

Prometheus 监控服务

操作指南

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

操作指南

实例

- 创建实例
- 搜索实例
- 修改实例名称
- 销毁实例
- 重建实例
- 修改存储时长
- 查看实例基本信息

Agent 管理

- Agent 概述
- 新建 Agent
- 安装 Agent
- 新建抓取任务

集成容器服务

集成中心

预聚合

- 预聚合概述
- 规则管理
- 默认预聚合规则列表

告警策略

- 告警策略概述
- 告警规则说明
- 新建告警策略
- 暂停告警策略
- 策略类型说明
- 通知模板

标签管理

- 标签示例
- 使用标签
- 编辑标签
- 使用限制

访问控制

- 访问控制概述
- 策略设置

策略授予

相关角色权限说明

Grafana

API 使用指南

API 概览

数据写入

监控数据查询

容器服务指标

按量付费免费指标

容器常用指标推荐

相关资源使用及计费说明

操作指南

实例

创建实例

最近更新时间：2024-01-29 16:01:49

本文以自定义配置方式为例，指导您如何创建 Prometheus 实例。

准备工作

在创建 Prometheus 实例前，您需要完成以下工作：

[注册腾讯云账号](#)，并完成 [实名认证](#)。

需要在目标地域 [创建一个私有网络](#)，并且在私有网络下的目标可用区 [创建一个子网](#)。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 单击**新建**，根据页面提示，配置以下信息：

类别	必选/ 可选	配置说明
可用区域	必选	根据您云产品所在区域选择。每个地区价格可能会不一样，以购买页实时的价格为主。处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择；例如，广州地域的服务无法通过内网上报数据到上海地域的Prometheus。
可用区	必选	根据您的实际需求选择。
网络	必选	表示在腾讯云上构建的逻辑隔离的网络空间，一个私有网络由至少一个子网组成。系统会为您在每个地域提供的默认私有网络和子网。如现有的私有网络/子网不符合您的要求，详情请参见 新建私有网络 和 新建子网 进行创建。
产品规格	必选	不同的产品规格价格有差异，详情请参见 产品定价 。
数据存储时间	必选	不同的存储时长价格有差异，详情请参见 产品定价 。
实例名称	必选	用户自定义 prometheus 实例名称。
Grafana/Grafana 密码	必选	默认开启 Grafana，实例创建成功后系统会生成一个公网可访问的域名。该 Grafana 的默认账号为：admin，登录密码由您自定义。

购买时长	必选	提供多种时长购买。
------	----	-----------

3. 配置完后，单击**立即购买**即可。

Region and Network Config

Region

South China

Guangzhou
Shenzhen
Shenzhen Finance

Tencent Cloud services in different regions cannot communicate with each other over the private network. For example, the service in Guangzhou region cannot report data to TMP in Shang it after purchasing the instance.

AZ

Guangzhou Zone 1
Guangzhou Zone 2
Guangzhou Zone 3
Guangzhou Zone 4
Guangzhou Zone 5
Guangzhou Zone 6
Guangzhou Zone

Network

vpc-rdalicw7 | intl_test | 10.0.0.0/16
subnet-9apu3jks | intl_test_1 | 10.0.0.0/24
Available

If the existing VPC/subnet does not meet your requirement, you can go to the console to [create a VPC](#) or [Create Subnet](#). Only services in the "intl_test VPC" can report monitoring data. Please select the network with caution as you cannot change it after purchasing the instance.

Basic Instance Config

Data Retention Period

15 days
30 days
45 days

Instance Name

Grafana

grafana-test-ai0efv5c | test0802
Available

If the existing Grafana instance does not meet your requirement, you can [create one](#) in the console.

Tag (optional)

kkk
del
删除

+ Add

If the existing tag/tag value does not meet your requirement, you can [create one](#) in the console.

Terms of Agreement I've read and agree to [Tencent Cloud Terms of Service](#), [Tencent Cloud Prometheus Service Level Agreement](#), [Billing Overview](#), and [Payment Overdue](#)

搜索实例

最近更新时间：2024-01-29 16:01:49

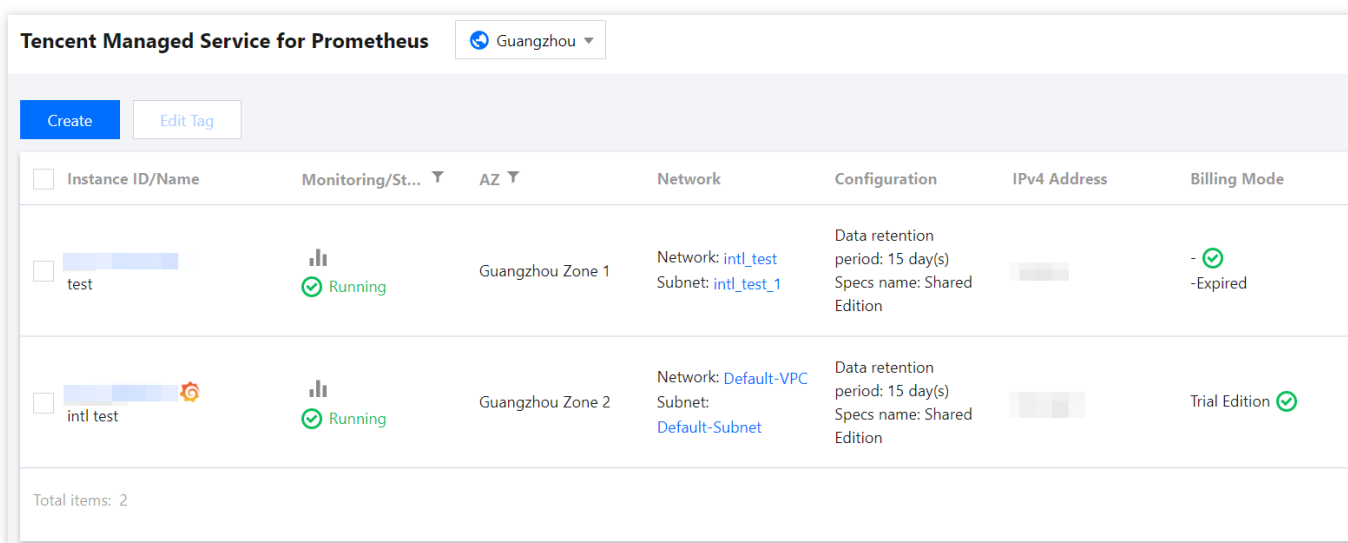
默认情况下，Prometheus 监控服务控制台展示的是当前地域下 Prometheus 实例。为了帮助用户快速搜索出当前地域下的 Prometheus 实例，腾讯云提供搜索功能，目前可通过实例 ID、实例名称、实例状态、可用区、IPv4 地址、标签等资源属性维度进行过滤。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在搜索框中，根据实际需求，输入需搜索的内容，单击



进行搜索。



3. 支持用户根据不同的维度来过滤实例列表，目前支持如果几种维度：

过滤项	说明
实例 ID	支持多个实例 ID 过滤，每个实例 ID 仅支持完全匹配过滤，支持直接输入实例 ID 来快速过滤。
实例名称	不支持多个名称过滤，支持模糊匹配过滤。
实例状态	支持用户多个状态选择过滤，同时也支持在列表头进行快速过滤。
可用区	支持用户多个可用区过滤，切换地域之后显示该地域下的 Prometheus 对应的可用区，同时也支持

	在列表头进行快速过滤。
IPv4 地址	支持多个 IPv4 过滤，每个 IPv4 仅支持完全匹配过滤。
标签	支持多个标签组合来过滤实例，同时也支持直接单击实例列表中对应的标签值来进行过滤。

修改实例名称

最近更新时间：2024-01-29 16:01:49


为了方便用户在 Prometheus 监控服务控制台上进行 Prometheus 实例管理，可快速辨识出 Prometheus 实例的名字，腾讯云支持给每台实例命名，并且可随时更改，立即生效。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择需要被修改实例名称的 Prometheus 实例，单击右侧的[更多 > 实例配置 > 修改名称](#)。
3. 在弹出的“修改名称”窗口中，输入新实例名称，单击**确定**即可。

Modify Instance Name

You have selected this instance:

Instance ID/...	Status	AZ	Network	Configu
	✔ Running	Guangzhou Zone 1	Network: intl_test Subnet: intl_test_1	Data rete period: 1 day(s)

Instance Name

销毁实例

最近更新时间：2024-01-29 16:01:49

当您不再使用某个实例时，可以对实例进行销毁，被销毁的实例会处于停用状态。对于停用状态的实例，您可以根据不同场景和需求进行重建实例。

相关影响

当实例进入停用状态时，实例数据相关影响如下：

IP：对应的 IP 地址还会保留，但不对外提供正常的服务。

Grafana：无法再通过对应的域名访问 Grafana。

数据：对应的实例数据还会保留相应天数，具体天数以控制台销毁过程中的确认信息中为准。


操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择需要被销毁的 Prometheus 实例，单击右侧的**更多 > 实例状态 > 销毁**。
3. 由于销毁操作属于高危操作，在弹出的“销毁”窗口中，完成销毁操作步骤，单击**确定**即可。

Terminate

- 1 Termination Details >
- 2 Confirm >
- 3 Notes

You have selected this instance:

Instance ID/...	Status	AZ	Network	Configuration
	✔ Running	Guangzhou Zone 1	Network: intl_test Subnet: intl_test_1	Data retention period: 15 day(s)

- i • Once terminated, the instance will be in the shutdown status for seven days, during which its data will be retained.
- After the resources are terminated, the five-day unconditional refund (for one instance) will be returned to your Cloud account. The normal refund will be returned to your account based on the proportion of the cash and gift cards paid for the purchase.
- The discount or voucher you used when purchasing the instance is not refundable.

[Next](#) [Cancel](#)

重建实例

最近更新时间：2024-01-29 16:01:49

如果您需要对停用或者异常状态的实例进行重建，您可以在控制台上对相应的实例进行重建操作。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择需要被销毁的 Prometheus 实例，单击右侧的**更多 > 实例状态 > 重建**。
3. 在弹出的“重建操作须知”窗口中，单击**确定**即可。

Instance Renewal

You have selected this instance:

Instance ID/Name	Status	AZ	Network	Configuration	Bill
	Running	Guangzhou Zone 1	Network: intl_test Subnet: intl_test_1	Data retention period: 15 day(s)	-

Validity Period

Auto-Renewal Auto-renew the device every month when my account has sufficient balance

Fees Querying configuration fees...

说明：

如果当前实例状态为运行中，对其进行重建操作，会导致在重建过程中服务中断的情况，请您确认操作的风险。

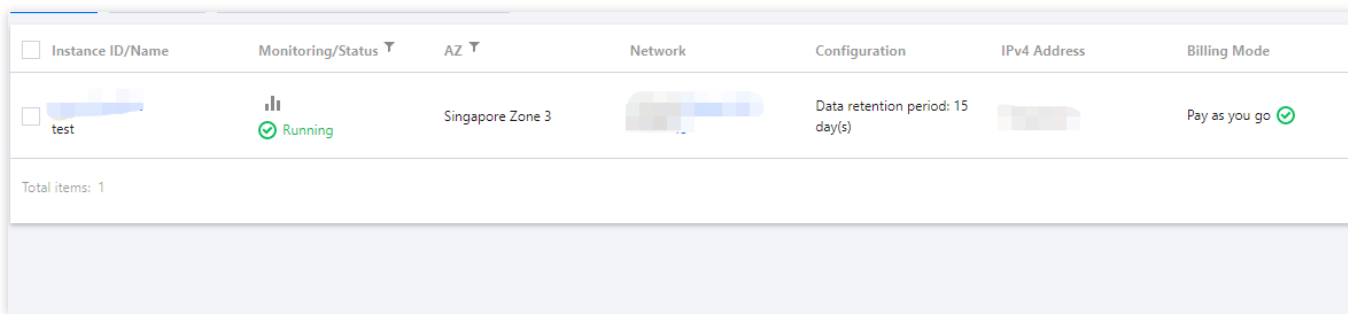
修改存储时长





最近更新时间：2024-01-29 16:01:49

Prometheus 支持您在创建实例后修改数据存储时长。存储时长越长，实例单价越高，您可以参见 [按量计费说明](#) 进行合理调整。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，找到需要被修改的 Prometheus 实例，单击右侧的 **更多 > 实例配置 > 修改存储时长**。



Instance ID/Name	Monitoring/Status	AZ	Network	Configuration	IPv4 Address	Billing Mode
<input type="checkbox"/> test	 Running	Singapore Zone 3		Data retention period: 15 day(s)		Pay as you go 

Total items: 1

3. 在弹框中选择需要修改的存储时长，单击 **确定** 即可。

Modify Storage Period

You have selected this instance:

Instance ID/...	Status	AZ	Network	Configuration	Billing
test	Running	Singapore Zone 3	Network:Default-VPC Subnet:rs	Data retention period: 15 day(s)	Pay as

Data Storage Period

Note: The storage period impacts the unit price of pay-as-you-go instances. For details, see [here](#).

OK Cancel

说明：

修改成功后，第二天0点开始，新采集的数据将按照新的存储时长存储和新的计费单价进行计费。历史数据的存储时长仍按照修改前的存储时长存储。

查看实例基本信息






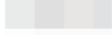

最近更新时间：2024-01-29 16:01:49

为了方便用户查看 Prometheus 实例基本信息，您可以在实例列表页面选择对应的实例，进入实例管理页查看实例的基本信息。



操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择需要查看的 Prometheus 实例，单击**实例 ID** 或者右侧的**管理**。

Basic Info

Name	modify 
Instance ID	 
Status	 Running
Region	Guangzhou
AZ	Guangzhou Zone 1
Network	intl_test
Subnet	intl_test_1
Tag	
IPv4 Address	 
Grafana Status	<input type="checkbox"/> Disabled
Billing Mode	- -Expired
Creation Time	2021/11/24 16:56:45

Service Address

Token	***** 
Remote Write Address	 
HTTP API	 
Pushgateway Address	 

3. 在基本信息页支持如下操作：

修改实例名称。

编辑实例对应的标签。

修改 Grafana Admin 密码。

升级 Grafana 预设 Dashboard。

Agent 管理

Agent 概述

最近更新时间：2024-01-29 16:01:49

Agent 管理用于实现监控对象部署在 CVM 或者 自建 IDC 时，自定义上报监控数据到 Prometheus 监控服务。您成功上报数据后，可以通过集成的开源可视化的 Grafana 查看监控大屏，还可以对该监控对象设置告警规则，实时监控其状态，在状态异常时第一时间给您发送告警通知。

操作介绍

1. [新建 Agent](#)
2. [安装 Agent（上报监控数据到 Prometheus 监控服务）](#)
3. [新建抓取任务（定义 Agent 抓取任务规则）](#)

新建 Agent

最近更新时间：2024-01-29 16:01:49

本文将为您介绍如何新建 Agent。

前提条件

已创建 Prometheus 实例。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例，单击左侧菜单栏的 **Agent 管理**。
3. 在 Agent 管理页单击**新建**。
4. 在弹框中填写 Agent 名称，填写完后单击**保存**即可。

Create Agent

×

Agent Name ✓

安装 Agent

最近更新时间：2024-01-29 16:01:49

本文将为您介绍如何安装 Agent。

前提条件

已创建 Prometheus 实例。

已创建 Agent。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例，单击左侧菜单栏的 **Agent 管理**。
3. 单击 Agent 对应的 ID，进入 Agent 安装指南页，复制 Agent 安装命令于您的云服务器或自建 IDC，修改命令中的 `<secret_id>` 和 `<secret_key>` 后即可执行。

Installation Guide Scrape Task

i • Note: the agent must be deployed on a server interconnected with the current instance.

