AnomalyDetectorLastEvaluationDuration	WorkspaceId, AnomalyDetectorId	AWS/Prometheus	异常检测器上次评估的持续时间 单位：秒 有效统计数据：Average、Minimum、Maximum、Sum

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
AnomalyDetectorMissedEvaluations	WorkspaceId, AnomalyDetectorId	AWS/Prometheus	<p>间隔内错过的异常检测器评估次数</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>
DiscardedSamples ^{**}	-	AWS/Prometheus	<p>按原因划分的丢弃样本数量</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>
DiscardedSeries ^{**}	-	AWS/Prometheus	<p>按原因包含丢弃样本的序列数</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>
DiscardedSamplesPerLabelSet	-	AWS/Prometheus	<p>每个用户定义的标签集的丢弃样本计数</p> <p>单位：计数</p> <p>有效统计数据：Average、Minimum、Maximum、Sum</p>

CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
Discarded SeriesPer LabelSet	-	AWS/Prometheus	包含每个用户定义标签集的已丢弃样本的系列计数 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
Ingestion RatePerLabelSet	-	AWS/Prometheus	每个用户定义的标签集的摄取率 单位：计数 有效统计数据：Average、Minimum、Maximum、Sum
QuerySamplesProcessed	-	AWS/Prometheus	处理的查询样本数 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


CloudWatch 指标名称	资源名称	CloudWatch 命名空间	说明
RuleEvaluationFailures	-	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
RuleGroupIterationsMissed	-	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
RuleGroupLastEvaluationDuration	-	AWS/Prometheus	规则组的上次评估的持续时间。 单位：秒 有效统计数据：Average、Minimum、Maximum、Sum

*TPS 指标每分钟生成一次，并且是该分钟内的每秒平均值。TPS 指标中不会捕捉到短暂的爆发期间。


**导致样本被丢弃的一些原因如下。并非以下所有原因都出现在 DiscardedSeries 指标中。

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上一个样本的时间戳相同，但值不同。

Reason	含义
per_labelset_series_limit	用户已达到每标签集的活跃系列总数上限。
per_metric_series_limit	用户已达到每个指标活跃系列数上限。
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	标签值超过支持的字符限制。
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。

 Note

指标不存在或缺失等同于该指标的值为 0。

 Note

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

RuleGroupNamespace;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

要在数量上设置警报 ActiveSeries 在 Prometheus

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。

使用日志监控亚马逊托管服务的 Prometheus 事件 CloudWatch

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。

JSON

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

配置 CloudWatch 日志

您可以使用控制台或 Prometheus 在 Amazon 托管服务中配置登录。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

您可以使用来设置日志配置 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](#)。

在 Amazon Managed Service for Prometheus 中管理查询成本

Amazon Managed Service for Prometheus 通过限制单个查询可以使用的已处理的查询样本 (QSP) 数量，来限制查询成本。您可以为 QSP 配置两种类型的阈值，即警告和错误，以协助有效地管理和控制查询成本。

当查询达到警告阈值时，API 查询响应中会显示一条警告消息。对于通过 Amazon Managed Grafana 查看的查询，该警告将在 Amazon Managed Grafana UI 中显示，有助于用户识别代价高昂的查询。达到错误阈值的查询不需要付费，并且会因错误而被拒绝。

除了查询限制外，适用于 Prometheus 的 Amazon 托管服务还提供将查询性能数据记录到日志的功能。CloudWatch 此功能支持您详细分析查询，有助于您优化 Amazon Managed Service for Prometheus 查询并更有效地管理成本。查询日志记录捕获有关超出指定的已处理查询样本（QSP）数阈值的查询的信息。然后，这些数据将发布到 CloudWatch 日志，使您能够调查和分析查询性能。记录的查询包括 API 查询和规则查询。默认情况下，查询日志处于禁用状态，以最大限度地减少不必要的 CloudWatch 日志使用量。当查询分析需要时，您可以启用此功能。

主题

- [配置查询日志记录](#)
- [配置查询节流阈值](#)
- [日志内容](#)
- [限制](#)

配置查询日志记录

您可以在适用于 Prometheus 的亚马逊托管服务控制台中配置查询日志，也可以通过调用 API 请求在 AWS CLI 中配置查询日志。create-query-logging-configuration 此 API 正文包含目的地列表，但目前，我们仅支持 CloudWatch Logs 作为目的地，目标应该只包含一个带有 CloudWatch 配置的元素。

先决条件

确保已创建了 logGroup。用于进行配置的 ID 或角色应具有以下策略或等效的权限。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:GetLogDelivery",
        "logs:UpdateLogDelivery",
        "logs>DeleteLogDelivery",
        "logs>ListLogDeliveries",
        "logs:PutResourcePolicy",

```

```

        "logs:DescribeResourcePolicies",
        "logs:DescribeLogGroups",
        "aps:CreateQueryLoggingConfiguration",
        "aps:UpdateQueryLoggingConfiguration",
        "aps:DescribeQueryLoggingConfiguration",
        "aps:DeleteQueryLoggingConfiguration"
    ],
    "Resource": "*"
}
]
}

```

配置 CloudWatch 日志

您可以使用或登录适用于 Prometheus 的亚马逊 Prometheus 托管服务来配置 CloudWatch 日志。
AWS 管理控制台 AWS CLI

使用 Amazon Managed Service for Prometheus 控制台配置查询日志记录

1. 导航到工作区详细信息面板中的日志选项卡。
2. 在查询洞察下，选择创建。
3. 选择日志组下拉列表并选择要发布日志的日志组。

您也可以在 CloudWatch 控制台中创建新的日志组。

4. 输入阈值 (QSP)。
5. 选择保存。

要使用命令配置查询日志 AWS CLI

```

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

有关如何更新、删除和描述操作的信息，请参阅 [Amazon Managed Service for Prometheus API 参考](#)。

配置查询节流阈值

要配置 QSP 阈值，必须在 [QueryMetrics API](#) 中提供查询参数。

- `max_samples_processed_warning_threshold`：为已处理的查询样本设置警告阈值
- `max_samples_processed_error_threshold`：为已处理的查询样本设置错误阈值

对于 Amazon Managed Grafana 用户，您可以使用 grafana 数据来源配置对来自该数据来源的所有查询应用限制：

1. 在 Amazon Managed Grafana 中浏览到 Amazon Managed Service for Prometheus 数据来源配置。
2. 在自定义查询参数下，添加阈值标头。
3. 选择保存。

日志内容

对于源自规则的查询，您将在 CloudWatch 日志中看到有关该查询的以下信息：

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

对于源自 API 调用的查询，您将在 CloudWatch 日志中看到有关该查询的以下信息：

```
{
  workspaceId: "ws-5e7658c2-7ccf-4c30-9de9-2ab26fa30639",
  message: {
    query: "sum by (instance) (go_memstats_alloc_bytes{job=\"node\"})",
    queryType: "range",
  }
}
```

```
    start: "1683308700000",
    end: "1683913500000",
    step: "300000",
    samples: "11496",
    userAgent: "AWSPrometheusDPJavaClient/2.0.436.0 ",
    dashboardUid: "11234",
    panelId: "12"
  },
  component: "query-frontend"
}
```

限制

策略大小限制- CloudWatch 日志资源策略限制为 5120 个字符。当 CloudWatch Logs 检测到策略接近大小限制时，它会自动启用以开头的日志组 `/aws/vendedlogs/`。启用查询日志时，适用于 Prometheus 的亚马逊托管服务必须使用您指定的日志组更新 CloudWatch 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/`。

了解并优化 Amazon Managed Service for Prometheus 的成本

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 \[https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config\]\(https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config\) 在 Prometheus 文档中。](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。

要更深入地了解对您的查询费用影响最大的查询，请参阅[在 Amazon Managed Service for Prometheus 中管理查询成本](#)。

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。

有关如何配置工作区的保留期的信息，请参阅[配置工作区](#)。

如何降低警报查询成本？

警报功能会对数据进行查询，从而增加查询成本。以下是一些可以用来优化警报查询并降低成本的策略。

- 使用 Amazon Managed Service for Prometheus 警报 – Amazon Managed Service for Prometheus 外部的警报系统可能需要额外的查询来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 中发出警报，以实现高可用性。这可能会使您的成本增加三倍或更多。Amazon Managed Service for Prometheus 中的警报功能经过优化，将以最少的查询次数为您提供高可用性和弹性。