1. Install agent:

- You need to specify the SecretId/SecretKey when installing the agent. [Obtain SecretId/SecretKey](#)

```
wget https://rig-...cos.ap-guangzhou.myqcloud.com/prometheus-agent/agent_install -O agent_install && chmod +x ...ap-guangzhou <secret_id> <secret_key>
```

执行成功示例如下：

```

100%[=====>] 87,502,263 16.9MB/s
2021-01-28 21:29:05 (17.7 MB/s) - '/usr/bin/prometheus' saved [8750
]
■ prometheus.service - The Tencent Cloud Prometheus Agent
  Loaded: loaded (/usr/lib/systemd/system/prometheus.service; disa
  preset: disabled)
  Active: active (running) since Thu 2021-01-28 21:29:05 CST; 3s a
  Main PID: [REDACTED] (prometheus)
  CGroup: /system.slice/prometheus.service
          └─ [REDACTED] /usr/bin/prometheus --agent.enable-sidecar --ter

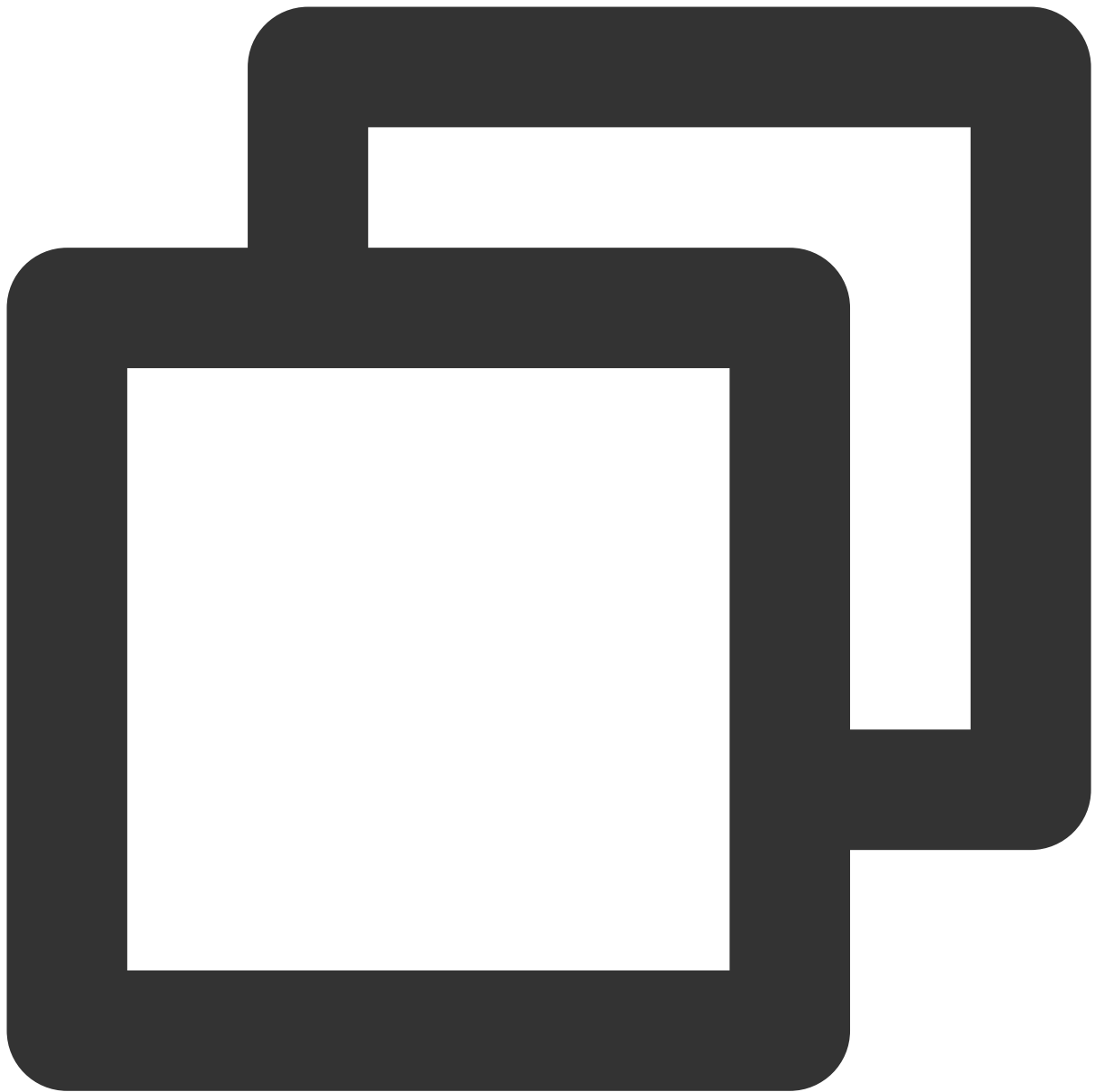
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Jan 28 21:29:05 VM-0-10-centos prometheus[ [REDACTED] ]: level=info ts=202
Hint: Some lines were ellipsized, use -l to show in full.
    
```

4. 返回 Agent 管理页面，如果 Agent 正常运行，可以看到 Agent 上报的版本、IP 地址和心跳时间。

其它命令

重启 Agent

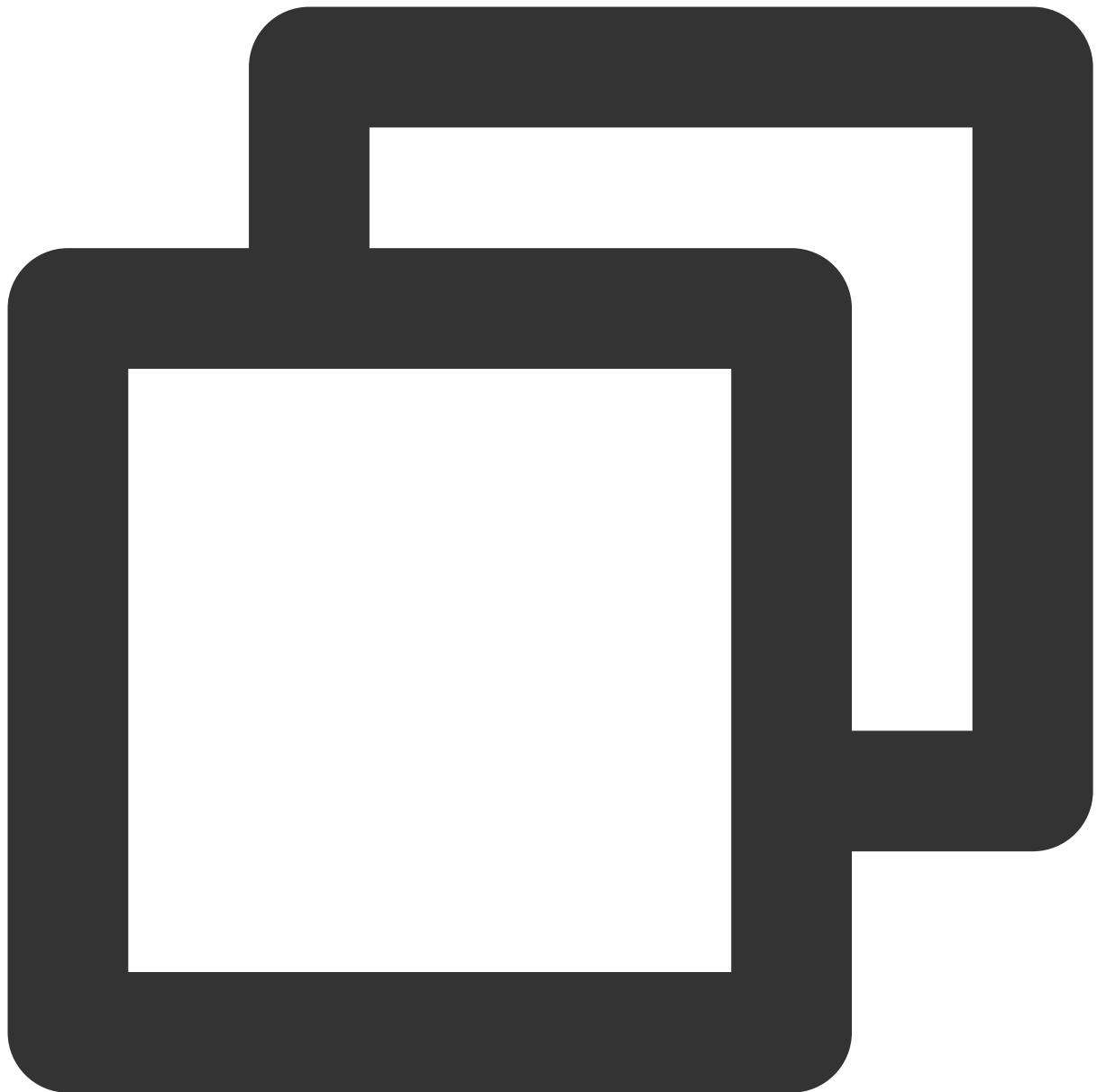
执行如下命令：



```
systemctl restart prometheus
```

停止 Agent

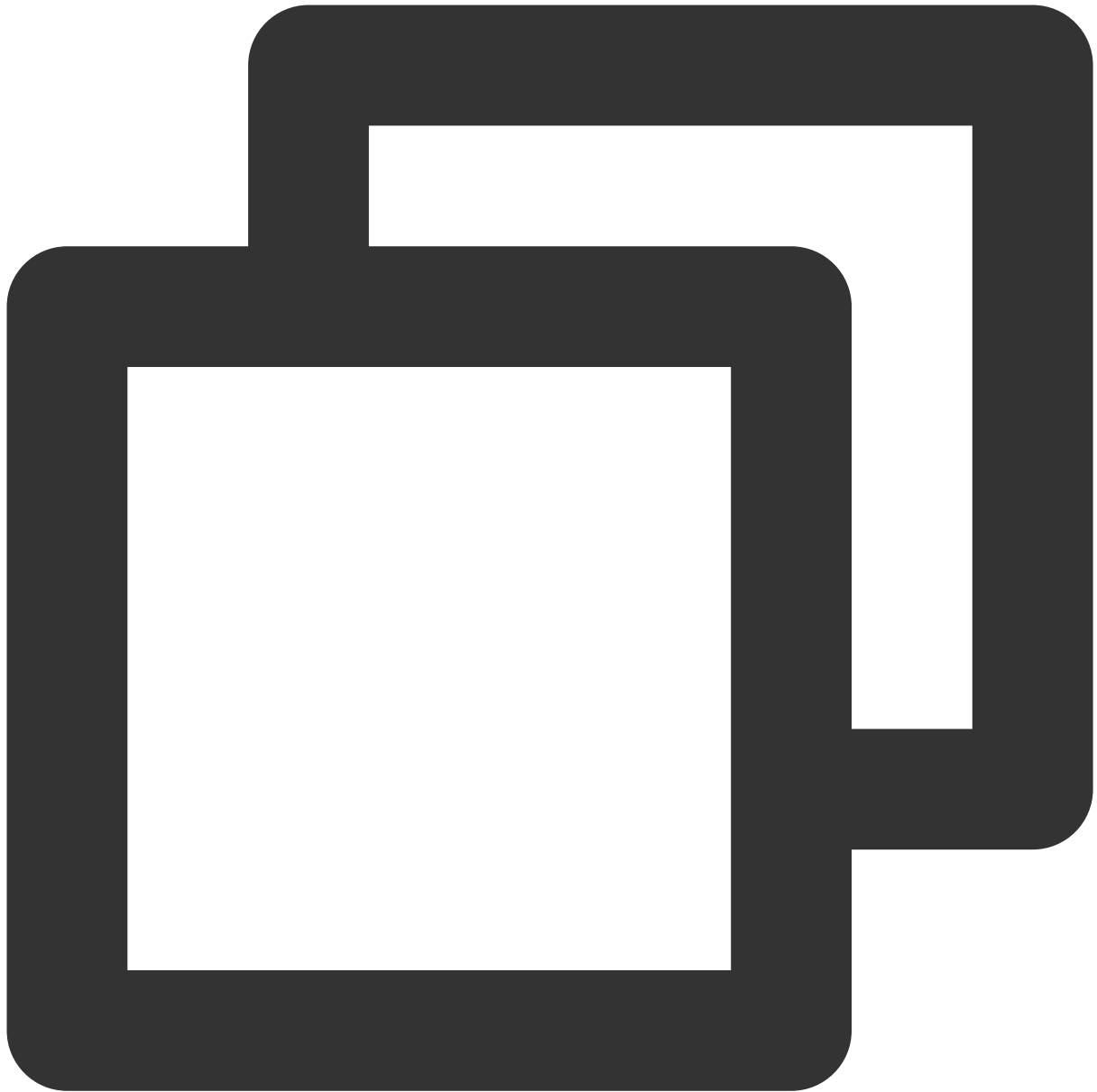
执行如下命令：



```
systemctl stop prometheus
```

检查 Agent 状态

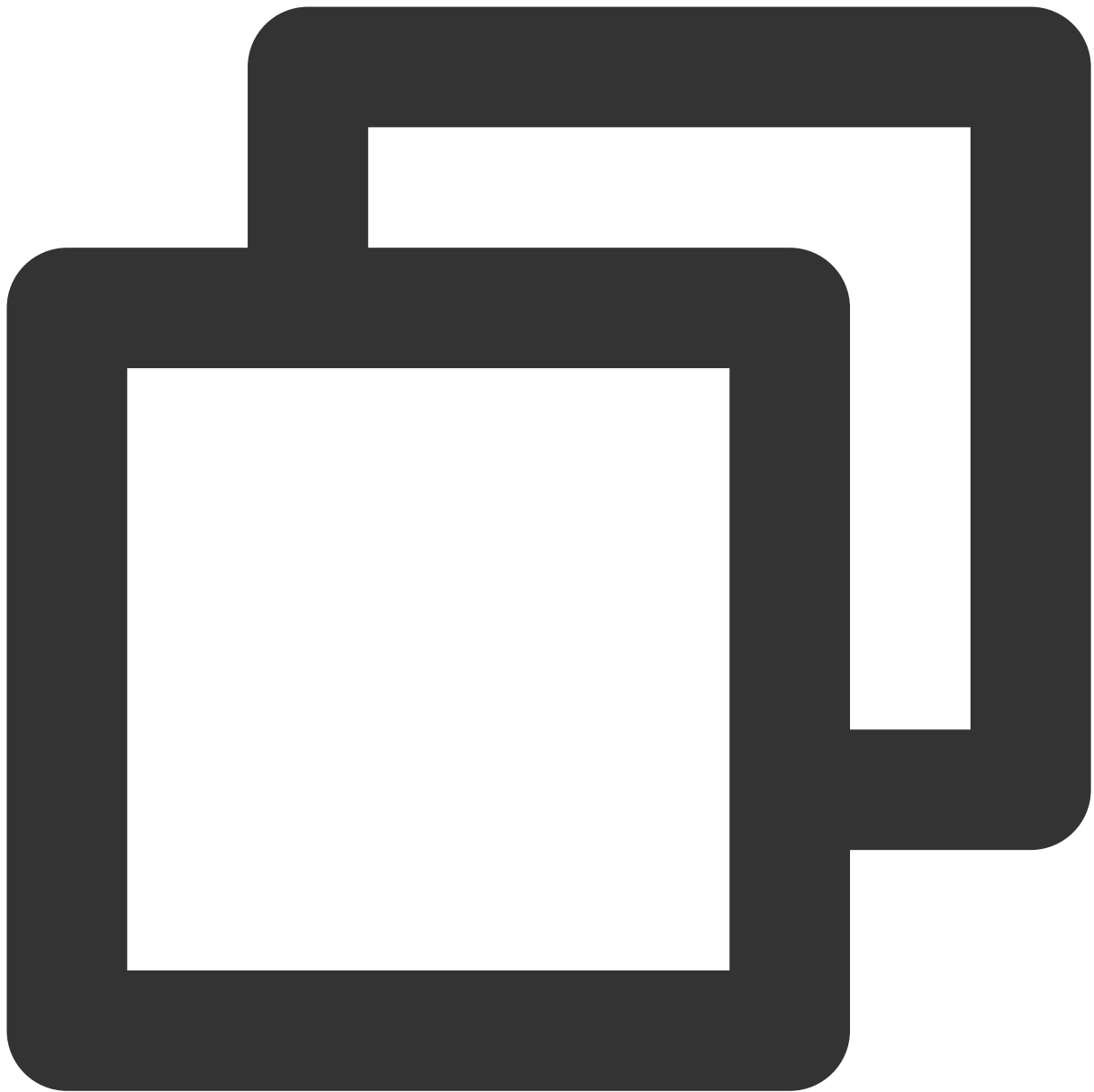
执行如下命令：



```
systemctl status prometheus
```

查看 Agent 日志

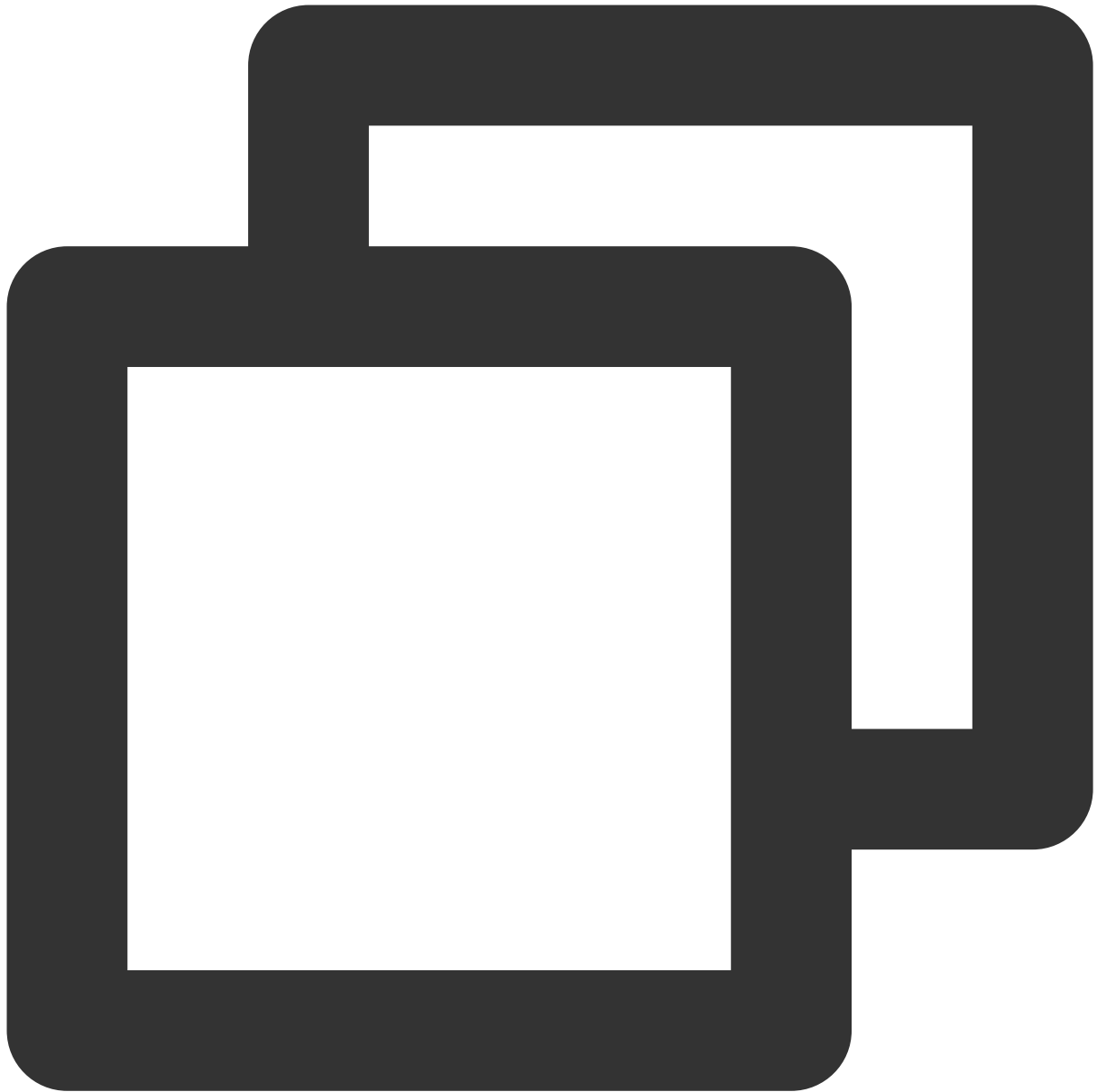
执行如下命令：



```
journalctl -f --unit=prometheus
```

卸载 Agent

执行如下命令：



```
systemctl stop prometheus && rm -rf /usr/lib/systemd/system/prometheus.service
```

新建抓取任务

最近更新时间：2024-01-29 16:01:49

本文将为您介绍如何新建抓取任务。

前提条件

已创建 Prometheus 实例。

已创建 Agent 并安装 Agent。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例，单击左侧菜单栏的 **Agent 管理**。
3. 单击 Agent 对应的 ID，单击**抓取任务**。
4. 进入抓取任务页，单击**新建**。
5. 在新建抓取任务页，根据提示填写 Agent 抓取任务规则，任务配置可以参考[抓取配置说明](#)。

Create Scrape Task

 • For scrape configuration details, click [here](#) 

```
1 ScrapeJob(en-US)
```

Save

Cancel

6. 配置完成后，单击**保存**即可。

集成容器服务

最近更新时间：2024-01-29 16:01:49

集成容器服务后即可对腾讯云容器服务业务场景进行监控。本文将为您介绍如何集成容器服务。

腾讯云容器服务（Tencent Kubernetes Engine, TKE）是基于原生 Kubernetes 提供以容器为核心的解决方案，解决用户开发、测试及运维过程的环境问题、帮助用户降低成本，提高效率。而 Kubernetes 是一款由 Google 开发的开源的容器编排工具，在 Google 已使用超过15年。作为容器领域事实的标准，Kubernetes 可以极大地简化应用的管理和部署复杂度。通过与容器服务集成，可以极大简化用户通过 Prometheus 来监控 Kubernetes 状态及其运行在上面的服务。

说明：

为保证正常运行，存量实例在编辑采集配置和新关联集群时会自动更新组件版本，更新过程中可能会造成已关联的集群数据断点。

操作步骤

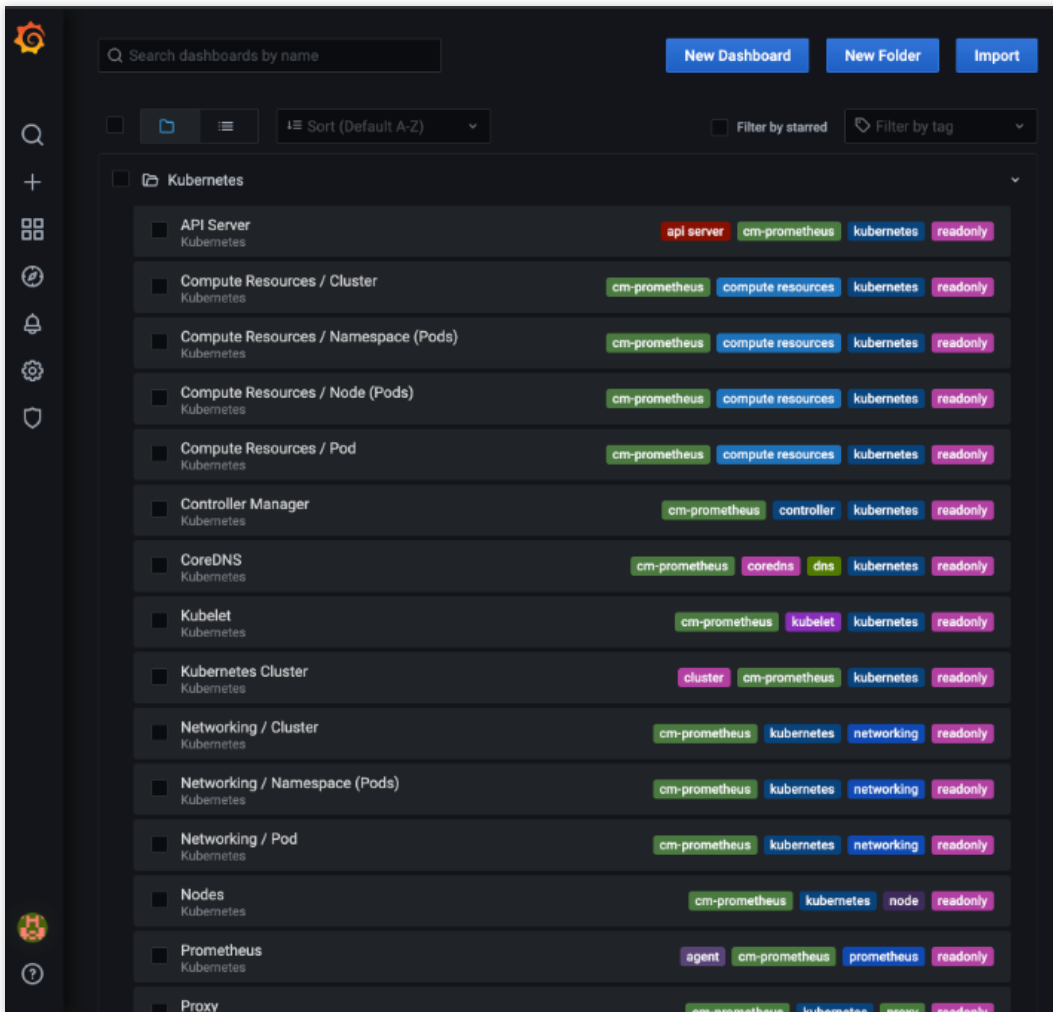
1. 登录 [Prometheus 监控服务控制台](#)。
2. 在 Prometheus 实例列表中，单击新建的**实例 ID/名称**。
3. 进入 Prometheus 管理中心，在左侧列表中单击**集成容器服务**。
4. 在集群监控页面进行下列操作：

关联集群：将集群和 Prometheus 实例关联，参见指引 [关联集群](#)。

数据采集配置：支持通过控制台新增或 YAML 文件配置两种方式，创建新的数据采集规则来监控您的业务数据，参见指引 [数据采集配置](#)。

精简基础监控指标：选择需要上报的指标，避免不必要的费用支出，参见指引 [精简基础监控指标](#)。

5. 完成以上操作后，即可在 Grafana 查看您容器服务的监控数据。



集成中心

最近更新时间：2024-01-29 16:01:49

Prometheus 监控服务对常用的开发语言/中间件/大数据/基础设施数据库进行了集成，支持一键安装和自定义安装方式，用户只需根据指引即可对相应的组件进行监控，同时提供了开箱即用的 Grafana 监控大盘。集成中心涵盖了基础服务监控，应用层监控、Kubernetes 容器监控三大监控场景，方便您快速接入并使用。

支持服务列表

服务类型	服务名称	监控项	是否支持一键安装	接入文档
大数据	ElasticSearch	包括集群/索引/节点等监控	支持	ElasticSearch Exporter 接入
	Flink	包括集群/Job/Task 等监控	不支持	Flink 接入
开发	CVM 云服务器	使用扩展的 cvm_sd_config 配置 CVM 抓取任务，采集 node-exporter 或业务自定义指标	支持	CVM node_exporter
	Golang	包括 GC/Heap/Thread/Goroutine 等监控	不支持	Golang 应用接入
	JVM	包括 Heap/Thread/GC/CPU/File 等监控	不支持	JVM 接入
	Spring MVC	包括 HTTP接口/异常/JVM 等监控	不支持	Spring Boot 接入
中间件	Kafka	包括 Broker/Topic/Consumer Group 等监控	支持	Kafka Exporter 接入
	Consul	Consul 监控	支持	Consul Exporter 接入
	Etcd	Etcd 监控	不支持	-
	Istio	Istio 监控	不支持	-
基础设施	Kubernetes	包括 API Server/DNS/Workload/Network 等监控	支持	安装 Agent 接入 Kubernetes
数据库	云数据库	包括文档数/读写性能/网络流量等	支持	MongoDB Exporter

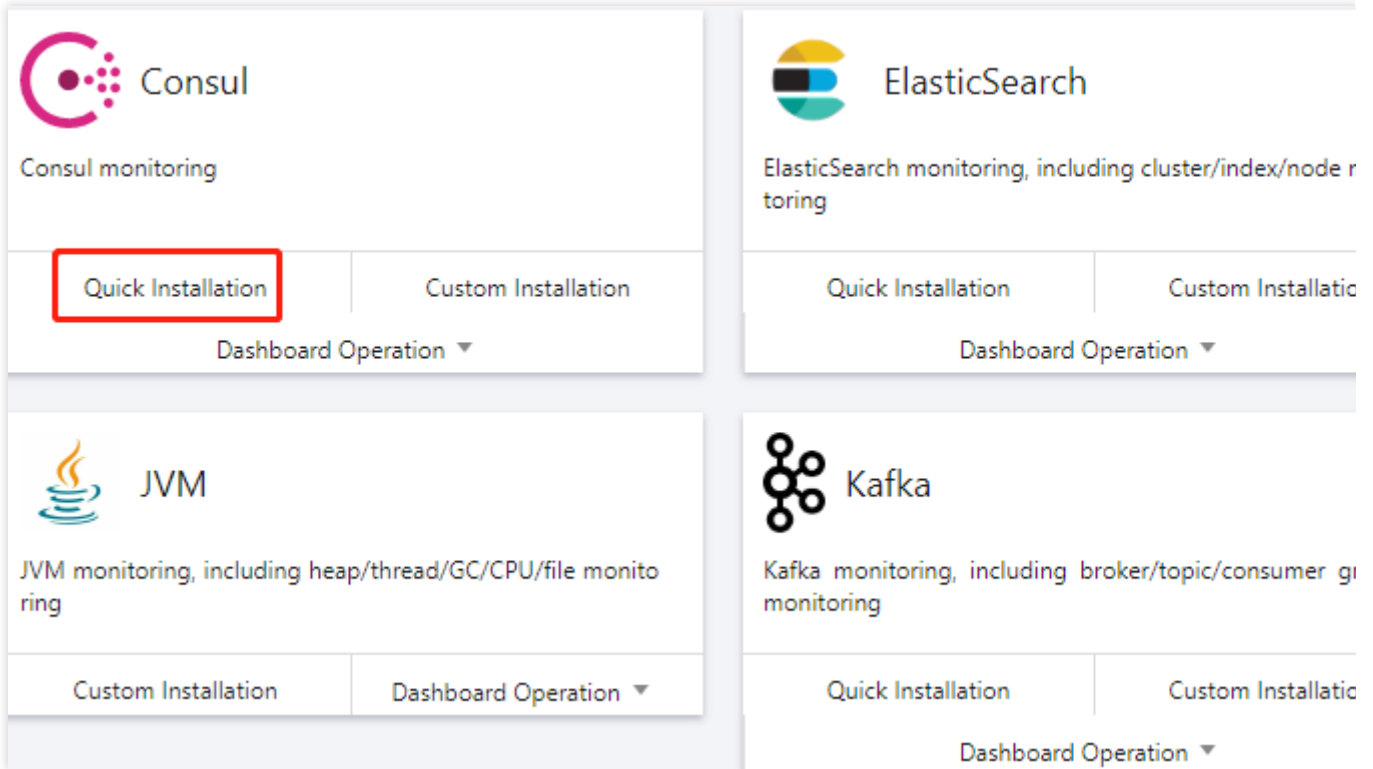
	MongoDB			接入
	云数据库 MySQL	包括网络/连接数/慢查询等	支持	MySQL Exporter 接入
	云数据库 PostgreSQL	包括 CPU/Memory/事务/Lock/读写等监控	支持	PostgreSQL Exporter 接入
	云数据库 Redis	包括内存使用率/连接数/命令执行情况等监控	支持	Redis Exporter 接入
	云数据库 Memcached	Memcached 监控	支持	Memcached Exporter 接入
巡检	健康巡检	通过 Blackbox 定期对目标服务进行连通性测试，帮助您掌握服务的健康状况，及时发现异常	支持	健康巡检
云监控	云监控	云产品监控	支持	-
自定义	抓取任务	使用原生 static_config 配置抓取任务	支持	抓取配置说明
	CVM 抓取任务	使用扩展的 cvm_sd_config 配置 CVM 抓取任务	支持	抓取配置说明

操作步骤

一键安装

部分服务支持一键安装 Agent，详情请参见 [支持服务列表](#)。

1. 登录 [云监控 Prometheus 控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例。
3. 进入实例详情页，单击 **集成中心**。
4. 在集成中心选择支持一键安装的服务，单击模块左下角的 **安装**。



5. 在集成列表页，填写指标采集名称和地址等信息，并单击保存即可。如下图以 Kafka 为例：

Kafka metric collection

name *

Kafka instance

address * [+ Add](#)

tag ⓘ [+ Add](#)

Exporter config

topic regular

group regular

[Save](#) [Cancel](#) Extra costs will be incurred. [Billing Overview](#) [🔗](#)










自定义安装

1. 登录 [云监控 Prometheus 控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例。
3. 进入实例详情页，单击 **集成中心**。
4. 在集成中心选择对应的服务。您可以单击 **接入指南** 查看接入指引，接入成功后即可实时监控对应的服务。您还可以单击 **Dashboard 安装/升级** 安装或升级该服务的 Grafana Dashboard。

Integration Center

Search for access mode by keyword

Category: [All](#) [Middleware](#) [Big Data](#) [Application](#) [Infrastructure](#) [Database](#)

 Consul Consul monitoring Quick Installation Custom Installation Dashboard Operation ▾	 ElasticSearch ElasticSearch monitoring, including cluster/index/node monitoring Quick Installation Custom Installation Dashboard Operation ▾	 Flink Flink monitoring, including cluster/job/task monitoring Custom Installation Dashboard Operation ▾
 JVM JVM monitoring, including heap/thread/GC/CPU/file monitoring Custom Installation Dashboard Operation ▾	 Kafka Kafka monitoring, including broker/topic/consumer group monitoring Quick Installation Custom Installation Dashboard Operation ▾	 Kubernetes Kubernetes monitoring, including API server/DNS/workload/network monitoring Custom Installation Dashboard Operation ▾
 MongoDB MongoDB instance monitoring, including file count/read and write performance/network traffic monitoring Quick Installation Custom Installation Dashboard Operation ▾	 MySQL MySQL instance monitoring, including network/connection count/slow query monitoring Quick Installation Custom Installation Dashboard Operation ▾	 PostgreSQL PostgreSQL instance monitoring, including CPU/memory/transaction/lock/read/write monitoring Quick Installation Custom Installation Dashboard Operation ▾

预聚合

预聚合概述

最近更新时间：2024-01-29 16:01:49

预聚合 (Recording Rule) 可以让我们对一些常用的指标或者计算相对复杂的指标进行提前计算，然后将这些数据存储到新的数据指标中，查询这些计算好的数据将比查询原始的数据更快更便捷。这对于 Dashboard 场景非常适用，可以解决用户配置以及查询慢的问题。

预聚合以规则组 (Rule Group) 的形式存在，相同组中的规则以一定的间隔顺序执行。聚合规则的名字必须符合 [相应的 Prometheus 规范](#)。

通常一个规则文件如下：



```
groups:  
  [ - <rule_group> ]
```

以下为一个简单预聚合规则例子：



```
groups:  
  - name: example  
    rules:  
      - record: job:http_inprogress_requests:sum  
        expr: sum by (job) (http_inprogress_requests)
```

规则组



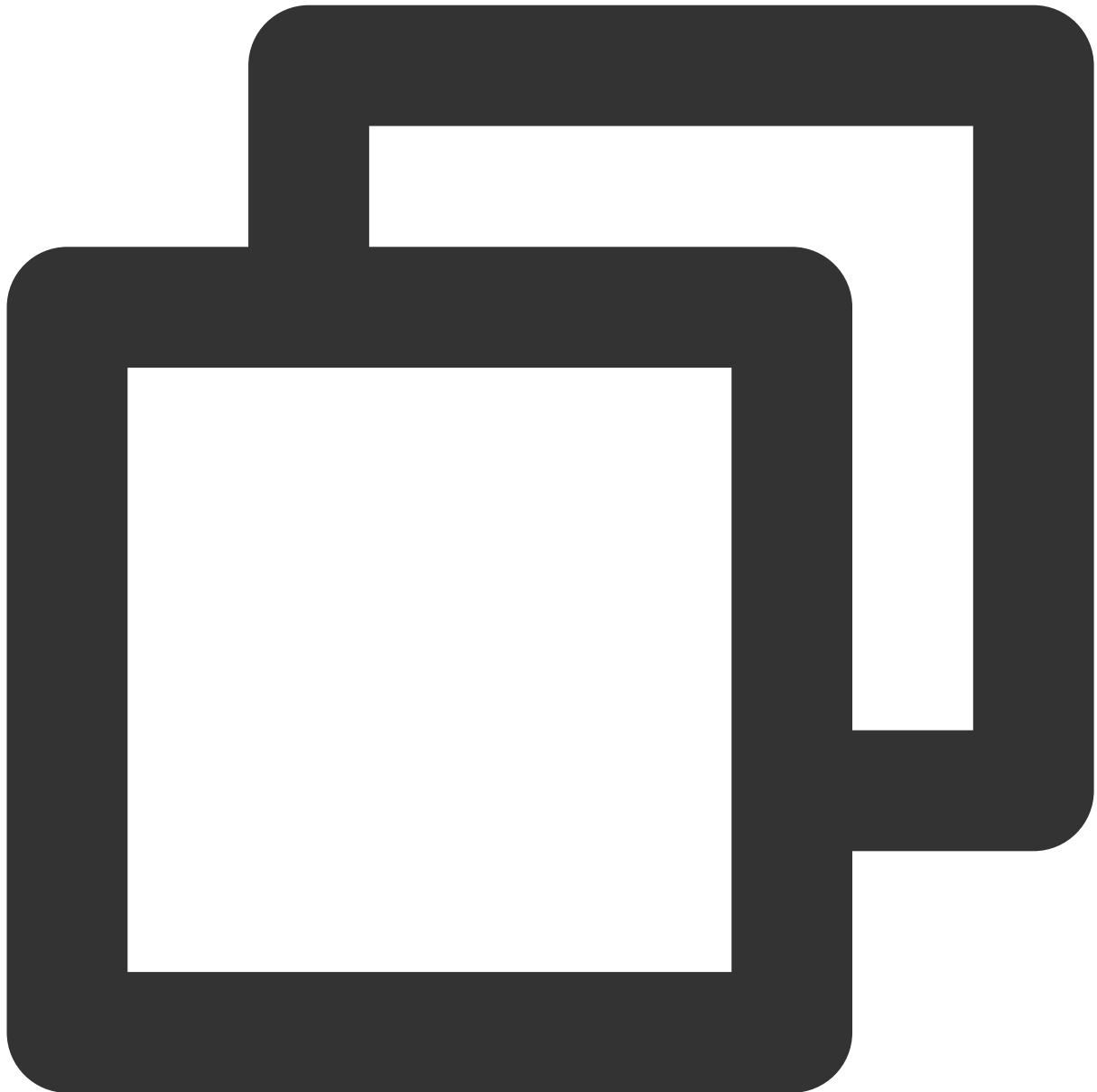
```
# 规则组名称, 在同一文件中必须唯一
name: <string>

# 规则探测周期.
[ interval: <duration> | default = global.evaluation_interval ]

rules:
  [ - <rule> ... ]
```