我们建议使用 Amazon Managed Service for Prometheus 中的原生警报，而不是外部警报系统。

- 优化警报间隔 - 优化警报查询的一种快速方法是增加自动刷新闻隔。如果您有一个每分钟查询一次的警报，但只需要每五分钟查询一次，那么增加自动刷新时间间隔可以为您节省五倍的警报查询成本。
- 使用最佳回溯 - 查询中的回溯窗口越大，查询的成本就越高，因为它需要提取更多的数据。确保 PromQL 查询中的回溯窗口大小与需要警报的数据相匹配。例如，在下面的规则中，表达式包括一个十分钟的回溯窗口：

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

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

一般来说，查看您的警报规则，确保针对服务的最佳指标发出警报。在同一指标上创建重叠警报或提供相同信息的多个警报非常容易，尤其是当您随着时间的推移添加警报时。如果您发现经常有几组警报同时发出，您可以优化警报，不将所有警报都包括在内。

这些建议有助于您降低成本。最终，您必须在成本与创建一套正确的警报以了解系统状态之间取得平衡。

有关 Amazon Managed Service for Prometheus 中警报的更多信息，请参阅[使用警报管理器在 Amazon Managed Service for Prometheus 中管理和转发警报](#)。

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

我可以使用哪些指标来监控我的成本？

您采集的指标样本是亚马逊 Prometheus 托管服务的主要成本驱动因素。摄取的样本数量直接决定了您的每月费用，因此监控和了解您的摄入模式至关重要。

[AWS Cost Explorer](#)是监控您的亚马逊托管服务 Prometheus 成本的真实来源。您可以从多个维度监控 Cost Explorer， day-by-day了解适用于Prometheus的亚马逊托管服务的历史和成本趋势，包括摄取的样本。[AWS 成本异常检测](#)还可以让您监控支出模式的意外变化。

使用IngestionRate指标提供了一种辅助方法来监控与成本直接相关的摄取趋势。IngestionRate用作附加指标的好处包括：

- 工作空间级别的跟踪 — 以每个工作空间为基础监控摄取情况，而不仅仅是账户级别。
- 精细可见性 — 每小时甚至每分钟跟踪一次摄取模式， minute-by-minute以获得实时见解。
- 主动监控-设置 CloudWatch警报，在使用量峰值出现在账单中之前对其进行检测。

Note

IngestionRate可用于估算成本和趋势或归因每个工作空间的成本，但并非 100% 准确。由于IngestionRate报告每隔 1 分钟采样的平均速率，因此将该速率乘以时间可以得出摄取的样本的近似值，而不是确切的样本数。此外，Amaz CloudWatch on 的数据保留政策会影响历史查询的可用粒度，超过 63 天的数据间隔限制为 1 小时。

有关监控 Amazon Prometheus 托管服务指标的更多信息，请参阅。CloudWatch [使用 CloudWatch 指标监控亚马逊托管服务的 Prometheus 资源](#)

如何查看我的费用 AWS Cost Explorer ?

作为亚马逊Prometheus托管服务费 AWS Cost Explorer 用的真实来源，提供您对Prometheus样本的亚马逊托管服务的实际账单使用量和费用，包括按月份和地区划分的历史账单数据。使用 Cost Explorer 查看最终账单金额和 day-by-day成本趋势。

要查看您的 Amazon Prometheus 托管服务费用，请执行以下操作：

访问权限 AWS Cost Explorer

1. 登录到 AWS 管理控制台。
2. 导航至“账单和成本管理”控制面板。
3. 从左侧导航菜单中选择 Cost Explorer。
4. 选择 Launch Cost Explorer (如果你是第一次使用它) 。

配置报告

1. 将您的时间范围设置为所需的账单周期 (例如，2025 年 3 月至 2026 年 2 月) 。
2. 在“筛选器”下，选择：
 - 服务：选择“适用于 Prometheus 的亚马逊托管服务”。
 - 使用类型：筛选“MetricSampleCount”以隔离样本摄取费用。

对数据进行分组和查看

1. 在“分组依据”下，选择“区域”以查看每个区域的成本和使用情况数据。
2. 选择您的首选可视化效果 (条形图、折线图或表格) 。
3. 选择“应用”以生成报告。

导出数据 (可选)

1. 选择右上角的下载 CSV 以导出数据。
2. CSV 文件将包含：账单周期、区域、使用类型、账单金额和使用量 (计费样本数量) 。

Note

Cost Explorer 的数据通常会延迟 24 小时。对于最新的账单周期，数据可能要等到第二天才可用。

如何计算一个月内摄入的样本数量？

您可以使用带有 Amazon CloudWatch 的 IngestionRate 指标来计算采集的大致样本数量。AWS Command Line Interface 这对于查看月度账单和了解跨工作空间的使用模式非常有用。

要检索摄取数据，请执行以下操作：

```
aws cloudwatch get-metric-data \  
  --region your-region \  
  --start-time start-timestamp \  
  --end-time end-timestamp \  
  --metric-data-queries '[  
    {  
      "Id": "e1",  
      "Expression": "SUM(METRICS())",  
      "Period": 3600  
    },  
    {  
      "Id": "ws1",  
      "MetricStat": {  
        "Metric": {  
          "Namespace": "AWS/Usage",  
          "MetricName": "ResourceCount",  
          "Dimensions": [  
            {"Name": "Service", "Value": "Prometheus"},  
            {"Name": "Resource", "Value": "IngestionRate"},  
            {"Name": "Type", "Value": "Resource"},  
            {"Name": "Class", "Value": "None"},  
            {"Name": "ResourceId", "Value": "YOUR_AMP_WORKSPACE_ID" }  
          ]  
        },  
        "Period": 3600,  
        "Stat": "Average"  
      }  
    }  
  ]
```

```
]'
```

该命令返回每小时平均IngestionRate值，以每秒采样数为单位。要计算一个月内摄取的样本的大致数量，请将每个小时的数据点乘以 3600（每小时秒），得出该小时内摄取的样本，然后将该月所有每小时的总量相加：

```
Monthly samples  $\approx \sum$  (hourly IngestionRate average  $\times$  3600)
```

例如，如果一个小时平均IngestionRate每秒返回 500 个样本，则该小时贡献了大约 $500 \times 3600 = 1,800,000$ 个样本。每月每小时重复此操作，然后将结果相加，得出大概的每月摄取次数。

关键参数：

- Period: 3600（以秒为单位 1 小时）
- StartTime: 您的月初（例如，2026-02-01T00:00:00Z）
- EndTime: 您的月底（例如，2026-03-01T00:00:00Z）
- Stat: 平均

要找到您的工作空间，请执行 IDs 以下操作：

```
aws amp list-workspaces --region your-region
```

使用工作空间 ID 筛选指标以仅显示指定工作空间的数据，而不是聚合该区域的所有 Prometheus 资源。

哪些数据粒度可用于历史成本分析？

Amaz CloudWatch on 的数据保留政策会影响历史查询的可用粒度：

- 存储时间@@ 少于 15 天的数据：以 1 分钟为间隔进行查询 (Period: 60)
- 15—63 天前的数据：以 5 分钟为间隔进行查询 (Period: 300)
- 超过 63 天的数据：限制在 1 小时间隔内 (Period: 3600)

对于超过 63 天的历史分析，CloudWatch 自动将数据缩减至至少 1 小时的时间段。在查看超过 63 天的月份的账单时，必须使用每小时的汇总数据。月度样本计算使用这些小时平均值的数据点，将每个值相加乘以整个月的 3600。

这种精细度的降低进一步促成了为什么要为旧数据IngestionRate提供估算值而不是精确计数——请务必参阅 [Cost Explorer](#) 了解您的权威账单金额。