规则

预聚合的语法如下：



```
# 生成的新的指标名称，必须是一个有效的指标名称
```

```
record: <string>
```

```
# PromQL 表达式，每次计算的数据都会存储到新的指标名称 'record' 中
```

```
expr: <string>
```

```
# 在要存储的数据中所要添加或者覆盖的标签
```



```
labels:  
  [ <labelname>: <labelvalue> ]
```

推荐命名格式

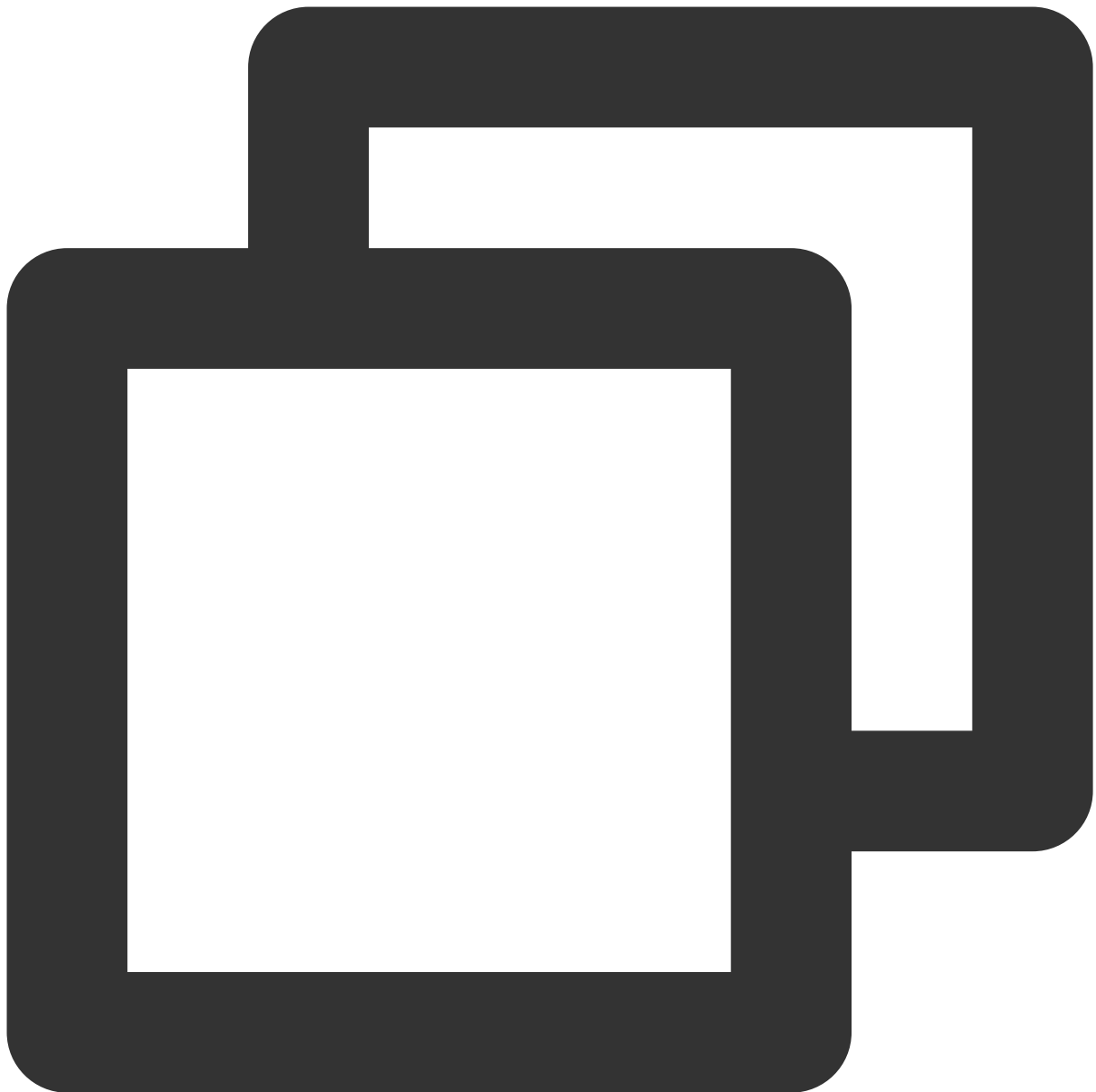
预聚合规则命名的推荐格式：`level:metric:operations`。

level：表示聚合级别，以及规则的输出标签。

metric：是指标名称。

operations：应用于指标的操作列表。

例如：



```
- record: instance_path:requests:rate5m
  expr: rate(requests_total{job="myjob"}[5m])

- record: path:requests:rate5m
  expr: sum without (instance)(instance_path:requests:rate5m{job="myjob"})
```

规则管理

最近更新时间：2024-01-29 16:01:49

操作场景

用户可以通过 Prometheus 监控服务控制台来管理 Prometheus 预聚合规则，以解决原生 Prometheus 需要修改配置文件的不便利性。

准备工作


1. 登录 [Prometheus 监控服务控制台](#)。
2. 创建 Prometheus 托管实例，详情请参见[创建实例](#)。
3. 通过实例列表进入到 Prometheus 实例的管理页面。
4. Prometheus 预聚合规则 (Recording Rule)，详情请参见[预聚合概述](#)。

操作步骤

新建规则

1. 通过实例管理页面的左侧菜单**预聚合** > **新建**打开规则新建页面，根据具体实际需求调整规则的表达式及需要聚合生成的新指标名，如下图，具体术语请参见[预聚合概述](#)。

Create Recording Rule (RecordingRule)

 Please use the native Prometheus recording rule YAML configurations. Note: you can enter configurations for only one group at a time.

Rule Group Name

YAML Configuration

```
1 name: example
2 rules:
3   - record: job:http_inprogress_requests:sum
4     expr: sum by (job) (http_inprogress_requests)
```

2. 单击**确定**即可。

管理规则

在规则列表中，可以对相应的规则进行临时的**禁用**或者对**未开启**的规则重新开启。禁用之后，规则将停止工作，相关的预聚合的指标将停止采集。

删除规则

1. 对一些不再使用的规则，可以进行删除操作。

2. 在列表中选择需要删除的规则，弹框确认之后，规则将会被删除，同时相关的规则将停止工作。

默认预聚合规则列表

最近更新时间：2024-01-29 16:01:49

关联集群会默认创建预聚合规则，[按量付费免费指标](#) 预聚合后生成下列新指标，下列指标将会正常计费。若您不需要采集下列指标，可在第一次关联集群时，或后续在预聚合页面的预聚合规则列表中取消开启默认预聚合规则。

metric（指标名称）	预设 Dashboard	计费
:node_memory_MemAvailable_bytes:sum	Kubernetes / Compute Resources / Cluster	-
node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	Kubernetes / Compute Resources / Cluster Kubernetes / Compute Resources / Workload	-
node_namespace_pod_container:container_memory_working_set_bytes	Kubernetes / Compute Resources / Pod	-
node_namespace_pod_container:container_memory_rss	Kubernetes / Compute Resources / Node (Pods)	-
node_namespace_pod_container:container_memory_cache	Kubernetes / Compute Resources / Namespace (Workloads)	-
node_namespace_pod_container:container_memory_swap	Kubernetes / Networking / Workload	-
namespace_workload_pod:kube_pod_owner:relabel	Kubernetes / Networking / Namespace (Workload)	-
namespace:kube_pod_container_resource_requests_memory_bytes:sum	-	K ¥
namespace:kube_pod_container_resource_requests_cpu_cores:sum	-	K ¥

告警策略

告警策略概述

最近更新时间：2024-01-29 16:01:49

Prometheus 监控服务支持您基于 Prometheus 的表达式设定告警条件。在指标达到告警条件时，通过邮件、短信告警渠道通知您采取措施。

说明：

Prometheus 监控服务告警结合了云监控告警能力和 Prometheus 开源生态告警能力，使告警更精准，更合理。

特性

支持 Alertmanager 的收敛、静默等特性，使告警合理。

支持指标数据进行聚合和告警规则进行周期性检测、计算，使告警更快速、更便捷。

告警规则支持 PromQL 进行定义，使告警更加灵活。

使用云监控告警通知模板，支持邮件、短信多种接收方式。

组成部分

术语	说明
策略名称	用户对告警策略命名。
告警规则	包含告警触发条件和持续时间，告警规则要由 PromQL 进行定义。
告警对象	自定义告警标题。
告警消息	自定义告警内容。
标签	用户指定要附加到告警上的一组附加标签。
注释	自定义告警附加消息。
通知模板	包含模板名称、通知类型、接收对象接收渠道，用户自定义接收方式和收敛方式。

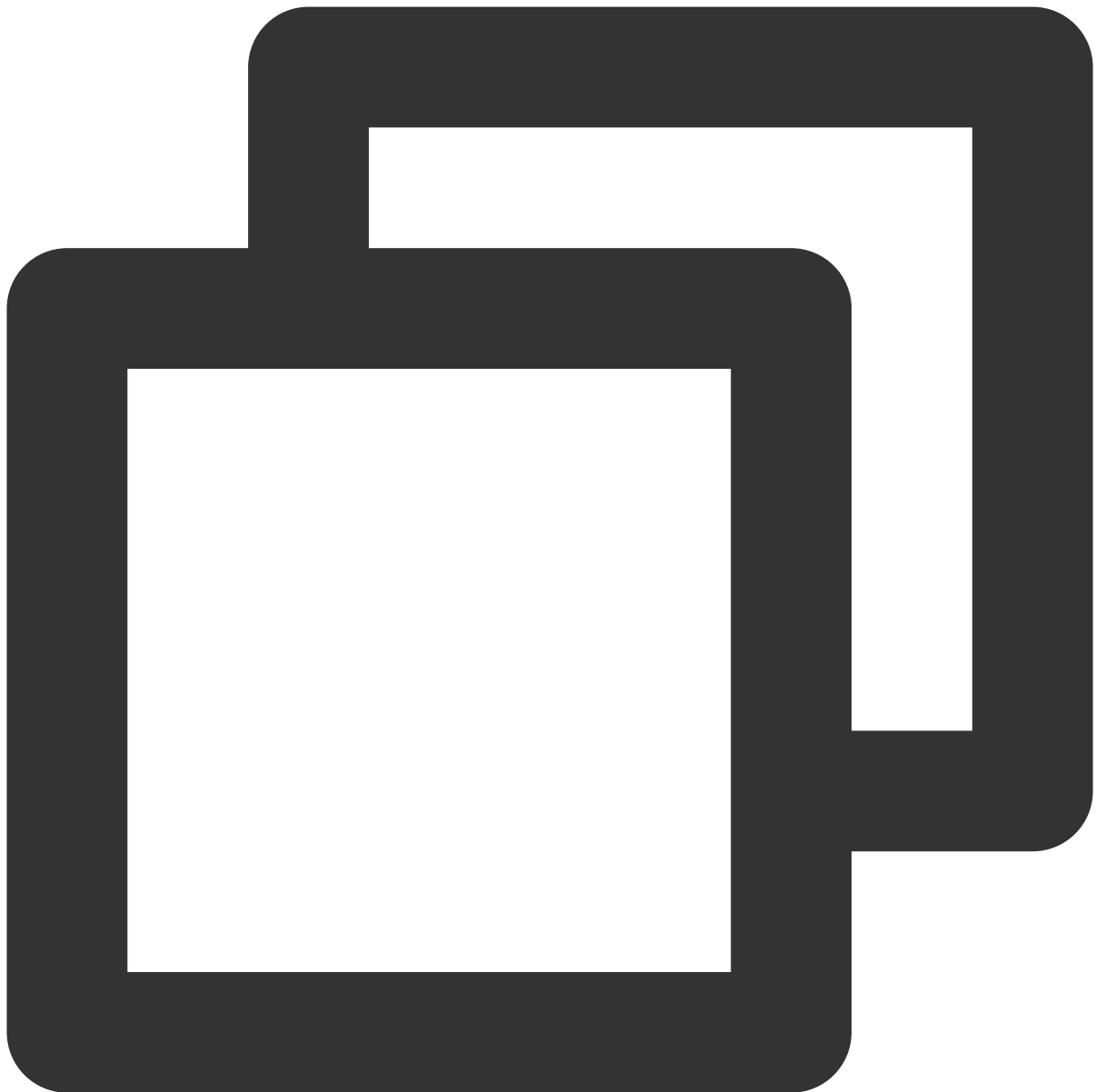
告警规则说明

最近更新时间：2024-01-29 16:01:49

告警规则允许我们基于 Prometheus 的表达式设定告警条件, 实时监控服务的状态, 及时通知触达服务异常情况.

如何定义一个告警规则

在 Prometheus 中, 告警规则和聚合规则的定义非常类似, 一个告警规则的示例可能如下:



```
groups:
- name: example
  rules:
- alert: HighRequestLatency
  expr: job:request_latency_seconds:mean5m{job="myjob"} > 0.5
  for: 10m
  labels:
    severity: page
  annotations:
    summary: High request latency
```

在告警规则文件中，我们可以将一组相关的规则设置定义在一个 `group` 下。在每一个 `group` 中我们可以定义多个告警规则 `rule`。一条告警规则主要由以下几部分组成：

`alert`：告警规则的名称。

`expr`：基于 PromQL 的表达式告警触发条件，用于计算是否有时间序列满足该条件。

`for`：评估等待时间，可选参数。用于表示只有当触发条件持续一段时间后才发送告警。在等待期间新产生告警的状态为 `pending`。

`labels`：自定义标签，允许用户指定要附加到告警上的一组附加标签。

`annotations`：用于指定一组附加信息，例如用于描述告警详细信息的文字等，`annotations` 的内容在告警产生时会一同作为参数发送到 `Alertmanager`。

模板

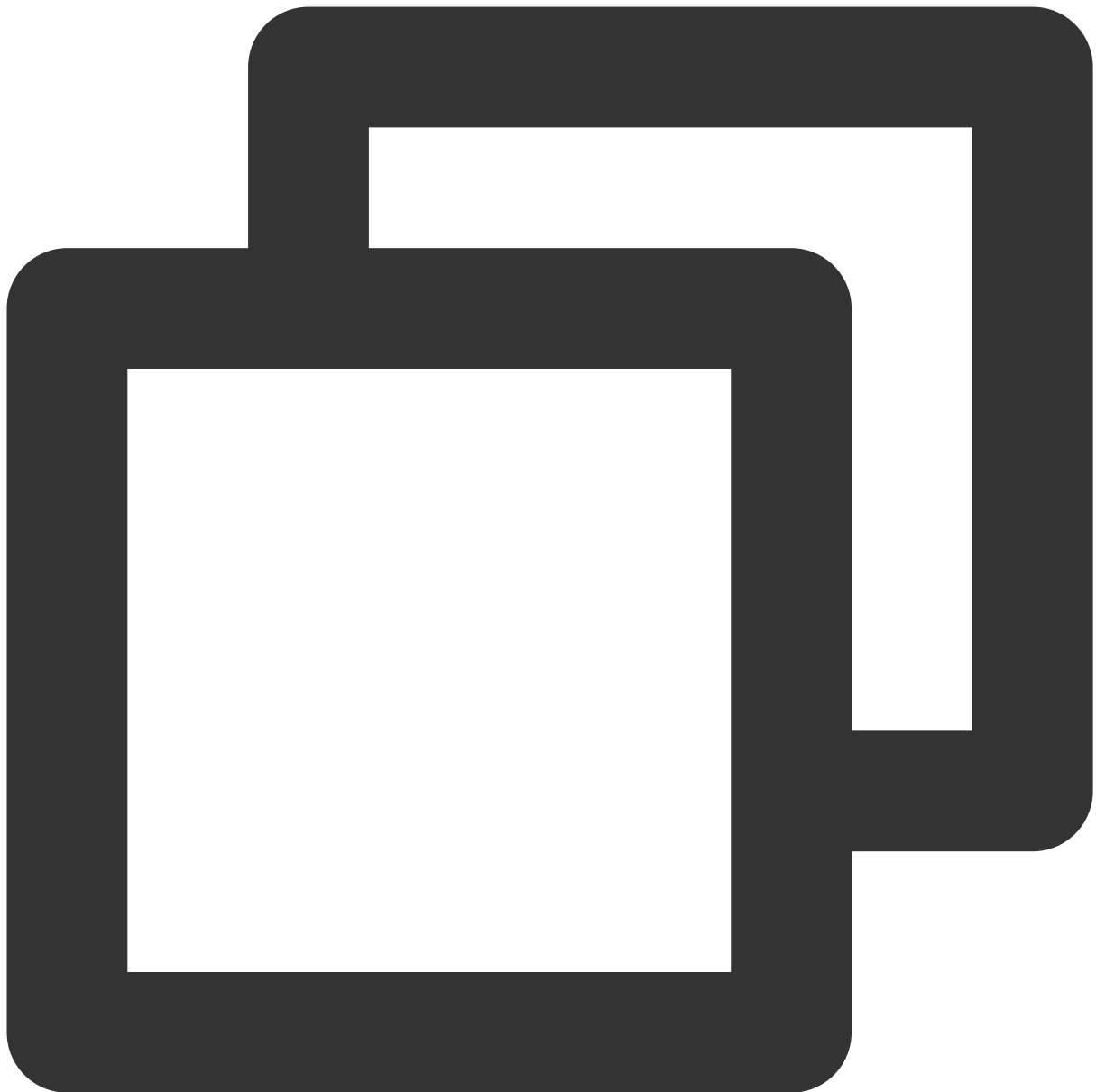
通常情况，在告警规则文件的 `annotations` 中使用 `summary` 描述告警的概要信息，`description` 用于描述告警的详细信息。同时 `Alertmanager` 的 UI 也会根据这两个标签值，显示告警信息。为使告警信息具有更好的可读性，Prometheus 支持模板化 `label` 和 `annotations` 的中标签的值。

通过 `$labels.<labelname>` 变量可以访问当前告警实例中指定标签的值。`$value` 则可以获取当前 PromQL 表达式计算的样本值。



```
# To insert a firing element's label values:
{{ $labels.<labelname> }}
# To insert the numeric expression value of the firing element:
{{ $value }}
```

例如，可以通过模板化优化 `summary` 以及 `description` 的内容的可读性：



```
groups:
- name: example
  rules:

  # Alert for any instance that is unreachable for >5 minutes.
  - alert: InstanceDown
    expr: up == 0
    for: 5m
    labels:
      severity: page
    annotations:
```

```
summary: "Instance {{ $labels.instance }} down"
description: "{{ $labels.instance }} of job {{ $labels.job }} has been down f

# Alert for any instance that has a median request latency >1s.
- alert: APIHighRequestLatency
  expr: api_http_request_latencies_second{quantile="0.5"} > 1
  for: 10m
  annotations:
    summary: "High request latency on {{ $labels.instance }}"
    description: "{{ $labels.instance }} has a median request latency above 1s (c
```

新建告警策略

最近更新时间：2024-01-29 16:01:49

本文指导您在 Prometheus 监控服务控制台中创建告警策略，在某些指标发生异常时及时通知您采取措施。

前提条件

已创建 [腾讯云容器服务—托管版集群](#)：在腾讯云容器服务中创建 Kubernetes 集群。

已创建 Prometheus 实例。

已安装 Prometheus Agent 及其监控组件。

操作步骤

详细的告警规则说明，请参见告警规则说明。

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例，单击左侧菜单栏的**告警策略**。
3. 在告警策略管理页中单击**新建**，在弹框中配置告警策略信息。

策略模板类型：选择策略模板类型，详细说明请参见告警策略类型说明。

策略名称：可使用默认策略名称也可自定义。

规则 PromQL：可使用默认模板也可自定义，表示基于 PromQL 的表达式告警触发条件，用于计算是否有时间序列满足该条件。

持续时间：可使用默认模板也可自定义，表示当触发条件持续多少时间后才发送告警。

告警对象：允许用户自定义告警标题。

告警消息：可使用默认模板也可自定义，表示允许用户自定义告警内容。

高级配置：单击开启配置，包含标签和注释配置。

标签：可使用默认模板也可自定义，表示允许用户指定要附加到告警上的一组附加标签，可根据接收到告警的标签匹配相应的处理方式。

注释：可使用默认模板也可自定义，表示允许用户定义告警附加消息。

告警通知：支持自定义告警通知模板，包含模板名称、通知类型、接收对象接收渠道等，详情请参见通知模板。

Policy Template *

Policy Name *

PromQL-Based Rule *

Duration

Alarm Object *

Alarm Message *

Labels

Key: Value: [Save](#)

Annotations

Key: Value: [Save](#)

Alarm Notification * [Create](#)

0 selected. 3 more can be selected

Notification Template Name	Included Operations	Opera
The notification template list is empty. You can select some by clicking "Select Template".		

暂停告警策略

最近更新时间：2024-01-29 16:01:49

本文指导您在 Prometheus 监控服务控制台中，如何关闭目标实例下的告警策略。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表中，选择对应的 Prometheus 实例，单击左边菜单栏的**告警策略**。
3. 找到需要关闭的告警策略，在操作列表中单击**关闭**。
4. 在弹框中单击**确定**即可。

Disable Alarm Policy

 Are you sure you want to delete the selected alarm policy?

OK

Cancel

策略类型说明

最近更新时间：2024-01-29 16:01:49

Prometheus 监控服务为 Kubernetes 集群预设了 [Master 组件](#)，[Kubelet](#)，[资源使用](#)，[工作负载](#) 和 [节点](#) 报警模板。

Kubernetes Master 组件

非托管集群提供如下指标：

策略名称	策略表达式	持续时间	策略描述
客户端访问 APIServer 出错	$(\text{sum}(\text{rate}(\text{rest_client_requests_total}\{\text{code}=\sim"5..\"}[5\text{m}])) \text{ by } (\text{instance}, \text{job}, \text{cluster_id}) / \text{sum}(\text{rate}(\text{rest_client_requests_total}[5\text{m}])) \text{ by } (\text{instance}, \text{job}, \text{cluster_id})) > 0.01$	15m	客户端访问 APIServe 出错率大于1%
客户端访问 APIServer 证书快过期	$\text{apiserver_client_certificate_expiration_seconds_count}\{\text{job}=\text{"apiserver"}\} > 0$ and on(job) histogram_quantile(0.01, sum by (cluster_id, job, le) (rate(apiserver_client_certificate_expiration_seconds_bucket{job="apiserver"} [5m]))) < 86400	无	访问 APIServe 的客户端 证书将在 24小时后 过期
聚合 API 出错	$\text{sum by}(\text{cluster_id}, \text{name}, \text{namespace})$ (increase(agggregator_unavailable_apiservice_count[5m])) > 2	无	聚合 API 最近5分 钟报错
聚合 API 可用性低	$(1 - \text{max by}(\text{name}, \text{namespace}, \text{cluster_id})$ (avg_over_time(agggregator_unavailable_apiservice[5m]))) * 100 < 90	5m	聚合 API 服务最近 5分钟可 用性低于 90%
APIServer 故障	$\text{absent}(\text{sum}(\text{up}\{\text{job}=\text{"apiserver"}\}) \text{ by } (\text{cluster_id}) > 0)$	5m	APIServe 从采集目 标中消失
Scheduler 故障	$\text{absent}(\text{sum}(\text{up}\{\text{job}=\text{"kube-scheduler"}\}) \text{ by } (\text{cluster_id}) > 0)$	15m	Schedule 从采集目 标中消失

Controller Manager 故障	<code>absent(sum(up{job="kube-controller-manager"}) by (cluster_id) > 0)</code>	15m	Controlle Manager 从采集目标中消失
-----------------------	--	-----	----------------------------

Kubelet

策略名称	策略表达式	持续时间	策略描述
Node 状态异常	<code>kube_node_status_condition{job=~".*kube-state-metrics",condition="Ready",status="true"} == 0</code>	15m	Node 状态异常持续15m
Node 不可达	<code>kube_node_spec_taint{job=~".*kube-state-metrics",key="node.kubernetes.io/unreachable",effect="NoSchedule"} == 1</code>	15m	Node 不可达, 上面的工作负载会重新调度
Node 上运行太多 pod	<code>count by(cluster_id, node) ((kube_pod_status_phase{job=~".*kube-state-metrics",phase="Running"} == 1) * on(instance,pod,namespace,cluster_id) group_left(node) topk by(instance,pod,namespace,cluster_id) (1, kube_pod_info{job=~".*kube-state-metrics"}))/max by(cluster_id, node) (kube_node_status_capacity_pods{job=~".*kube-state-metrics"} != 1) > 0.95</code>	15m	Node 上运行 pod 量快达到上限
Node 状态抖动	<code>sum(changes(kube_node_status_condition{status="true",condition="Ready"}[15m])) by (cluster_id, node) > 2</code>	15m	Node 状态在正常和异常之间抖动
Kubelet 的客户端证书快过期	<code>kubelet_certificate_manager_client_ttl_seconds < 86400</code>	无	Kubelet 客户端证书将在 24小时后过期
Kubelet 的服务端证书快过期	<code>kubelet_certificate_manager_server_ttl_seconds < 86400</code>	无	Kubelet 服务端证书将在 24小时后过期
Kubelet	<code>increase(kubelet_certificate_manager_client_expiration_renew_errors[5m])</code>	15m	Kubelet

客户端证书续签出错	> 0		续签客户端证书出错
Kubelet 服务端证书续签出错	increase(kubelet_server_expiration_renew_errors[5m]) > 0	15m	Kubelet 续签服务端证书出错
PLEG 耗时高	histogram_quantile(0.99, sum(rate(kubelet_peg_relist_duration_seconds_bucket[5m])) by (cluster_id, instance, le) * on(instance, cluster_id) group_left(node) kubelet_node_name{job="kubelet"}) >= 10	5m	PLEG 操作耗时的 99分位数超过 10秒
Pod 启动耗时高	histogram_quantile(0.99, sum(rate(kubelet_pod_worker_duration_seconds_bucket{job="kubelet"}[5m])) by (cluster_id, instance, le) * on(cluster_id, instance) group_left(node) kubelet_node_name{job="kubelet"}) > 60	15m	Pod 启动耗时的 99分位数值超过 60秒
Kubelet 故障	absent(sum(up{job="kubelet"}) by (cluster_id) > 0)	15m	Kubelet 从采集目标消失

Kubernetes 资源使用

策略名称	策略表达式	持续时间	策略描述
集群 CPU 资源过载	sum by (cluster_id) (max by (cluster_id, namespace, pod, container) (kube_pod_container_resource_requests_cpu_cores{job=~".*kubernetes-state-metrics"}) * on(cluster_id, namespace, pod) group_left() max by (cluster_id, namespace, pod) (kube_pod_status_phase{phase=~"Pending Running"} == 1))/sum by (cluster_id) (kube_node_status_allocatable_cpu_cores)>(count by (cluster_id) (kube_node_status_allocatable_cpu_cores)-1) / count by (cluster_id) (kube_node_status_allocatable_cpu_cores)	5m	集群内 Pod 申请的 CPU 总量过多，已无法容忍 Node 挂掉
集群内存资源过载	sum by (cluster_id) (max by (cluster_id, namespace, pod, container) (kube_pod_container_resource_requests_memory_bytes{job=~".*kubernetes-state-metrics"}) * on(cluster_id, namespace, pod) group_left() max by (cluster_id, namespace, pod) (kube_pod_status_phase{phase=~"Pending Running"} == 1))/sum by	5m	集群内 Pod 申请的内存总量过多，已无法容

	$(cluster_id) (kubernetes_node_status_allocatable_memory_bytes) > (count\ by\ (cluster_id) (kubernetes_node_status_allocatable_memory_bytes)-1) / count\ by\ (cluster_id) (kubernetes_node_status_allocatable_memory_bytes)$		忍 Node 挂掉
集群 CPU 配额过载	$sum\ by\ (cluster_id) (kubernetes_resourcequota\{job=\sim\}.*kubernetes_state-metrics",\ type="hard",\ resource="cpu") / sum\ by\ (cluster_id) (kubernetes_node_status_allocatable_cpu_cores) > 1.5$	5m	集群内 CPU 配额超过可分配 CPU 总量
集群内存配额过载	$sum\ by\ (cluster_id) (kubernetes_resourcequota\{job=\sim\}.*kubernetes_state-metrics",\ type="hard",\ resource="memory") / sum\ by\ (cluster_id) (kubernetes_node_status_allocatable_memory_bytes) > 1.5$	5m	集群内内存配额超过可分配内存总量
配额资源快使用完	$sum\ by\ (cluster_id, namespace, resource) kubernetes_resourcequota\{job=\sim\}.*kubernetes_state-metrics",\ type="used" / sum\ by\ (cluster_id, namespace, resource) (kubernetes_resourcequota\{job=\sim\}.*kubernetes_state-metrics",\ type="hard") > 0) >= 0.9$	15m	配额资源使用率超过90%
CPU 执行周期受限占比高	$sum(increase(container_cpu_cfs_throttled_periods_total\{container!\="",\}[5m]))\ by\ (cluster_id, container, pod, namespace) / sum(increase(container_cpu_cfs_periods_total\{\}[5m]))\ by\ (cluster_id, container, pod, namespace) > (25 / 100)$	15m	CPU 执行周期受到限制的占比高
Pod 的 CPU 使用率高	$sum(rate(container_cpu_usage_seconds_total\{job="kubelet",\ metrics_path="/metrics/cadvisor",\ image!\="",\ container!="POD"\}[1m]))\ by\ (cluster_id, namespace, pod, container) / sum(kubernetes_pod_container_resource_limits_cpu_cores)\ by\ (cluster_id, namespace, pod, container) > 0.75$	15m	Pod 的 CPU 使用率超过 75%
Pod 的内存使用率高	$sum(rate(container_memory_working_set_bytes\{job="kubelet",\ metrics_path="/metrics/cadvisor",\ image!\="",\ container!="POD"\}[1m]))\ by\ (cluster_id, namespace, pod, container) / sum(kubernetes_pod_container_resource_limits_memory_bytes)\ by\ (cluster_id, namespace, pod, container) > 0.75$	15m	Pod 的内存使用率超过75%

Kubernetes 工作负载

策略名称	策略表达式	持续时间

Pod 频繁重启	<code>increase(kube_pod_container_status_restarts_total{job=~".*kubernetes-state-metrics"}[5m]) > 0</code>	15m	F i
Pod 状态异常	<code>sum by (namespace, pod, cluster_id) (max by(namespace, pod, cluster_id) (kube_pod_status_phase{job=~".*kubernetes-state-metrics", phase=~"Pending Unknown"}) * on(namespace, pod, cluster_id) group_left(owner_kind) topk by(namespace, pod) (1, max by(namespace, pod, owner_kind, cluster_id) (kube_pod_owner{owner_kind!="Job"}))) > 0</code>	15m	F f i
容器状态异常	<code>sum by (namespace, pod, container, cluster_id) (kube_pod_container_status_waiting_reason{job=~".*kubernetes-state-metrics"}) > 0</code>	1h	多 \
Deployment 部署版本不匹配	<code>kube_deployment_status_observed_generation{job=~".*kubernetes-state-metrics"} != kube_deployment_metadata_generation{job=~".*kubernetes-state-metrics"}</code>	15m	多 f i
Deployment 副本数不匹配	<code>(kube_deployment_spec_replicas{job=~".*kubernetes-state-metrics"} != kube_deployment_status_replicas_available{job=~".*kubernetes-state-metrics"}) and (changes(kube_deployment_status_replicas_updated{job=~".*kubernetes-state-metrics"}[5m]) == 0)</code>	15m	多 f i
Statefulset 部署版本不匹配	<code>kube_statefulset_status_observed_generation{job=~".*kubernetes-state-metrics"} != kube_statefulset_metadata_generation{job=~".*kubernetes-state-metrics"}</code>	15m	多 f i
Statefulset 副本数不匹配	<code>(kube_statefulset_status_replicas_ready{job=~".*kubernetes-state-metrics"} != kube_statefulset_status_replicas{job=~".*kubernetes-state-metrics"}) and (changes(kube_statefulset_status_replicas_updated{job=~".*kubernetes-state-metrics"}[5m]) == 0)</code>	15m	多 f i
Statefulset 更新未生效	<code>(maxwithout(revision)(kube_statefulset_status_current_revision{job=~".*kubernetes-state-metrics"}unless kube_statefulset_status_update_revision{job=~".*kubernetes-state-metrics"})*(kube_statefulset_replicas{job=~".*kubernetes-state-metrics"}!=kube_statefulset_status_replicas_updated{job=~".*kubernetes-state-metrics"})) and (changes(kube_statefulset_status_replicas_updated{job=~".*kubernetes-state-metrics"}[5m])==0)</code>	15m	多 f
Daemonset 变更卡住	<code>((kube_daemonset_status_current_number_scheduled{job=~".*kubernetes-state-metrics"}!=kube_daemonset_status_desired_number_scheduled{job=~".*kubernetes-state-metrics"}) or (kube_daemonset_status_number_misscheduled{job=~".*kubernetes-state-metrics"}!=0) or (kube_daemonset_updated_number_scheduled{job=~".*kubernetes-state-metrics"}!=kube_daemonset_status_desired_number_scheduled{job=~".*kubernetes-state-metrics"}))</code>	15m	[f

	<code>state-metrics")) or (kube_daemonset_status_number_available{job=~".*kube-state-metrics"}!=kube_daemonset_status_desired_number_scheduled{job=~".*kube-state-metrics"})) and (changes(kube_daemonset_updated_number_scheduled{job=~".*kube-state-metrics"}[5m])==0)</code>		
Daemonset 部分 node 未调度	<code>kube_daemonset_status_desired_number_scheduled{job=~".*kube-state-metrics"} - kube_daemonset_status_current_number_scheduled{job=~".*kube-state-metrics"} > 0</code>	10m	[~]
Daemonset 部分 node 被错误调度	<code>kube_daemonset_status_number_misscheduled{job=~".*kube-state-metrics"} > 0</code>	15m	[i r
Job 运行太久	<code>kube_job_spec_completions{job=~".*kube-state-metrics"} - kube_job_status_succeeded{job=~".*kube-state-metrics"} > 0</code>	12h	~ ~
Job 执行失败	<code>kube_job_failed{job=~".*kube-state-metrics"} > 0</code>	15m	~
副本数和 HPA 不匹配	<code>(kube_hpa_status_desired_replicas{job=~".*kube-state-metrics"} != kube_hpa_status_current_replicas{job=~".*kube-state-metrics"}) and changes(kube_hpa_status_current_replicas[15m]) == 0</code>	15m	~ ~ ~
副本数达到 HPA 最大值	<code>kube_hpa_status_current_replicas{job=~".*kube-state-metrics"} == kube_hpa_spec_max_replicas{job=~".*kube-state-metrics"}</code>	15m	~ ~ ~
PersistentVolume 状态异常	<code>kube_persistentvolume_status_phase{phase=~"Failed Pending",job=~".*kube-state-metrics"} > 0</code>	15m	F ~ F

Kubernetes 节点

策略名称	策略表达式	持续时间	策略描述
文件系统空间快耗尽	<code>(node_filesystem_avail_bytes{job="node-exporter",fstype!=""}/node_filesystem_size_bytes{job="node-exporter",fstype!=""}*100<15 and predict_linear(node_filesystem_avail_bytes{job="node-exporter",fstype!=""}[6h],4*60*60)<0 and node_filesystem_readonly{job="node-exporter",fstype!=""}==0)</code>	1h	文件系统空间预计在4小时使用完