有关 CloudWatch 指标保留的更多详情，请参阅 Amazon CloudWatch 用户指南中的 [指标保留率](#)。

监控亚马逊托管服务 Prometheus 成本的最佳做法是什么？

为了有效管理和优化您的 Amazon 托管服务 Prometheus 支出，请考虑实施以下监控实践：

- 定期监控 Cost Explorer，跟踪实际支出趋势，识别多个维度的成本异常，包括摄取的样本。
- 启用“AWS 成本异常检测”以接收提醒，提醒您的亚马逊托管服务 Prometheus 支出出现意外成本增加。
- 将 CloudWatch 警报设置IngestionRate为开启，以便在工作空间级别进行监控，并尽早发现摄入峰值。
- 定期导出 Cost Explorer 数据以进行长期成本分析和报告。

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

- 等级是针对区域内的使用情况，而不是跨区域的使用情况。一个区域内的使用量必须达到下一等级，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）。当一个组织内所有账户的摄取指标总量（在一个区域内）达到下一等级时，所有账户都将按照较低的费率收费。

我删除了我所有的 Amazon Managed Service for Prometheus 工作区，但似乎仍在收费。这可能是怎么回事？

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 [查找和删除抓取程序](#)。

与其他 AWS 服务集成

Amazon Managed Service for Prometheus 与其他 AWS 服务集成。本节介绍如何与 Amazon Elastic Kubernetes Service (Amazon EKS) 成本监控 (具有 Kubecost) 集成，以及如何使用 Amazon Data Firehose 从 CloudWatch 摄取指标。本节还介绍如何使用 AWS Observability Accelerator Terraform 模块或使用 AWS Controllers for Kubernetes 来设置和管理 Amazon Managed Service for Prometheus。

主题

- [与 Amazon EKS 成本监控集成](#)
- [使用可观测性加速器为 Prometheus 设置亚马逊托管服务 AWS](#)
- [使用适用于 Kubernetes 的控制器管理适用于 Prometheus 的亚马逊托管服务](#)
- [将 CloudWatch 指标与 Amazon Managed Service for Prometheus 集成](#)

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](#)。

使用可观测性加速器为 Prometheus 设置亚马逊托管服务 AWS

AWS 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、警报和控制面板。这包括适用于 Prometheus 的亚马逊托管服务、[亚马逊托管 Grafana](#)、[Distro for](#) 和其他[AWS 工具](#)。OpenTelemetry [为了帮助您一起使用这些工具，AWS 提供了名为“可观察性加速器”的 Terraform 模块，用于通过这些服务配置可观测性。AWS](#)

AWS Observability Accelerator 为适用于 Prometheus 的亚马逊托管服务提供了两个收集器配置文件：

- 托管指标 (无代理) — 使用适用于 [Prometheus 的 Amazon 托管服务收集器](#)，这是一种在集群外部运行的完全托管的无代理抓取工具。不需要管理的收集器 pod。仅限指标。
- 自我管理-通过 Helm 在集群中部署 OpenTelemetry 收集器。支持指标、跟踪 (AWS X-Ray) 和日志 (Amazon CloudWatch)。

本节将介绍这两个选项，首先介绍推荐的无代理方法。

Terraform 模板和详细说明可以在[适用于 Terraform AWS 的可观察性加速器页面](#)上找到。GitHub

先决条件

要使用 AWS 可观测性加速器，您必须拥有现有 Amazon EKS 集群，并且必须满足以下先决条件：

- [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](#) (>= 1.5.0) — 用于自动为该解决方案创建资源。您必须为 AWS 提供商设置一个 IAM 角色，该角色有权在您的账户中创建和管理适用于 Prometheus 的亚马逊托管服务、Amazon Managed Grafana 和 IAM。AWS 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 provider](#)。

使用托管指标（无代理）示例

此示例使用适用于 Prometheus 的亚马逊托管服务收集器从您的 Amazon EKS 集群中抓取 Prometheus 指标，而无需部署任何收集器 pod。收集器需要在两个不同的可用区中至少有两个子网。有关更多详细信息，请参阅中的[eks-amp-managed](#)示例 GitHub。

使用无代理基础设施监控 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/eks-amp-managed  
  
terraform init
```

3. 创建一个新 terraform.tfvars 文件，如以下示例所示。使用您的 Amazon EKS 集群的 AWS 区域、集群 ID 和 VPC 联网详细信息。收集器需要在两个不同的可用区中至少有两个子网。

```
# (mandatory) AWS Region where your resources will be located  
aws_region = "eu-west-1"  
  
# (mandatory) EKS Cluster name  
eks_cluster_id = "my-eks-cluster"  
  
# (mandatory) Subnets for the managed scraper (>= 2 AZs)  
scraper_subnet_ids = ["subnet-aaa", "subnet-bbb"]  
  
# (mandatory) Security group allowing scraper access to the EKS API  
scraper_security_group_ids = ["sg-xxx"]
```

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。你需要用你的 Grafana 工作区中的 ID 替换。

```
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] 要使用适用于 Prometheus 的现有亚马逊托管服务工作空间，请将 ID 添加到 `terraform.tfvars` 文件中，如下例所示，将替换为您的 Prometheus 工作空间 ID。 `prometheus-workspace-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，包括以下资源：

- 一个新的 Amazon Managed Service for Prometheus 工作区（除非您选择使用现有工作区）。
- 适用于 Prometheus 的亚马逊托管服务收集器（无代理抓取器），配置为从您的亚马逊 EKS 集群中抓取 Prometheus 指标。
- Prometheus 在您的 Prometheus 亚马逊托管服务工作区中记录和提醒规则。
- kube-state-metrics 并在您的 Amazon EKS 集群中部署用于基础设施指标的节点导出器。
- 您当前工作区中的全新 Amazon Managed Grafana 数据来源和控制面板。仪表板将列在 EKS 监控下。

备选方案：自行管理 OpenTelemetry 的收集器

如果您需要跟踪、日志或对收集管道的完全控制，请使用自我管理的配置文件。这将通过 Helm 在你的 Amazon EKS 集群中部署一个 OpenTelemetry 收集器，该收集器配置为抓取 Prometheus 指标并远程写入适用于 Prometheus 的亚马逊托管服务。它还支持跟踪 (AWS X-Ray) 和日志 (Amazon CloudWatch)。有关更多详细信息，请参阅中的 [eks-amp-otel](#) 示例 GitHub。

使用自我管理的 Terraform 模块

1. 克隆存储库并初始化 Terraform。

```
git clone https://github.com/aws-observability/terraform-aws-observability-
accelerator.git
cd examples/eks-amp-otel
terraform init
```

2. 创建一个新 terraform.tfvars 文件，如以下示例所示。

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

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

3. 使用与托管指标示例相同的步骤设置 Amazon Managed Grafana 工作空间和 API 密钥（上面的步骤 4-6）。
4. 使用以下命令部署解决方案。

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

这将在您的 AWS 账户中创建以下资源（与无代理方法不同，收集器在您的集群内运行）：

- 适用于 Prometheus 工作空间的亚马逊托管服务（如果未提供）。
- 包含数据源和控制面板的 Amazon 托管 Grafana 工作空间。
- 通过 Helm 在你的 Amazon EKS 集群中部署的 OpenTelemetry 收集器，配置为抓取 Prometheus 指标并远程写入适用于 Prometheus 的亚马逊托管服务。
- OpenTelemetry 收集器的服务账户 (IRSA) 的 IAM 角色。
- 将管道追踪到 AWS X-Ray（默认启用）。
- 发送至 Amazon 的日志管道 CloudWatch（默认启用）。

查看控制面板

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。基础架构仪表板由 Terraform 自动配置。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](#)。

使用适用于 Kubernetes 的控制器管理适用于 Prometheus 的 Amazon EKS 的亚马逊托管服务

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。您可以使用 Kubernetes 的 AWS 控制器自定义资源定义 (CRDs) 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 Amazon 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 Prometheus 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 Kubernetes 的 AWS 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。请记住用您账户的正确值替换 `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 使用 *region* 适合您系统的正确值，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 | jq -r '.tag_name | ltrimstr("v")'`
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`。`WORKSPACE-ID` 替换为上一步中的工作空间 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`。 *WORKSPACE-ID* 替换为上一步中的工作空间 ID。 *TOPIC-ARN* 替换为要向其发送通知的 Amazon SNS 主题的 ARN， *REGION* 以及您正在使用 AWS 区域的。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`、`RuleGroupNamespace`。 *ResourceName* 替换为要删除的资源名称。

```
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`。`WORKSPACE-ID` 替换为上一节中的工作空间 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`。`account` 用您的账户 ID、`workspaceURL`/您刚刚找到的 URL 以及 `region` AWS 区域 适合您系统的相应网址替换。

```
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 `PrometheusChartNamePrometheusNamespace`、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。`Region` 替换为您 AWS 区域正在使用的，以及 `workspaceURL`/您在步骤 1 中找到的 URL。

```
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 (CRDs)，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 Amazon Managed Service for Prometheus 集成

将所有指标集中在一个地方会有所帮助。Amazon Managed Service for Prometheus 不会自动摄取 Amazon CloudWatch 指标。不过，您可以使用 Amazon Data Firehose 和 AWS Lambda 将 CloudWatch 指标推送到 Amazon Managed Service for Prometheus。

本节介绍如何检测 [Amazon CloudWatch 指标流](#) 以及使用 [Amazon Data Firehose](#) 和 [AWS Lambda](#) 将指标摄取到 Amazon Managed Service for Prometheus。

您将使用 [AWS 云开发工具包 \(CDK\)](#) 设置堆栈，以创建 Firehose 传输流、Lambda 和 Amazon S3 存储桶来演示完整的场景。

基础设施

首先，您必须为该配方设置基础设施。

通过 CloudWatch 指标流可以将流式指标数据转发到 HTTP 终端节点或 [Amazon S3 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。

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-ol1ly-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命令：

```
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 流

现在，您已经有一个用来处理这些指标的 lambda 函数，您可以从 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。

现在，您的指标应该在从 Amazon CloudWatch 传输到 Amazon Managed Service for Prometheus。

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 [Amazon CloudWatch 控制台](#) 删除 Amazon CloudWatch 指标流。

Amazon Managed Service for Prometheus 中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。AWS 还为您提供可以安全使用的服务。作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于亚马逊 Prometheus 托管服务的合规计划，[AWS 请参阅按合规计划划分的范围内服务按合规AWS 计划划分的范围内的](#)。
- 云端安全-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，包括您的数据的敏感性、您的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。您还将学习如何使用其他 AWS 服务来帮助您监控和保护适用于 Prometheus 的亚马逊托管服务资源。

主题

- [Amazon Managed Service for Prometheus 的数据保护](#)
- [Amazon Managed Service for Prometheus 的身份和访问管理](#)
- [IAM 权限和策略](#)
- [Amazon Managed Service for Prometheus 的合规性验证](#)
- [Amazon Managed Service for Prometheus 中的数据恢复](#)
- [Amazon Managed Service for Prometheus 的基础设施安全性](#)
- [使用 Amazon Managed Service for Prometheus 的服务相关角色](#)
- [使用记录适用于 Prometheus 的亚马逊托管服务 API 调用 AWS CloudTrail](#)
- [设置服务账户的 IAM 角色](#)
- [将 Amazon Managed Service for Prometheus 与接口 VPC 终端节点结合使用](#)

Amazon Managed Service for Prometheus 的数据保护

AWS [分担责任模式](#)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结构上的内

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[数据隐私常见问题解答AWS](#)条款。有关欧洲数据保护的信息，请参阅[通用数据保护条例 \(GDPR\) 中心](#)。

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。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)。
- 用于 SSL/TLS 与 AWS 资源通信。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。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的[使用跟 CloudTrail 踪](#)。
- 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 140-3 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 140-3 版》<https://aws.amazon.com/compliance/fips/>。

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 URL，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](#)
- [静态加密](#)

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

Amazon Managed Service for Prometheus 支持创建客户自主管理型密钥，用于加密数据。有关 Amazon Managed Service for Prometheus 默认使用的密钥以及如何使用您自己的客户自主管理型密钥的更多信息，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
 - 建立和维护 IAM 策略和授权
 - 启用和禁用密钥策略

- 轮换加密材料
- 添加 标签
- 创建密钥别名
- 安排密钥删除

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不能转换为使用 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策略、禁用密钥或撤销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，或者在密钥策略中删除了 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 管理控制台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自主管理型密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管式密钥时，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，密钥策略中必须支持以下 API 操作：

- [kms:CreateGrant](#)：向客户托管密钥添加授权。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` 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》。

步骤 2：为 Amazon Managed Service for Prometheus 指定客户自主管理型密钥

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。

步骤 3：从其他服务（如 Amazon Managed Grafana）访问数据

此步骤是可选的 – 只有当您需从其他服务访问 Amazon Managed Service for Prometheus 数据时才需要此步骤。

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，如果要使用 Amazon Managed Grafana 根据数据创建控制面板或警报，则必须授予 Amazon Managed Grafana 访问密钥的权限。

让 Amazon Managed Grafana 可以访问客户自主管理型密钥

1. 在 [Amazon Managed Grafana 工作区列表](#) 中，选择要能够访问 Amazon Managed Service for Prometheus 的工作区的名称。这将显示有关 Amazon Managed Grafana 工作区的摘要信息。
2. 记下您的工作区使用的 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。您将在后面的步骤中在 AWS KMS 控制台中指定此名称。
3. 在 [AWS KMS 客户自主管理型密钥列表](#) 中，选择您在创建 Amazon Managed Service for Prometheus 工作区时使用的客户自主管理型密钥。这将打开密钥配置详细信息页面。
4. 在密钥用户旁边，选择添加按钮。

5. 从名称列表中选择上面提到的 Amazon Managed Grafana IAM 角色。为了方便查找，您还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 Amazon Managed Grafana 工作区现在可以访问 Amazon Managed Service for Prometheus 工作区中的数据。您可以在密钥用户中添加其他用户或角色，使其他服务也能访问您的工作区。

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上下文，其中密钥 `aws:amp:arn` 为工作空间的亚马逊资源名称 (ARN)，值为工作空间的 [亚马逊资源名称](#) (ARN)。

Example

```
"encryptionContext": {
  "aws:amp: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

以下是密钥策略语句示例，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 logs 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": "57f5dbec-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"  
}
```

了解详情

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](#)
- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [Amazon Managed Service for Prometheus 如何与 IAM 配合使用](#)
- [Amazon Managed Service for Prometheus 的基于身份的策略示例](#)
- [Amazon Managed Service for Prometheus 身份和访问故障排除](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅[Amazon Managed Service for Prometheus 身份和访问故障排除](#)）

- 服务管理员：确定用户访问权限并提交权限请求（请参阅[Amazon Managed Service for Prometheus 如何与 IAM 配合使用](#)）
- IAM 管理员：编写用于管理访问权限的策略（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 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录用户指南》中的[如何登录您的 AWS 账户](#)。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的[适用于 API 请求的 AWS 签名版本 4](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，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，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service，或者 AWS 服务使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。AWS IAM Identity Center 有关更多信息，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

[IAM 用户](#)是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的[要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问](#)。

[IAM 组](#)指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 用户使用案例](#)。

IAM 角色

[IAM 角色](#)是具有特定权限的身份，可提供临时凭证。您可以通过[从用户切换到 IAM 角色（控制台）](#)或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的[担任角色的方法](#)。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 中的跨账户资源访问](#)。

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。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 [JSON 策略概述](#)。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管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。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。您必须在基于资源的策略中[指定主体](#)。

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。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 实体的权限边界](#)。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的 [服务控制策略](#)。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的 [资源控制策略 \(RCPs\)](#)。
- 会话策略 – 在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。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	是
策略操作	是
策略资源	是
策略条件密钥	否
ACLs	否

IAM 功能	Amazon Managed Service for Prometheus 支持
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
服务关联角色	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[使用客户管理型策略定义自定义 IAM 权限](#)。

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。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：是

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。

要启用跨账户访问，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。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 中的跨账户资源访问](#)。

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。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。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN\)](#) 指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

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

要查看适用于 Prometheus 的亚马逊托管服务资源类型 ARNs 及其列表，请参阅服务授权参考中的[亚马逊托管服务为 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键：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。

Condition 元素根据定义的条件指定语句何时执行。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。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](#)。

适用于 Prometheus 的亚马逊托管服务中的访问控制列表 (ACLs)