文件系统空间使用率高	<code>(node_filesystem_avail_bytes{job="node-exporter",fstype!=""}/node_filesystem_size_bytes{job="node-exporter",fstype!=""}*100<5 and node_filesystem_readonly{job="node-exporter",fstype!=""}=0)</code>	1h	文件系统可用空间低于5%
文件系统inode快耗尽	<code>(node_filesystem_files_free{job="node-exporter",fstype!=""}/node_filesystem_files{job="node-exporter",fstype!=""}*100<20 and predict_linear(node_filesystem_files_free{job="node-exporter",fstype!=""}[6h],4*60*60)<0 and node_filesystem_readonly{job="node-exporter",fstype!=""}=0)</code>	1h	文件系统inode预计在4小时后使用完
文件系统inode使用率高	<code>(node_filesystem_files_free{job="node-exporter",fstype!=""}/node_filesystem_files{job="node-exporter",fstype!=""}*100<3 and node_filesystem_readonly{job="node-exporter",fstype!=""}=0)</code>	1h	文件系统可用inode低于3%
网卡状态不稳定	<code>changes(node_network_up{job="node-exporter",device!~"veth.+"}[2m])</code>	2m	网卡状态不稳定，在 up 和 down 间频繁变化
网卡接收出错	<code>increase(node_network_receive_errs_total[2m]) > 10</code>	1h	网卡接收数据出错
网卡发送出错	<code>increase(node_network_transmit_errs_total[2m]) > 10</code>	1h	网卡发送数据出错
机器时钟未同步	<code>min_over_time(node_timex_sync_status[5m]) == 0</code>	10m	机器时间最近未同步，检查 NTP 是否正常配置
机器时钟漂移	<code>(node_timex_offset_seconds>0.05 and deriv(node_timex_offset_seconds[5m])>=0) or (node_timex_offset_seconds<-0.05 and deriv(node_timex_offset_seconds[5m])<=0)</code>	10m	机器时间漂移超过300秒，检查 NTP 是否正常配置

通知模板

最近更新时间：2024-01-29 16:01:49

本文指导您在云监报告警模块中创建通知模板。

应用场景

多个策略一键复用模板，减少用户重复配置用户通知。

个性化配置用户通知方式。例如：白天可以配置告警接收渠道为邮件、短信。夜间可以配置告警接收渠道为电话。

前提条件

查看通知模板：子账号需拥有云监控读权限。

创建、编辑通知模板：子账号需拥有云监控写权限。

说明：

详情可参见 [访问权限](#) 进行子账号授权。

相关限制

功能	限制
用户通知	最多可添加五项
接口回调	最多填写三个公网可访问的 URL

操作步骤

新建通知模板

1. 登录 [腾讯云可观测平台 - 告警通知模板](#)。

2. 单击**新建**，在“新建通知模板”填写信息。

模板名称：自定义模板名称。

通知类型：

告警触发：告警触发时发送通知。

告警恢复：告警恢复时发送通知。

用户通知：

接收对象：可选接收组或接收人，如需创建告警接收组请参考 [创建告警接收组](#)。

通知时段：定义接收告警时间段。

接收渠道：支持邮箱、短信、微信、电话四种告警渠道。您还可以根据不同的用户维度，设置不同的告警接收渠道和通知时段。

电话告警配置说明：

轮询次数：在无有效触达时，对所有接收人逐一轮询拨打的最多次数。

轮询顺序：电话告警按照接收人顺序轮询拨打，上下拖动接收人调整拨打顺序。

轮询间隔：电话告警按照接收人顺序轮询拨打的时间间隔。

触达通知：电话成功接收或轮询结束发送信息给所有接收人。短信需计算额度。

接口回调：填写公网可访问到的url作为回调接口地址，云监控将及时把告警信息推送到该地址，当 HTTP 返回 200 为验证成功。告警回调字段说明请参考 [告警回调说明](#)。

说明：

回调地址保存后自动验证一次您的 URL，验证超时时间为5s；当用户创建的告警策略被触发或被恢复均会通过接口回调推送告警消息，此告警消息最多推送三次，每次请求的超时时间为5s。

当用户创建的告警策略被触发或恢复时，均会通过接口回调推送告警消息。接口回调也支持重复告警。

云监控回调 API 出方向 IP 为动态随机分配，无法将具体的 IP 信息提供给您，但 IP 端口固定为80端口，建议您根据80端口在安全组上配置加全放通策略。

←

New Notification Template

Basic Info

Template Name *

Notification Template ⓘ Alarm Trigger Alarm Recovery

Notification Language

Notifications (Fill in at least one item)

User Notification When adding a user, you can also add a user only for receiving messages.

Recipient Object Add User Delet

Notification Period ⓘ

Receiving Channel Email SMS

Add User Notification

API Callback ⓘ

Add API Callback

i It supports pushing to the WeCom group robotCome and try it out [↗](#)

Complete

默认通知模板

系统自动为您创建默认通知模板，模板内容如下：

功能	默认配置
模板名称	系统预设通知模板
通知类型	告警触发，告警恢复
告警接收人	主账号管理员
通知时间段	00:00:00 - 23:59:59（全天）

接收渠道

邮件、短信

删除模板

说明：

默认通知模板不支持删除。

1. 找到需要删除的模板名称，在操作列中单击**删除**。
2. 在弹框中确认删除即可。

复制模板

1. 找到需要复制的模板名称，在操作列中单击**复制**。
2. 在跳转页中修改信息或直接单击**完成**即可。

标签管理

标签示例

最近更新时间：2024-01-29 16:01:49

简介

标签是腾讯云提供的用于标识云上资源的标记，是一个键-值对（Key-Value）。

您可以根据各种维度（例如业务、用途、负责人等）使用标签对 Prometheus 监控资源进行分类管理，通过标签非常方便地筛选过滤出对应的资源。标签键值对对腾讯云没有任何语义意义，会严格按字符串进行解析匹配，腾讯云不会使用您设定的标签，标签仅用于您对资源的管理。

以下通过一个具体的案例来介绍标签的使用。

案例背景

某公司在腾讯云上拥有10个 Prometheus 监控服务实例，分属电商、游戏、文娱三个部门，服务于营销活动、游戏 A、游戏 B、后期制作等业务，三个部门对应的运维负责人为张三、赵四、王五。

设置标签

为了方便管理，该公司使用标签分类管理对应的 Prometheus 监控服务资源，定义了下述标签键/值。

标签键	标签值
部门	电商、游戏、文娱
业务	营销活动、游戏 A、游戏 B、后期制作
运维负责人	张三、赵四、王五

将这些标签键/值绑定到 Prometheus 实例上，资源与标签键/值的关系如下表所示：

实例 ID	部门	业务	运维负责人
prom-1jqwv1	电商	营销活动	王五
prom-1jqwv12	电商	营销活动	王五
prom-1jqwv13	游戏	游戏 A	张三
prom-1jqwv13	游戏	游戏 B	张三
prom-1jqwv14	游戏	游戏 B	张三
prom-1jqwv15	游戏	游戏 B	赵四

prom-1jqwv16	游戏	游戏 B	赵四
prom-1jqwv17	游戏	游戏 B	赵四
prom-1jqwv18	文娱	后期制作	王五
prom-1jqwv19	文娱	后期制作	王五
prom-1jqwv110	文娱	后期制作	王五

使用标签

筛选出王五负责的 Prometheus 监控服务实例

按照筛选规则筛选出运维负责人为“王五”的 Prometheus 资源即可，具体筛选办法请参见使用标签。

筛选出游戏部门中赵四负责的 Prometheus 监控服务实例

按照筛选规则筛选出部门为“游戏”、运维负责人为“赵四”的 Prometheus 资源即可，具体筛选办法请参见使用标签。

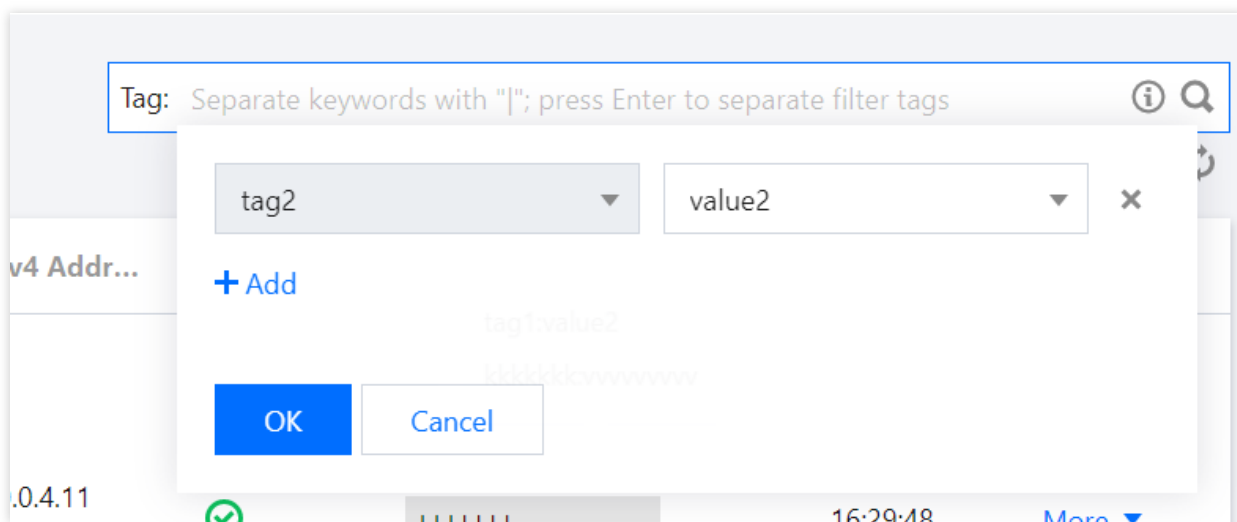
使用标签

最近更新时间：2024-01-29 16:01:49

本文指导您在 Prometheus 监控服务控制台中，根据标签对实例进行资源筛选，过滤出对应的资源。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例列表页顶部，选择地域。
3. 在实例列表右上角的搜索框，单击空白处，单击**标签**，弹出标签过滤选择框，如下图所示：



4. 在标签过滤选择框中选择对应的条件，单击**确定**之后进行过滤。
5. 如果需要调整对应的标签条件，单击搜索框里的**标签**后面的标签内容进行编辑。
6. 同时也支持直接单击实例列表中对应的标签值来进行过滤，如下图所示：

Create Edit Tag

Tag: tag2 value2 ✕ |
Separate keywords with "|"; press E

<input type="checkbox"/>	Instance ID/Name	Monit... ▾	AZ ▾	Network	Configura...	IPv4 Addr...	Billing Mo...	Tag (key:va
<input type="checkbox"/>	prom-1	Running	Guangzhou Zone 2	Network: default_vpc Subnet: default_vpc_subnet	Data retention period: 15 day(s) Specs name: Shared Edition		Trial Edition	<input type="text" value="tag2:value2"/> kkkkkk:vv tag1:value1

Total items: 1 10 ▾ /

编辑标签

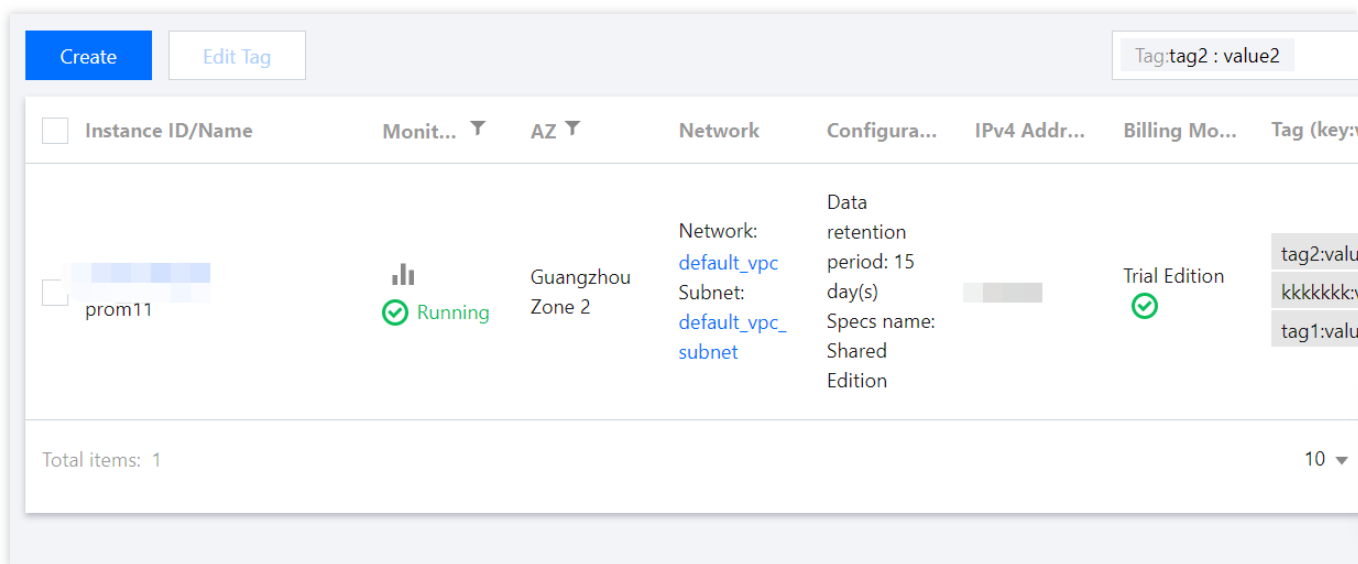
最近更新时间：2024-01-29 16:01:49

本文指导您在 Prometheus 监控服务控制台中，对目标实例进行标签的编辑操作。

操作步骤

对单个实例编辑标签

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在实例的管理页面，选择需要编辑标签的实例，选择**更多** > **实例配置** > **编辑标签**，如下图所示：



3. 在弹出的“您已经选择1个云资源”窗口中，根据实际需求进行添加、修改或者删除标签：

Edit Tags ✕

The tag is used to manage resources by category from different dimensions. If the existing tag does not meet your requirements, please go to [Manage Tags](#)

1 resource selected

tag2	value2	✕
kkkkkkk	wwwwwww	✕
tag1	value1	✕
Tag key	Tag value	✕

[+ Add](#)

OK Cancel

使用限制

最近更新时间：2024-01-29 16:01:49

标签是一个键-值对（Key-Value），您可以在 Prometheus 监控服务控制台，通过对 Prometheus 实例设置标签实现资源的分类管理。通过标签，可以非常方便筛选过滤出对应的资源。

数量限制

每个云资源允许的最大标签数是50。

标签键限制

qcloud、tencent、project 开头为系统预留标签键，禁止创建。

只能为字母、数字、空格或汉字，支持+、-、=、.、_、:、/、@。

标签键长度最大为255个字符。

标签值限制

只能为字母、数字、空格或汉字，支持+、-、=、.、_、:、/、@。

标签值最大长度为127个字符。

访问控制

访问控制概述

最近更新时间：2024-01-29 16:01:49

如果您在腾讯云中使用到了 Prometheus 监控服务，该服务由不同的人管理，但都共享您的云账号密钥，将存在以下问题：

您的密钥由多人共享，泄密风险高。

您无法限制其它人的访问权限，易产生误操作造成安全风险。

此时，您就可以通过子账号实现不同的人管理不同的服务，来规避以上的问题。默认情况下，子账号无使用 Prometheus 权限。因此，我们需要创建策略来允许子账号使用他们所需要资源的权限。

简介

[访问管理](#)（Cloud Access Management, CAM）是腾讯云提供的一套 Web 服务，它主要用于帮助客户安全管理腾讯云账户下的资源的访问权限。通过 CAM，您可以创建、管理和销毁用户（组），并通过身份管理和策略管理控制哪些人可以使用哪些腾讯云资源。

当您使用 CAM 时，可以将策略与一个用户或一组用户关联起来，策略能够授权或者拒绝用户使用指定资源完成指定任务。有关 CAM 策略的更多相关基本信息，请参见 [策略语法](#)。有关 CAM 策略的更多相关使用信息，请参见 [策略](#)。

若您无需对子账号进行 Prometheus 相关资源的访问管理，您可以跳过此章节。跳过这些部分不会影响您对文档中其余部分的理解和使用。

入门

CAM 策略必须授权使用一个或多个 Prometheus 操作或者必须拒绝使用一个或多个 Prometheus 操作。同时还必须指定可以用于操作的资源（可以是全部资源，某些操作也可以是部分资源），策略还可以包含操作资源所设置的条

件。
Prometheus 监控服务部分 API 操作不支持资源级权限，意味着，对于该类 API 操作，您无法在使用该类操作的时候指定某个具体的资源来使用，而必须指定全部资源来使用。

策略设置

最近更新时间：2024-01-29 16:01:49

概述

访问策略可用于授予访问 Prometheus 监控服务实例的权限。访问策略使用基于 JSON 的访问策略语言。您可以通过访问策略语言授权指定委托人（principal）对指定的 Prometheus 监控服务资源执行指定的操作。

访问策略语言描述了策略的基本元素和用法，有关策略语言的说明可参考 [CAM 策略管理](#)。

访问策略中的元素

访问策略语言包含以下基本意义的元素：

语句（statement）：描述一条或多条权限的详细信息。该元素包括效力、操作、资源、条件等多个其他元素的权限或权限集合。一条策略有且仅有一个语句元素。

效力（effect）：描述声明产生的结果是“允许”还是“显式拒绝”，包括 allow 和 deny 两种情况。该元素是必填项。

操作（action）：描述允许或拒绝的操作。操作可以是 API（以 name 前缀描述）或者功能集（一组特定的 API，以 permid 前缀描述）。该元素是必填项。

资源（resource）：描述授权的具体数据。资源是用六段式描述。每款产品的资源定义详情会有所区别。该元素是必填项。

条件（condition）：描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。有些服务允许您在条件中指定其他值。该元素是选填项。

元素用法

指定效力

如果没有显式授予（允许）对资源的访问权限，则隐式拒绝访问。同时，也可以显式拒绝（deny）对资源的访问，这样可确保用户无法访问该资源，即使有其他策略授予了访问权限的情况下也无法访问。下面是指定允许效力的示例：