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。ACLs 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

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

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 [使用 ABAC 授权定义权限](#)。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

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。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 用户指南》中的 [IAM 中的临时安全凭证](#) 和 [使用 IAM 的 AWS 服务](#)

Amazon Managed Service for Prometheus 的转发访问会话

支持转发访问会话 (FAS)：否

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。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

支持服务关联角色：是

服务相关角色是一种与服务相关联的 AWS 服务角色。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 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关联角色的权限。

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。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。

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 \(控制台\)](#)。

有关亚马逊托管服务为 Prometheus 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅 [ARNs 《服务授权参考》中的 Amazon Prometheus 托管服务的操作、资源和条件密钥](#)。

主题

- [策略最佳实践](#)
- [使用 Amazon Managed Service for Prometheus 控制台](#)
- [允许用户查看他们自己的权限](#)

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 策略设置权限时，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 策略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 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 策略，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 策略语言（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 [使用 IAM Access Analyzer 验证策略](#)。
- 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 [使用 MFA 保护 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 中的安全最佳实践](#)。

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式策略。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:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

Amazon Managed Service for 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，指明您无权执行某个操作，则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，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

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
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的 [为经过外部身份验证的用户 \(身份联合验证\) 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之间的差别，请参阅《IAM 用户指南》中的 [IAM 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。这些证书必须具有执行操作和访问资源的权限，例如检索有关您的云 AWS 资源的亚马逊托管服务 Prometheus 数据。以下各节详细介绍了如何使用 AWS Identity and Access Management (IAM) 和 Amazon Prometheus 托管服务，通过控制谁可以访问资源来帮助保护您的资源。有关更多信息，请参阅 [IAM 中的策略和权限](#)。

Amazon Managed Service for Prometheus 权限

要查看可能的 Amazon Managed Service for Prometheus 操作、资源类型和条件键的列表，请参阅 [Amazon Managed Service for Prometheus 的操作、资源和条件键](#)。

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。

JSON

```
{
  "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。有关您在使用时的合规责任的更多信息 AWS 服务，请参阅[AWS 安全文档](#)。

Amazon Managed Service for Prometheus 中的数据恢复

AWS全球基础架构围绕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、容错能力和可扩展性。

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 云安全性](#)。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。

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他 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，因为您无需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[使用 IAM 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[服务关联角色文档](#)。

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[服务关联角色权限](#)。

为 Amazon Managed Service for Prometheus 创建服务相关角色

您无需手动创建服务关联角色。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 管理控制台

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，此服务关联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](#)。

如果您删除该服务关联角色，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。AWS Service RoleForAmazonPrometheusScraper 创建服务关联角色后，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[编辑服务关联角色](#)。

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 管理控制台、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](#)。

使用记录适用于 Prometheus 的亚马逊托管服务 API 调用 AWS CloudTrail

适用于 Prometheus 的亚马逊托管服务 [AWS CloudTrail](#) 与一项服务集成，该服务可记录用户、角色或用户所采取的操作。AWS 服务 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。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、发出请求的时间以及其他详细信息。

每个事件或日志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：

- 请求是使用根用户凭证还是用户凭证发出的。
- 请求是否代表 IAM Identity Center 用户发出。
- 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。

CloudTrail 在您创建账户 AWS 账户 时在您的账户中处于活动状态，并且您自动可以访问 CloudTrail 活动历史记录。CloudTrail 事件历史记录提供了过去 90 天中记录的管理事件的可查看、可搜索、可下载且不可变的记录。AWS 区域有关更多信息，请参阅《AWS CloudTrail 用户指南》中的 [“使用 CloudTrail 事件历史记录”](#)。查看活动历史记录不 CloudTrail 收取任何费用。

要持续记录 AWS 账户 过去 90 天内的事件，请创建跟踪或 [CloudTrail Lake](#) 事件数据存储。

CloudTrail 步道

跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。使用创建的所有跟踪 AWS 管理控制台 都是多区域的。您可以通过使用 AWS CLI 创建单区域或多区域跟踪。建议创建多区域跟踪，因为您可以捕获账户 AWS 区域 中的所有活动。如果您创建单区域跟踪，则只能查看跟踪的 AWS 区域中记录的事件。有关跟踪的更多信息，请参阅《AWS CloudTrail 用户指南》中的 [为您的 AWS 账户创建跟踪](#) 和 [为组织创建跟踪](#)。

通过创建跟踪，您可以免费将正在进行的管理事件的一份副本传送到您的 Amazon S3 存储桶，但会收取 Amazon S3 存储费用。CloudTrail 有关 CloudTrail 定价的更多信息，请参阅[AWS CloudTrail 定价](#)。有关 Amazon S3 定价的信息，请参阅 [Amazon S3 定价](#)。

CloudTrail 湖泊事件数据存储

CloudTrail Lake 允许你对自己的事件运行基于 SQL 的查询。CloudTrail Lake 将基于行的 JSON 格式的现有事件转换为 [Apache ORC](#) 格式。ORC 是一种针对快速检索数据进行优化的列式存储格式。事件将被聚合到事件数据存储中，它是基于您通过应用[高级事件选择器](#)选择的条件的不可变的事件集合。应用于事件数据存储的选择器用于控制哪些事件持续存在并可供您查询。有关 CloudTrail Lake 的更多信息，[请参阅 AWS CloudTrail 用户指南中的使用 AWS CloudTrail Lake](#)。

CloudTrail 湖泊事件数据存储和查询会产生费用。创建事件数据存储时，您可以选择要用于事件数据存储的[定价选项](#)。定价选项决定了摄取和存储事件的成本，以及事件数据存储的默认和最长保留期。有关 CloudTrail 定价的更多信息，请参阅[AWS CloudTrail 定价](#)。

适用于 Prometheus 的亚马逊托管服务管理活动 CloudTrail

[管理事件](#)提供有关对中的资源执行的管理操作的信息 AWS 账户。这些也称为控制面板操作。默认情况下，CloudTrail 记录管理事件。

Amazon Managed Service for Prometheus 会将所有 Amazon Managed Service for Prometheus 控制面板操作记录为管理事件。[有关亚马逊 Prometheus 托管服务控制平面操作的列表，请参阅亚马逊 Prometheus 托管服务 API 参考 CloudTrail](#)。

Amazon Managed Service for Prometheus 事件示例

事件代表来自任何来源的单个请求，包括有关所请求的 API 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，因此事件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"
      }
    },
    "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":
    "YWxlcnRtYW5hZ2VyX2NvbWZpZzogaAogIGdsb2JhbDoKICAgIHhtdHBfc21hcnRob3N00iAnbG9jYWxob3N00jI1JwogI",
    "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":
      "Z3JvdXBz0gogIC0gYmFtZTogdGVzdFJ1bGVHcm91cHN0YW1lc3BhY2UKICAgIHJ1bGVz0gogICAgLSBhbGVydDogdGVzd
      "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"
}
```

有关 CloudTrail 录音内容的信息，请参阅《AWS CloudTrail 用户指南》中的[CloudTrail 录制内容](#)。

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，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](#)。

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

如果您使用亚马逊虚拟私有云（亚马逊 VPC）托管 AWS 资源，则可以在您的 VPC 和适用于 Prometheus 的亚马逊托管服务之间建立私有连接。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，可用于在您定义的虚拟网络中启动 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](#)。

Interface VPC 终端节点由 AWS PrivateLink 一种 AWS 技术提供支持，该技术使用带有私有 IP 地址的弹性网络接口实现 AWS 服务之间的私密通信。有关更多信息，请参阅“[新增 AWS 服务](#)”博客文章。
AWS PrivateLink

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 APIs 兼容的服务。有关更多信息，请参阅《适用于 [Prometheus 的亚马逊托管服务用户指南 APIs](#)》中的 Prometheus 兼容内容。

- `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《适用于 [Prometheus 的亚马逊托管服务用户指南](#)》中的亚马逊 APIs Prometheus 托管服务。

Note

如果您在无法直接访问互联网的 VPC 中使用 `remote_write`，则还必须为其创建接口 VPC 终端节点 AWS Security Token Service，以允许 `sigv4` 通过该终端节点工作。有关为创建 VPC 终端节点的信息 AWS STS，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](#)。您必须设置 AWS STS 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 step-by-step 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。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AmazonManagedPrometheusPermissions",
      "Effect": "Allow",
      "Action": [
        "aps:DescribeWorkspace",
        "aps:DescribeRuleGroupsNamespace",
        "aps:ListRuleGroupsNamespaces",
        "aps:ListWorkspaces"
      ],
      "Resource": "arn:aws:aps:*:*:/workspaces*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/PromUser"
        ]
      }
    }
  ]
}
```

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"
        }
      }
    }
  ]
}
```

对 Amazon Managed Service for Prometheus 错误进行故障排除

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 或超出限制错误](#)
- [我看到重复的样本](#)
- [我看到关于样本时间戳的错误](#)
- [我看到一条与限制相关的错误消息](#)
- [您的本地 Prometheus 服务器输出超出了限制。](#)
- [我的一些数据没有出现](#)

429 或超出限制错误

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
```

如果您看到类似下面示例的 429 错误，则说明您的请求已超过 Amazon Managed Service for Prometheus 对使用与 Prometheus 兼容的 RemoteWrite API 向工作区发送数据的速率（每秒事务数）的配额。

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

如果您看到类似下面示例的 400 错误，则说明您的请求已超出 Amazon Managed Service for Prometheus 的活动时间序列配额。有关如何处理活动时间序列配额的详细信息，请参阅[活跃系列默认配额](#)。

```
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 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](#)。

有关复制数据的其他问题将在下一节讨论。

我看到关于样本时间戳的错误

Amazon Managed Service for Prometheus 按顺序摄取数据，并希望每个样本的时间戳都晚于前一个样本。

如果数据没有按顺序到达，则会出现以下错误：`out-of-order samples`、`duplicate sample for timestamp` 或 `samples with different value but same timestamp`。这些问题通常是由于向 Amazon Managed Service for Prometheus 发送数据的客户端设置不正确造成的。如果使用的是在代理模式下运行的 Prometheus 客户端，请检查配置中是否有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，请检查它们是否错乱。

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 Out-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](#)。

- `<value>` 已超出每用户系列限制，请联系管理员提高该限制
- `<value>` 已超出每指标系列的限制，请联系管理员提高
- 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	<code>prometheus_tsdb_head_series</code>
当前摄取率	<code>rate(prometheus_tsdb_head_samples_appended_total[5m])</code>
Most-to-least 每个指标名称的活动系列列表	<code>sort_desc(count by(__name__))</code> <code>({__name__!=""})</code>
每个指标系列的标签数	<code>group by(mylabelname)</code> <code>({__name__!=""})</code>

我的一些数据没有出现

由于各种原因，发送到 Amazon Managed Service for Prometheus 的数据可能会被丢弃。下表列出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 [使用 CloudWatch 指标监控亚马逊托管服务的 Prometheus 资源](#)。

Reason	含义
greater_than_max_sample_age	丢弃比当前时间更早的日志行
new-value-for-timestamp	发送重复样本的时间戳与上一个样本的时间戳相同，但值不同。
per_metric_series_limit	用户已达到每个指标活跃系列数上限
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

Amazon Managed Service for Prometheus 中的标记

标签是您或 AWS 分配给 AWS 资源的自定义属性标签。每个 AWS 标签分为两部分：

- 标签键（例如，CostCenter、Environment、Project 或 Secret）。标签密钥区分大小写。
- 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。

标签可帮助您识别和整理 AWS 资源。许多 AWS 服务都支持标记，因此您可以为来自不同服务的资源分配相同的标签，以表明这些资源是相关的。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 APIs、或 SDKs 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)。
- 如果您的标记架构用于多个 AWS 服务和资源，请记住，其他服务可能对允许的字符有限制。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）。这些仅供 AWS 使用。无法编辑或删除带此前缀的标签键或值。带此前缀的标签不计入您的 tags-per-resource 限制。

主题

- [标记 Amazon Managed Service for Prometheus 工作区](#)
- [标记规则组命名空间](#)

标记 Amazon Managed Service for Prometheus 工作区

标签是可分配给资源的自定义标签。它们包括一个唯一键和一个可选值（键值对）。标签有助于您标识和组织 AWS 资源。在 Amazon Managed Service for Prometheus 中，可以对工作区（和规则组命名空间）进行标记。您可以使用控制台 APIs、AWS CLI 或 SDKs 为这些资源添加、管理和移除标签。除了使用标签标识、组织和跟踪工作区外，您还可以在 IAM 策略中使用标签来控制谁可以查看您的 Amazon Managed Service for Prometheus 资源并与之交互。

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](#)
- [查看工作区的标签](#)
- [编辑工作区的标签](#)
- [从工作区中删除标签](#)

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 向适用于 Prometheus 的亚马逊托管服务工作区添加标签。

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 Amazon Managed Service for Prometheus 工作区](#)。

主题

- [向工作区添加标签 \(控制台 \)](#)
- [向工作区添加标签 \(AWS CLI \)](#)

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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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. 在键中，输入标签的名称。您可以在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)

按照以下步骤使用向适用于 Prometheus 的亚马逊托管服务工作区添加标签。AWS CLI 要在创建工作区时为其添加标签，请参阅[创建 Amazon Managed Service for Prometheus 工作区](#)。

在这些步骤中，我们假设您已经安装了最新版本 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，要为名为 My-Workspace 的 Amazon Prometheus 托管服务工作空间添加两个标签，一个标签键 `Status` 以标签值命名，一个标签键名为 `Secret` 的标签值为 `:TeamMy-Team`

```
aws amp tag-resource --resource-arn arn:aws:aps:us-west-2:123456789012:workspaces/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

标签可以帮助您识别和整理 AWS 资源并管理对资源的访问权限。有关标记策略的更多信息，请参阅为资源[添加标签](#)。AWS

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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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）

按照以下步骤使用 AWS CLI 来查看工作空间的 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。您也可以更改键的名称，这相当于移除当前的标签并使用新名称和与另一个键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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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. 在键中，输入标签的名称。您可以在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 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。移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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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）。

按照以下步骤使用从 AWS CLI 工作区中移除标签。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，要在名为 My-Workspace 的工作空间上删除带有标签键的标签，请执行以下操作：`Status`

```
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

标签是可分配给资源的自定义标签。它们包括一个唯一键和一个可选值 (键值对)。标签有助于您标识和组织 AWS 资源。在 Amazon Managed Service for Prometheus 中，可以对规则组命名空间 (和工作区) 进行标记。您可以使用控制台 APIs、AWS CLI 或 SDKs 为这些资源添加、管理和移除标签。除了通过标签标识、组织和跟踪规则组命名空间外，您还可以在 IAM 策略中使用标签来控制谁可以查看您的 Amazon Managed Service for Prometheus 资源并与之交互。

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](#)
- [查看规则组命名空间的标签](#)
- [编辑规则组命名空间的标签](#)
- [从规则组命名空间中删除标签](#)

向规则组命名空间添加标签

为适用于 Prometheus 的 Amazon 托管服务规则组命名空间添加标签可以帮助您识别和组织 AWS 资源并管理对资源的访问权限。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 向适用于 Prometheus 的亚马逊托管服务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 \(控制台 \)](#)
- [向规则组命名空间添加标签 \(AWS CLI \)](#)

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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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. 在键中，输入标签的名称。您可以在值中添加可选的标签值。
9. (可选) 要添加其他标签，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)

按照以下步骤使用向适用于 Prometheus 的亚马逊托管服务规则组命名空间添加标签。AWS CLI 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，我们假设您已经安装了最新版本 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。例如，要为名为 My-Workspace 的 Amazon Prometheus 托管服务命名空间添加两个标签，一个标签键 *Status* 名为的标签值为，一个标签键名为 *Secret* 的标签值为：*TeamMy-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

标签可以帮助您识别和整理 AWS 资源并管理对资源的访问权限。有关标记策略的更多信息，请参阅为资源[添加标签](#)。AWS

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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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 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。您也可以更改键的名称，这相当于移除当前的标签并使用新名称和与另一个键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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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. 要添加其他标签，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。移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. 打开适用于 Prometheus 的亚马逊托管服务控制台，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。
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)