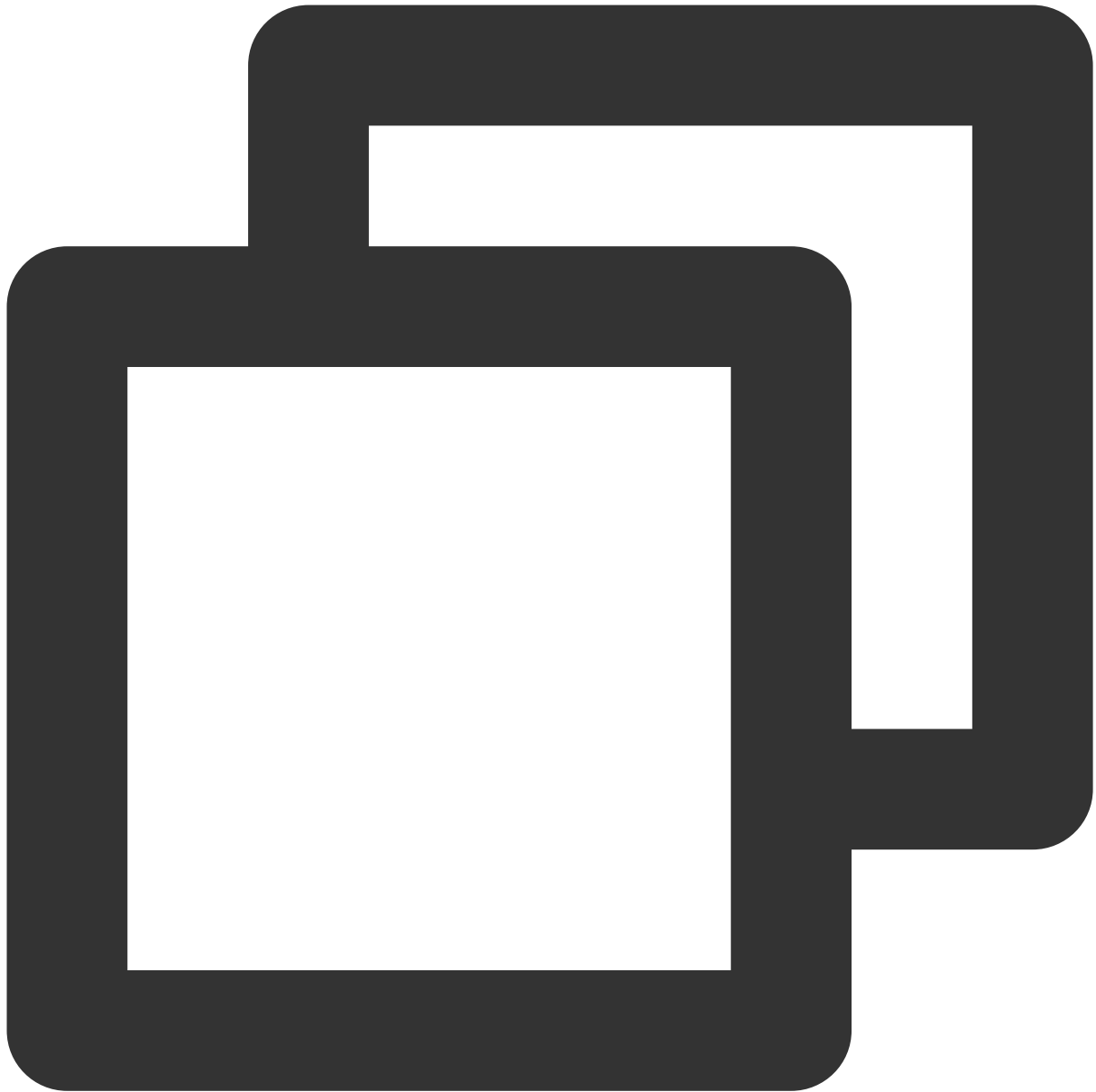


```
"effect" : "allow"
```

指定操作

云监控定义了可在策略中指定一类控制台的操作，指定的操作按照操作性质分为读取部分接口（`monitor:Describe*`）和全部接口（`monitor:*`）。

指定允许操作的示例如下：



```
"action": [  
  "name/monitor:Describe*"br/>]
```

指定资源

资源（resource）元素描述一个或多个操作对象，如腾讯云 Prometheus 监控服务资源等。所有资源均可采用下述的六段式描述方式。



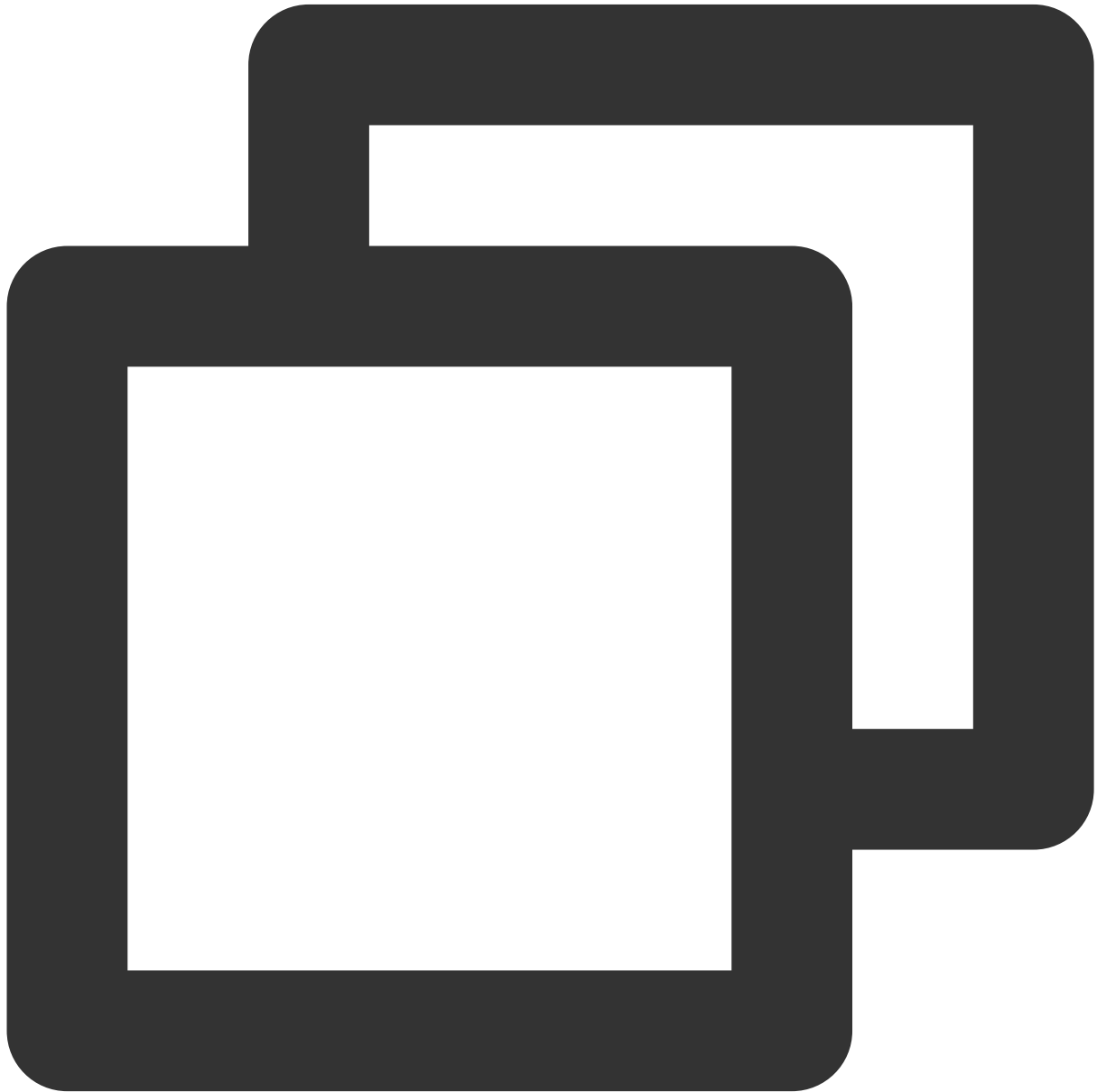
```
qcs:project_id:service_type:region:account:resource
```

参数说明如下：

参数	描述	是否必选
qcs	是 qcloud service 的简称，表示是腾讯云的云服务	是
project_id	描述项目信息，仅为了兼容 CAM 早期逻辑，一般不填	否

service_type	产品简称，这里为 monitor	是
region	描述地域信息	是
account	描述资源拥有者的主账号信息，即主账号的 ID，表示为 <code>uin/\${OwnerUin}</code> ，如 <code>uin/1000000000001</code>	是
resource	描述具体资源详情，前缀为 instance	是

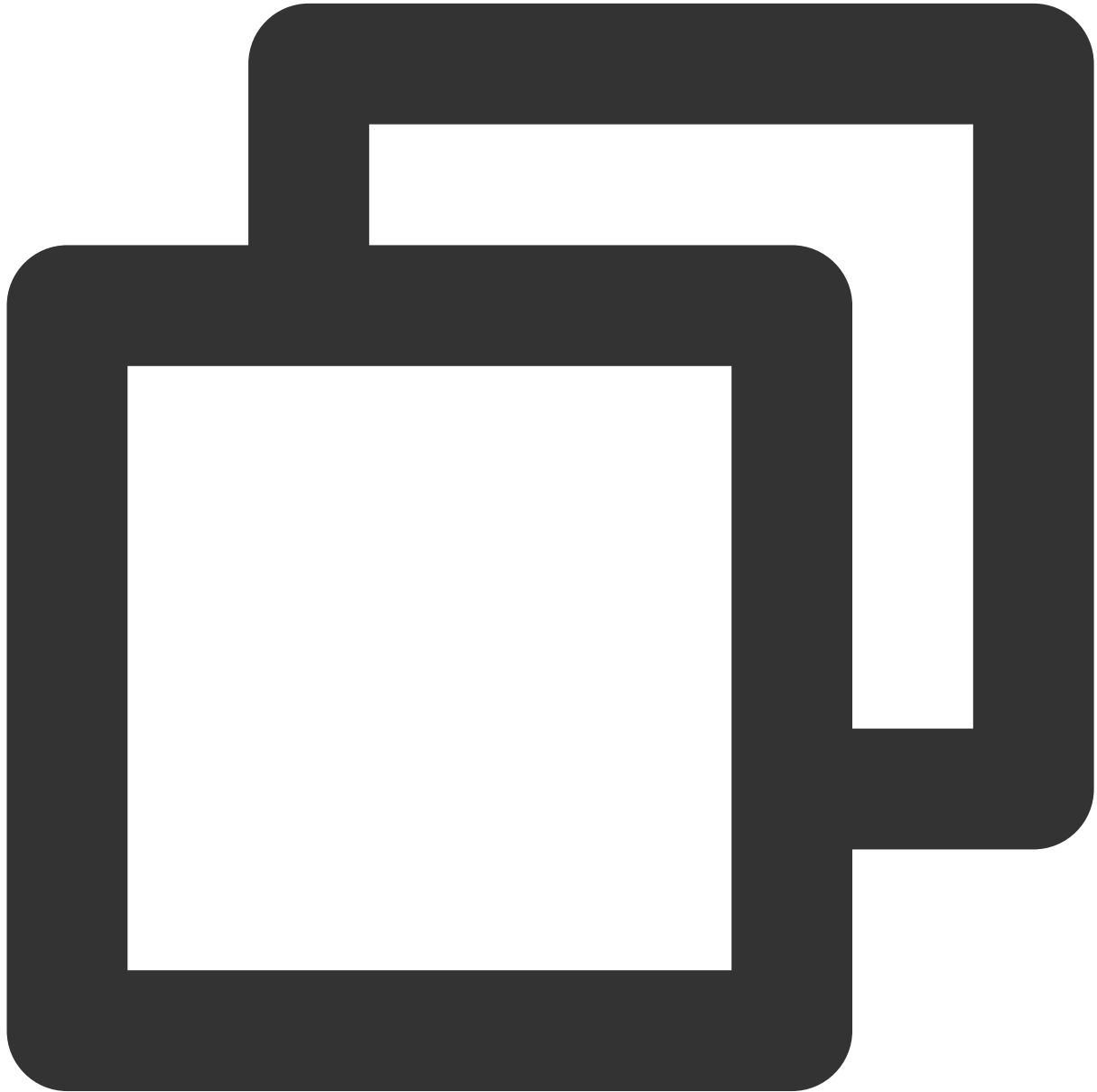
下面是一个 Prometheus 监控服务的六段式信息：




```
"resource":["qcs::monitor:ap-guangzhou:uin/100000000001:prom-instance/prom-73jingds
```

指定条件

访问策略语言可使您在授予权限时指定条件。主要是用于设置标签鉴权，标签条件只对绑定了该标签的集群生效，标签策略示例如下：



```
"condition": {  
  "for_any_value:string_equal": {  
    "qcs:tag": [  

```

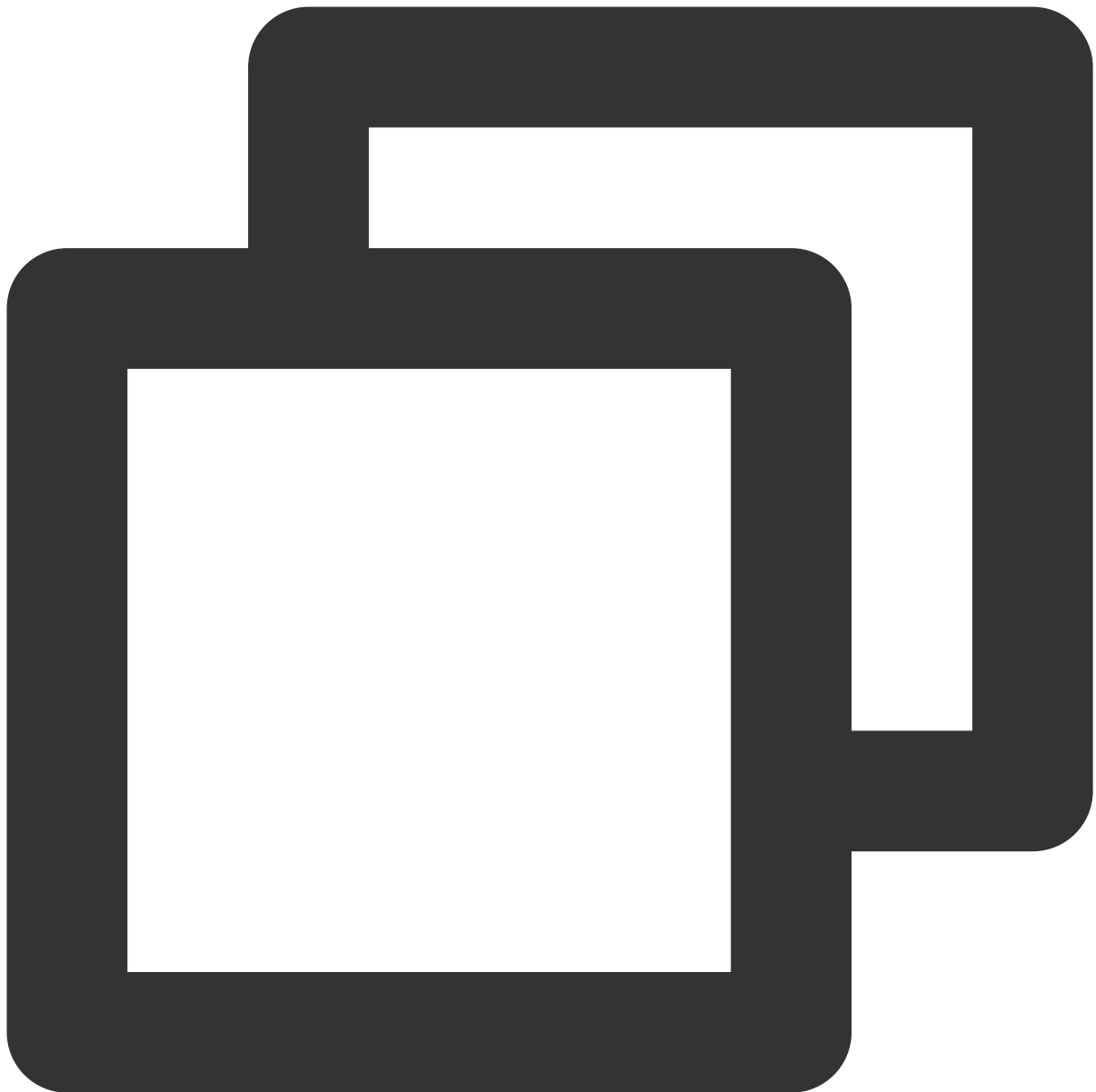
```
        "testkey&testvalue"  
    ]  
}  
}
```

这个语句含义是策略包含标签 key 为 testkey, value 为 testvalue 的资源。

实际案例

基于标签

如下案例中, 策略的含义是允许访问 UIN 为1250000000下面的实例 ID 为 prom-73jingds 的资源, 以及绑定了标签键为 testkey, 标签值为 testvalue 的资源 (如果该实例未属于标签 testkey& testvalue , 则仍不允许访问)。



```
{
  "version": "2.0",
  "statement": [
    {
      "action": [
        "name/monitor:*"
      ],
      "condition": {
        "for_any_value:string_equal": {
          "qcs:tag": [
            "testkey&testvalue"
          ]
        }
      }
    }
  ]
}
```

```
        ]
      },
      "effect": "allow",
      "resource": [
        "qcs::monitor:ap-guangzhou:uin/1250000000:prom-instance/prom-73jing"
      ]
    }
  ]
}
```