按照以下步骤使用从规则组命名空间中移除标签。AWS CLI 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，要在名为 My-Workspace 的工作空间上删除带有标签键的标签，请执行以下操作：*Status*

```
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。

服务配额

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 Prometheus CloudWatch s 的资源使用情况](#)。使用亚马逊 CloudWatch 使用指标警报功能，您可以监控 Prometheus 的资源和使用情况，以防止出现限制错误。

随着项目和工作区增长，您应监控或请求增加的最常见配额是：每个工作区的活跃系列数和每个工作区的摄取率。

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[请求增加配额](#)来请求增加配额。

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[活跃系列默认配额](#)。每个工作区的摄取率配额决定了您将数据摄入到工作区的速度。有关更多信息，请参阅[摄取节流](#)。

Note

除非另有说明，否则这些限额适用于每个工作区。每个工作区的活跃系列的最大值为十亿。

Name	默认值	可调整	说明
每个工作区具有元数据的活动指标	每个受支持的区域：2 万个	否	每个工作区具有元数据的独特活动指标数量。注：如果达到限制，则会记录指标样本，但会丢弃超过限制的元数据。
每个工作区的活跃系列数	每个受支持的区域：5000 万个	是	每个工作区的独特活跃序列数（最多可达 10 亿个）。如果在过去 2 小时内报告了样本，则该序列

Name	默认值	可调整	说明
			处于活动状态。2 M 到 50 M 的容量会自动根据最近 30 分钟的使用情况进行调整。
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的警报聚合组的最大大小。group_by 的每个标签值组合都会创建一个聚合组。
警报管理器定义文件大小	每个受支持的区域：1000000 个	否	警报管理器定义文件的最大大小（以字节为单位）。
警报管理器中的警报有效载荷大小	每个受支持的区域：2000 万个	否	每个工作区所有警报管理器警报的最大警报负载大小，以字节为单位。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：1,000 个	是	每个工作区中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 Tracker 将针对每个工作区摄取样本所跟踪的最大集群数。
每个工作区的摄取率	每个支持的区域：1,666,666	是	每个工作区每秒的指标样本摄取率。该限制会自动调整为每个工作空间 1/30 的活动系列限制，最高为 1,666,666。
警报管理器定义文件中的抑制规则	每个受支持的区域：100 个	是	警报管理器定义文件中最大的抑制规则数。

Name	默认值	可调整	说明
标签大小	每个受支持的区域：7 个	否	一个序列接受的所有标签和标签值的最大组合大小（以 KB 为单位）。
LabelSet 每个工作空间的限制	每个受支持的区域：100 个	<u>是</u>	每个工作区可以创建的标签集最大数量限制。
每个指标系列的标签数	每个支持的区域：150 个	<u>是</u>	每个指标序列的标签数。
元数据长度	每个受支持的区域：1 个	否	指标元数据接受的最大长度（以 KB 为单位）。元数据指的是指标名称、类型、单位和帮助文本。
每个指标的元数据	每个受支持的区域：10 个	否	每个指标的元数据数 注：如果达到限制，则会记录指标样本，但会丢弃超过限制的元数据。
警报管理器路由树中的节点	每个受支持的区域：100 个	<u>是</u>	警报管理器路由树中的最大节点数。
每个区域的 API 操作数（以每秒事务数为单位）	每个受支持的区域：10 个	<u>是</u>	所有 Amazon Managed Service for Prometheus API（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）每区域每秒的最大 API 操作数。

Name	默认值	可调整	说明
每个工作空间的 GetSeries、GetLabels 和 GetMetricMetadata API 操作数 (以每秒事务数为单位)	每个受支持的区域：10 个	否	每个工作区每秒 GetLabels 和每秒 GetMetricMetadata Prometheus-compatible API 操作的最大数量。GetSeries
每个工作空间的 QueryMetrics API 操作数 (以每秒事务数为单位)	每个受支持的区域：300 个	否	每个工作空间每秒的最大 QueryMetrics Prometheus-compatible API 操作数。
每个工作空间的 RemoteWrite API 操作数 (以每秒事务数为单位)	每个受支持的区域：3000 个	否	每个工作空间每秒的最大 RemoteWrite Prometheus-compatible API 操作数。
每个工作空间的其他 Prometheus-compatible API 操作数 (以每秒事务数为单位)	每个受支持的区域：100 个	否	所有其他 API (包括 ListAlerts、ListRules等) 每个工作区每秒的最大 Prometheus-compatible API 操作数。
每个工作空间的失序摄取率	每个支持的区域：83,333	<u>是</u>	每个工作空间每秒钟的失序样本摄取速率。除非被覆盖，否则该限制会自动调整为每个工作空间摄取速率限制的 5%。
每个工作空间的失序时间窗口	每个受支持的区域：600 个	<u>是</u>	每个工作区乱序样本的最大时间窗口，以秒为单位。

Name	默认值	可调整	说明
即时查询的查询字节数	每个受支持的区域：5 个	否	单个即时查询能扫描的最大字节数（以 GB 为单位）。
范围查询的查询字节数	每个受支持的区域：5 个	否	单个范围查询中每 24 小时能扫描的最大字节数（以 GB 为单位）。
查询样本	每个受支持的区域：5000 万个	否	在单个范围查询或单个即时查询中，每 24 小时间隔可以扫描的最大样本数。
已提取的查询序列	每个受支持的区域：1,200 万个	否	在单个范围查询或单个即时查询中，每 24 小时间隔可以扫描的最大序列数。
查询时间范围（以天为单位）	每个支持的区域：95	否	QueryMetrics、GetSeries和 GetLabels API 的最大时间范围。
请求大小	每个受支持的区域：1 个	否	摄取或查询的最大请求大小（以 MB 为单位）。
规则评估间隔	每个受支持的区域：30 个	<u>是</u>	每个工作区中一个规则组的最小规则评估间隔（以秒为单位）。
规则组命名空间定义文件大小	每个受支持的区域：1000000 个	否	一个规则组命名空间定义文件的最大大小（以字节为单位）。
每个工作区的规则数	每个受支持的区域：2,000 个	<u>是</u>	每个工作区的最大规则数。

Name	默认值	可调整	说明
每个工作区的静默数	每个受支持的区域：1,000 个	是	每个工作区的最大静默数，包括已过期、活动和待处理的静默。
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量。

活跃系列默认配额

Amazon Managed Service for Prometheus 工作区会自动根据您的摄取使用量进行调整。随着使用量增加，该服务将自动增加时间序列容量，直至达到默认配额。

Amazon Managed Service for Prometheus 工作区会根据使用量自动扩展，方式有两种：

1. 当 30 分钟平均使用量低于 500 万个系列时，容量将翻一番（例如，使用量为 350 万的工作区获得 700 万的容量）。
2. 当使用量超过 500 万个系列时，工作区会增加 1000 万的缓冲区（例如，使用量为 2500 万的工作区将获得 3500 万容量）。

Amazon Managed Service for Prometheus 会随着摄取量增加自动分配更多容量，直至您的配额。这有助于确保您的工作负载不会持续受到节流。但是，如果与过去 30 分钟内计算出的先前基准值相比增加了一倍或超过 1000 万，则可能会出现节流。为避免节流，Amazon Managed Service for Prometheus 建议在增加到超过之前的基准值时逐渐增加摄入量。

Note

活跃时间序列的最小容量为 200 万，并且当序列少于 200 万时没有节流。
要超出其默认配额，您可以请求[增加配额](#)。

扩展到超出默认配额

当您请求将配额增加到超过默认活跃系列配额时，Amazon Managed Service for Prometheus 会相应地调整您的工作区容量。如果您没有充分利用增加的容量，该服务将随着时间推移收回未使用的部分。随着使用量增长，工作区将再次自动纵向扩展。

但是，如果活跃时间序列比过去 2 小时计算的上一个基准值增加了一倍以上或超过 5000 万个，则可能会发生节流。例如：

- 如果配额为 1 亿，而基准值为 3000 万，则可以在 2 小时内纵向扩展到 6000 万，而不会发生节流。
- 如果配额为 1 亿，而基准值为 5000 万，则可以在 2 小时内纵向扩展到整 1 亿，而不会发生节流。

摄取节流

Amazon Managed Service for Prometheus 会根据您当前的限制对每个工作区的摄取量进行节流。这有助于保持工作区的性能。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 CloudWatch 监控您的摄取量，也可以创建警报，在接近限制限制时向您发出警报。有关更多信息，请参阅 [使用 CloudWatch 指标监控亚马逊托管服务的 Prometheus 资源](#)。

Amazon Managed Service for Prometheus 使用 [令牌存储桶算法](#) 来实现摄取节流。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。存储桶中的令牌数表示您在任何给定秒钟的摄取上限。

每摄取一个数据样本，就会从存储桶中移除一个令牌。如果您的存储桶大小（每个工作区的摄取率）为 1000000，那么您的工作区可以在一秒钟内摄取一百万个数据样本。如果要摄取的样本超过一百万个，就会被节流，不再摄取任何记录。其他数据样本将被丢弃。

存储桶会以设定的速率自动重填。如果存储桶的容量低于其最大容量，则每秒都会向其添加一定数量的令牌，直到其达到最大容量。如果重填令牌到达时存储桶已满，令牌就会被丢弃。存储桶中的令牌数量不能超过其最大数量。样本摄取的重填速率由每个工作区的摄取速率限制来设置。如果将每个工作区的摄取速率设置为 170,000，则存储桶的重填速率为每秒 170,000 个令牌。

如果您的工作区在一秒钟内摄取 1,000,000 个数据样本，您的存储桶就会立即缩减为零令牌。然后，存储桶中每秒重填 170,000 个令牌，直至达到其 1,000,000 个令牌的最大容量。如果不再进行摄取，则之前空的存储桶将在 6 秒钟内恢复到最大容量。

Note

摄取以批处理请求的方式进行。如果您有 100 个可用令牌，却发送了包含 101 个样本的请求，那么整个请求都会被拒绝。Amazon Managed Service for Prometheus 不部分接受请求。如果您正在编写收集器，则可以管理重试（使用较小的批次或在一段时间后重试）。

在工作区摄取更多数据样本之前，您无需等待存储桶填满。您可以在令牌被添加到存储桶时使用这些令牌。如果您立即使用重填令牌，存储桶就不会达到最大容量。例如，如果您耗尽了存储桶，您可以继续每秒摄取 170,000 个数据样本。只有当每秒摄取的数据样本少于 170,000 个时，存储桶才能重填到最大容量。

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。

Amazon Managed Service for Prometheus API 参考

Amazon Managed Service for Prometheus 提供两种类型的 API :

1. Amazon Managed Service for Prometheus API – 借助这些 API , 您可以创建和管理 Amazon Managed Service for Prometheus 工作区 , 包括针对工作区、抓取程序、警报管理器定义、规则组命名空间和日志记录的操作。您可以使用适用于各种编程语言的 AWS SDK 与这些 API 交互。
2. 与 Prometheus 兼容的 API – Amazon Managed Service for Prometheus 支持与 Prometheus 兼容的 HTTP API。通过这些 API , 可以构建自定义应用程序、自动执行 workflows、与其他服务或工具集成 , 以及使用 Prometheus 查询语言 (PromQL) 查询监控数据并与之交互。

本部分列出了 Amazon Managed Service for Prometheus 支持的 API 操作和数据结构。

有关系列、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](#)
- [兼容普罗米修斯 APIs](#)

Amazon Managed Service for Prometheus API

Amazon Managed Service for Prometheus 提供创建和维护 Amazon Managed Service for Prometheus 工作区的相关 API 操作。其中包括用于工作区、抓取程序、警报管理器定义、规则组命名空间和日志记录的 API。

有关 Amazon Managed Service for Prometheus API 的详细信息 , 请参阅 [Amazon Managed Service for Prometheus API 参考](#)。

将 Amazon Managed Service for Prometheus 与 AWS SDK 配合使用

AWS 软件开发工具包 (SDK) 适用于许多常用编程语言。每个 SDK 都提供 API、代码示例和文档 , 使开发人员能够更轻松地以其首选语言构建 AWS 应用程序。有关按语言划分的 SDK 和工具列表 , 请参阅 AWS 开发人员中心的 [用于在 AWS 上进行构建的工具](#)。

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 AWS 开发工具包以及任何其他开发工具包，并使开发工具包保持最新。AWS 开发工具包为您提供最新的特性和功能以及安全更新。

兼容普罗米修斯 APIs

适用于 Prometheus 的亚马逊托管服务支持以下与 Prometheus 兼容的服务。APIs

有关使用兼容 Prometheus 的更多信息，请参阅。APIs [使用 Prometheus-compatible API 进行查询](#)

主题

- [CreateAlertManagerAlerts](#)
- [DeleteAlertManagerSilence](#)
- [GetAlertManagerStatus](#)
- [GetAlertManagerSilence](#)
- [GetLabels](#)
- [GetMetricMetadata](#)
- [GetSeries](#)
- [ListAlerts](#)
- [ListAlertManagerAlerts](#)
- [ListAlertManagerAlertGroups](#)
- [ListAlertManagerReceivers](#)
- [ListAlertManagerSilences](#)
- [ListRules](#)
- [PutAlertManagerSilences](#)
- [QueryMetrics](#)
- [RemoteWrite](#)

CreateAlertManagerAlerts

CreateAlertManagerAlerts 操作在工作区中创建警报。

有效的 HTTP 动词：

POST

有效 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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 URIs：

`/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。

`max_samples_processed_warning_threshold=<integer>` (可选) 为已处理的查询样本 (QSP) 设置警告阈值。当查询达到此阈值时，将在 API 响应中返回一条警告消息。

`max_samples_processed_error_threshold=<integer>>` (可选) 为已处理的查询样本 (QSP) 设置错误阈值。超过此阈值的查询将被拒绝并显示错误，并且不收费。用于防止过高的查询成本。

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 URIs：

`/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 [https://github.com/prometheus/prometheus/blob/1c624c58ca934f618be737b4995e22051f5724c1/prompb/remote.pb.go](https://github.com/prometheus/prometheus/blob/1c624c58ca934f618be737b4995e22051f5724c1/prompb/remote.pb.go#L64) #L64 上的协议缓冲定义。

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
已推出对以下内容的支持 PagerDuty	适用于 Prometheus 的 Amazon 托管服务增加了 PagerDuty 对集成的支持，可实现自动事件响应工作流程，并确保关键警报在正确的时间到达正确的团队成员。有关更多信息，请参阅 PagerDuty 用作警报接收器 。	2025 年 8 月 29 日
添加了基于资源的策略支持	以下 API 操作现已可用： <ul style="list-style-type: none">• DeleteResourcePolicy• DescribeResourcePolicy• PutResourcePolicy	2025 年 8 月 15 日
更新托 AmazonPrometheusConsoleFullAccess 管 IAM 策略。	该 AmazonPrometheusConsoleFullAccess 策略已更新。aps:CreateQueryLoggingConfiguration、aps:UpdateQueryLoggingConfiguration、aps:DeleteQueryLoggingConfiguration、aps:DescribeQueryLoggingConfiguration 权限已添加到该策略中。	2025 年 5 月 5 日

[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](#)

Amazon Managed Service for Prometheus 增加了对从 Amazon Managed Service for Prometheus 控制台编辑[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](#)

Amazon Managed Service for Prometheus 增加了对 Amazon EKS 访问条目的支持，以简化[AWS 托管式收集器](#)的设置。[AmazonPrometheusScrapingServiceRolePolicy](#)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](#)

Prometheus 的亚马逊托管服务现已在其自己的参考文献中提供，即 AWS APIs Prometheus 的[亚马逊托管服务 API 参考](#)。《适用于 Prometheus 的[亚马逊托管服务用户指南](#)》中 APIs [继续记录与 Prometheus 兼容的内容](#)。

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](#)。

2023 年 12 月 21 日

[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， AmazonPrometheusSc raperServiceLinkedRolePolicy	添加了新的托管策略 , AmazonPrometheusSc raperServiceLinkedRolePolic y 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 AmazonPro metheusConsoleFullAccess	为 AmazonPrometheusCo nsoleFullAccess 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 Obser vability 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

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。