基于资源

基于资源 ID，分配指定资源的读写权限，主账号ID 为 1250000000：

配置广州地域（资源）只读权限。

示例：为子用户分配， instance1(prom-73jingds)、instance2(prom-65jdfafk) 资源的只读权限。

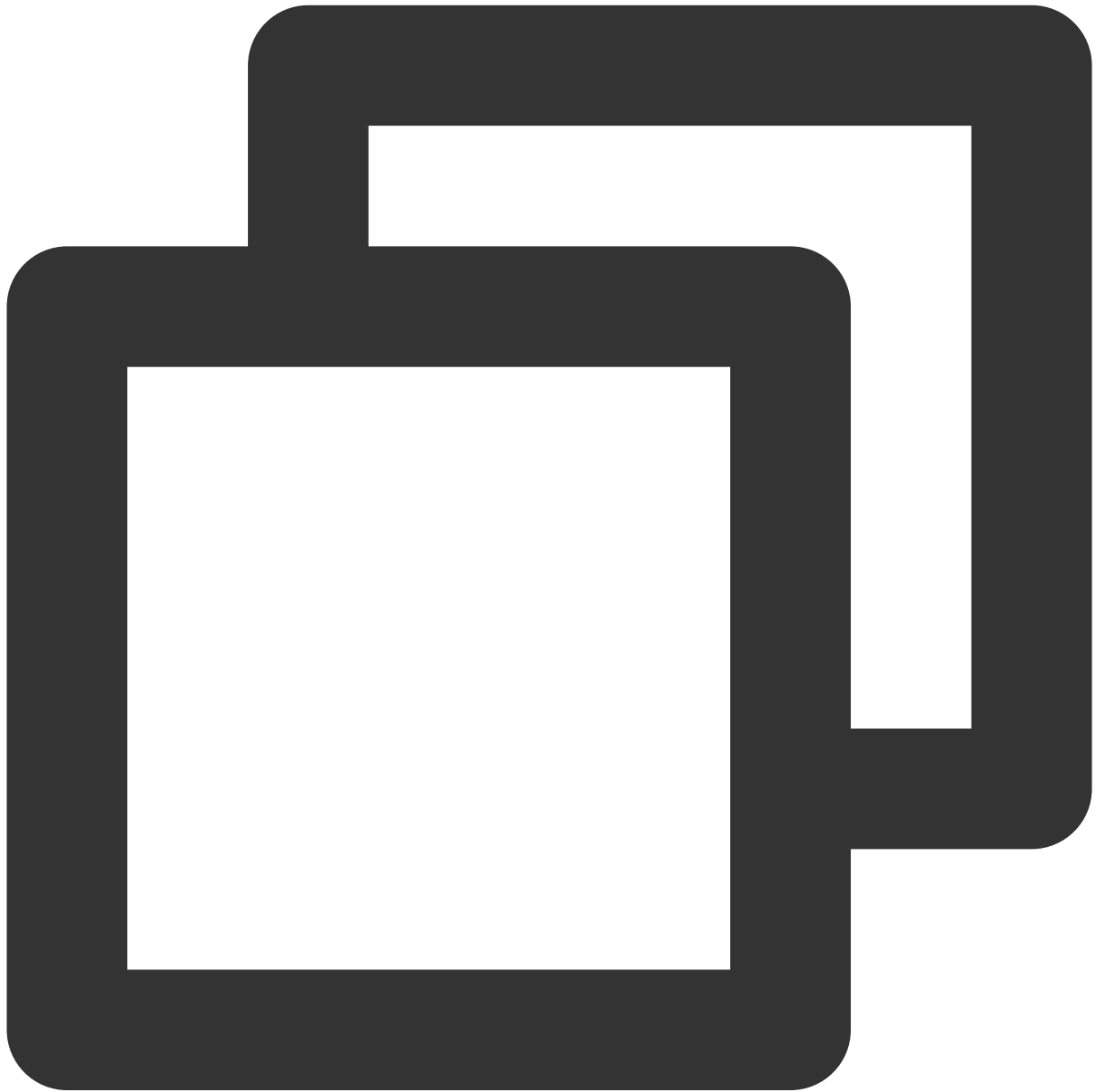


```
{
  "version": "2.0",
  "statement": [{
    "effect": "allow",
    "action": [
      "name/monitor:Describe*"
    ],
    "resource": [
      "qcs::monitor:ap-guangzhou:uin/1250000000:prom-instance/prom-73jingsd",
      "qcs::monitor:ap-guangzhou:uin/1250000000:prom-instance/prom-65jidfak"
    ],
  ]
}
```

```
    "condition": []  
  }]  
}
```

配置广州地域（资源）部分读写权限。

示例：为子用户分配，instance1(prom-73jingds) 资源的删除操作权限。



```
{  
  "version": "2.0",  
  "statement": [{  
    "effect": "allow",
```

```
    "resource": [  
      "qcs::monitor:ap-guangzhou:uin/1250000000:prom-instance/prom-73jingds"  
    ],  
    "action": [  
      "name/monitor:TerminatePrometheusInstances"  
    ]  
  }]  
}
```

策略授予

最近更新时间：2024-01-29 16:01:49

Prometheus 自定义策略

如果预设策略无法满足需求，可单击**新建自定义策略**创建自定义策略。

Policy

Associate users or user groups with policies to grant permissions.

Create Custom Policy Delete
All Policies Preset Policy Custom Policy

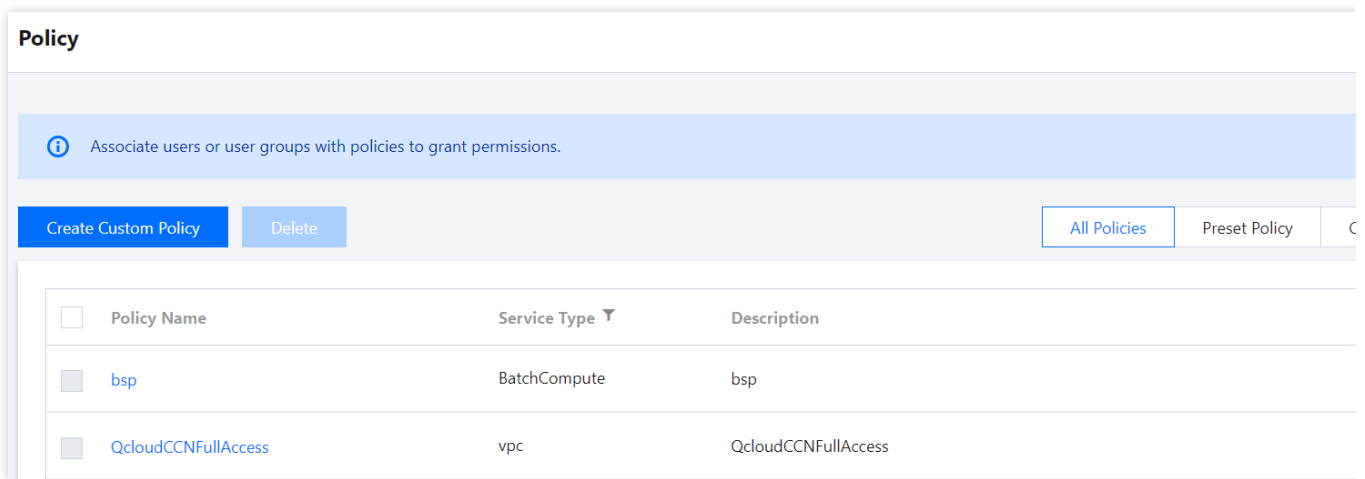
<input type="checkbox"/>	Policy Name	Service Type	Description	Last Modified
<input type="checkbox"/>	bsp	BatchCompute	bsp	2018-05-04 20:04:41
<input type="checkbox"/>	QcloudCCNFullAccess	vpc	QcloudCCNFullAccess	2019-10-09 19:58:06
<input type="checkbox"/>	FivAccess	fiv	FivAccess1	2020-01-06 15:35:35
<input type="checkbox"/>	CaptchaAccess	captcha	CaptchaAccess	2019-08-02 14:25:17
<input type="checkbox"/>	EMR_ADMIN	Elasticsearch MapReduce	EMR_ADMIN	2019-02-19 14:49:16
<input type="checkbox"/>	EMR_OBSERVER	Elasticsearch MapReduce	EMR_OBSERVER	2019-09-10 22:34:27
<input type="checkbox"/>	EMR_OPERATION	Elasticsearch MapReduce	EMR_OPERATION	2019-09-12 15:43:46
<input type="checkbox"/>	QcloudFCRFullAccess	Collection Robot	QcloudFCRFullAccess,Can add,modify,view and use all instances.	2019-05-08 15:16:18
<input type="checkbox"/>	FCR Administration	Collection Robot	Can add,modify,view and use all instances.	2019-04-08 16:26:54
<input type="checkbox"/>	FCR Visit and Access Instances	Collection Robot	Can view and use all instances.	2019-04-08 16:26:45

10 / page

创建自定义策略的方法可参考策略设置。

策略授权

已设置好的策略可以通过关联用户组或者子用户来授予权限。



自定义策略可授权资源类型

资源级权限指的是能够指定用户对哪些资源具有执行操作的能力。Prometheus 监控服务部分支持资源级权限，即表示针对支持资源级权限的 Prometheus 服务操作，您可以控制何时允许用户执行操作或是允许用户使用特定资源。访问管理 CAM 中可授权的资源类型如下：

资源类型	授权策略中的资源描述方法
Prometheus 监控服务	<pre>qcs::monitor:\$region:\$account:prom-instance/* qcs::monitor:\$region:\$account:prom-instance/\$instanceId</pre>

下表将介绍当前支持资源级权限的 Prometheus 监控服务 API 操作，设置策略时，action 填入 API 操作名称就可以对单独 API 进行控制，设置 action 也可以使用 * 作为通配符。

支持资源级授权的 API 列表

API 操作	API 描述
DescribePrometheusInstances	列出用户所有的 Prometheus 服务实例
TerminatePrometheusInstances	销毁 Prometheus 服务实例
RecreatePrometheusInstance	重新安装 Prometheus 服务实例
ModifyPrometheusInstanceAttributes	修改 Prometheus 实例相关属性
ChangeGrafanaAdminPassword	修改 Grafana Admin 密码
UpgradeGrafanaDashboard	升级 Grafana Dashboard
DescribePrometheusKubeClusters	列出 Prometheus 可集成的 Kubernetes 集群列表

InstallPrometheusAgent	安装 Prometheus Agent
UninstallPrometheusAgent	卸载 Prometheus Agent
DescribeServiceDiscovery	列出 Prometheus 服务发现列表
CreateServiceDiscovery	创建 Prometheus 服务发现
UpdateServiceDiscovery	更新 Prometheus 服务发现
DeleteServiceDiscovery	删除 Prometheus 服务发现
DescribePrometheusKubeBasicMonitor	查询基础监控状态
EnablePrometheusKubeBasicMonitor	开启基础监控
DisablePrometheusKubeBasicMonitor	关闭基础监控
DescribePrometheusAgentRuntime	获取 Prometheus Agent 运行时状态
DescribePrometheusJobTargets	列出 Prometheus 指标抓取任务的状态信息
DescribeRecordingRules	查询预聚合规则
CreateRecordingRule	创建预聚合规则
UpdateRecordingRule	更新预聚合规则
DeleteRecordingRules	删除预聚合规则
DescribeAlertRules	报警规则查询
DeleteAlertRules	删除报警规则
UpdateAlertRuleState	更新报警策略状态
CreateAlertRule	创建告警规则
UpdateAlertRule	更新报警规则

不支持资源级授权的 API 列表

针对不支持资源级权限的 Prometheus 监控服务 API 操作，您仍可以向用户授予使用该操作的权限，但策略语句的资源（resource）元素必须指定为 `*`。

API 操作	API 描述
CreatePrometheusInstance	创建 Prometheus 服务实例

相关角色权限说明

最近更新时间：2024-01-29 16:01:49

在使用 Prometheus 监控服务过程中，为了能够使用相关云资源，会遇到多种需要进行服务授权的场景。在使用该服务的过程中主要涉及 CM_QCSRole 服务角色。本文接下来将分角色展示各个授权策略的详情、授权场景及授权步骤。

CM_QCSRole 角色默认关联的预设策略包含如下：

QcloudAccessForCMRoleInPromHostingService：Prometheus 监控服务所需要的容器服务 TKE 权限。

场景

当您成功创建 Prometheus 服务实例之后，需要监控腾讯云容器服务（TKE）上运行的服务，为了可以更方便的集成容器服务，需要访问腾讯云容器服务（TKE）相关的 API 服务，需要您的授权委托，才能正常访问腾讯云容器服务（TKE）来安装基本的监控组件及获取对应监控组件的运行状态。

此角色无需主动寻找配置，在未授权的情况下，在您成功创建 Prometheus 服务实例后，进入到对应实例管理下的“集成容器服务”时会自动弹出授权界面。

授权步骤

主账号授权步骤

1. 当您成功创建 Prometheus 实例后，在访问“集成容器服务”时将出现授权提示框，需要授权云监控相关权限，如下所示。
2. 在子窗口中单击**前往授权**。
3. 在访问管理—角色管理页单击**同意授权**，提示授权成功即可。

说明：

此次授权只会出现一次，如果您已授权，则不再出现该授权提示框。

子账号授权步骤

主账号完成上述授权操作，成功创建了 CM_QCSRole 角色后，子账号无权限访问 CM_QCSRole 角色，需子账号对主账号授予 PassRole 权限，子账号才能在 Prometheus 监控服务中正常访问容器服务 TKE，否则访问容器服务列表时会提示失败。

在授予子账号 PassRole 权限时，请确保您的子账号有以下权限：

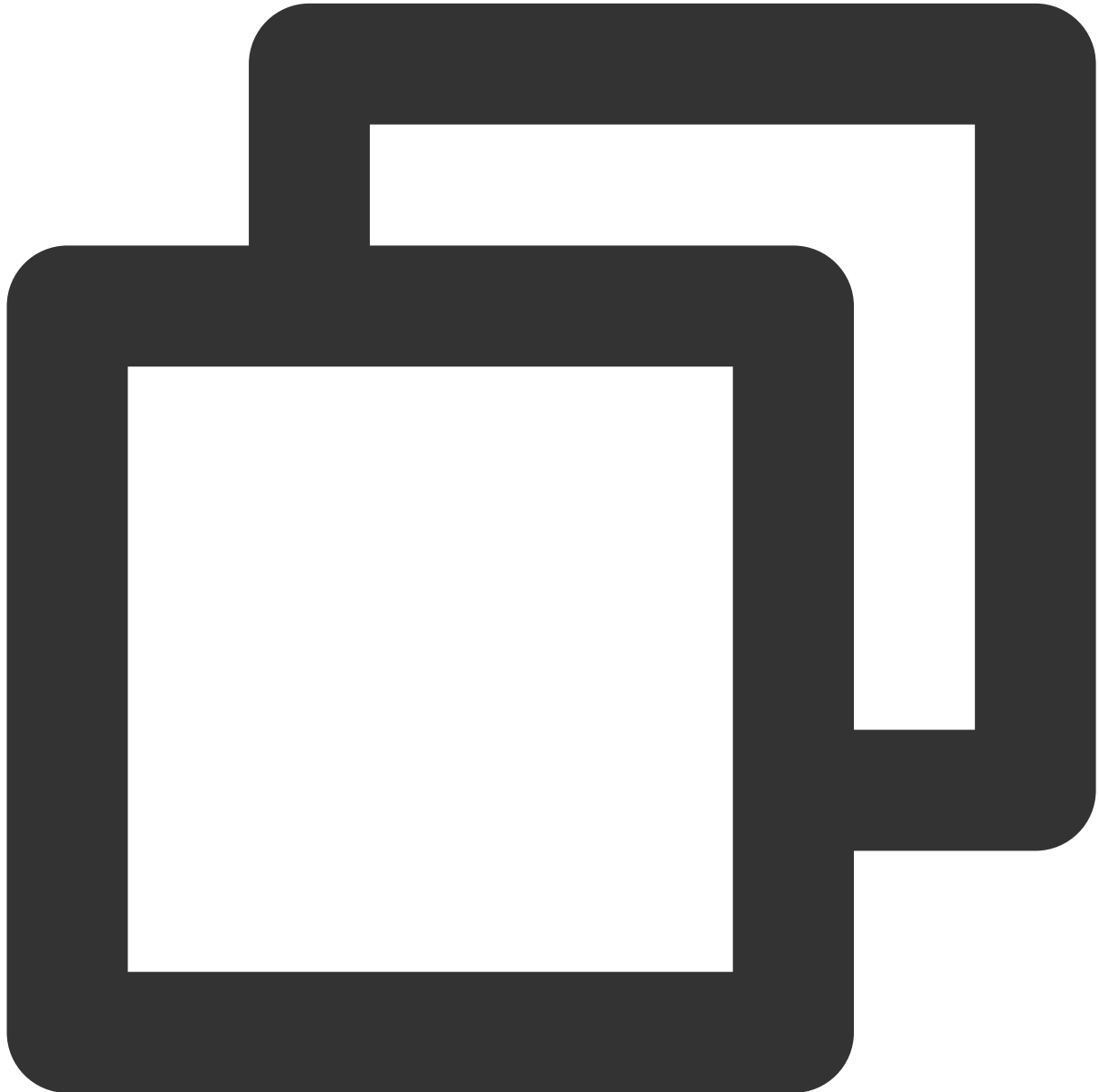
权限说明	授予策略
需授予子账号访问 CAM 权限，主账号授予子账号的 PassRole 权限才会生效	QcloudCamReadOnlyAccess 或 QcloudCamFullAcces

云监控策略依赖于云产品策略，因此授予子账号 PassRole 权限前，需确保子账号可在 TKE 下正常访问 TKE 资源

详情请参考 [容器服务（TKE）权限管理](#)

为确保上述权限授予成功，请参考以下步骤授予子账号 cam:PassRole 权限。

1. 使用主账号或具有管理权限的子账号创建如下自定义策略，策略语法如下：



```
{
  "version": "2.0",
  "statement": [
    {
```

```
    "effect": "allow",
    "action": "cam:PassRole",
    "resource": "qcs::cam::uin/${OwnerUin}:roleName/CM_QCSRole"
  }
]
}
```

2. 新建完后，参考 [访问管理-授权管理](#) 在自定义策略下关联子账号即可。

授予子账号 `cam:PassRole` 权限之后，再次访问对应 Prometheus 实例管理下的“集成容器服务”页面，将出现授权提示框。

Grafana

最近更新时间：2024-01-29 16:01:49

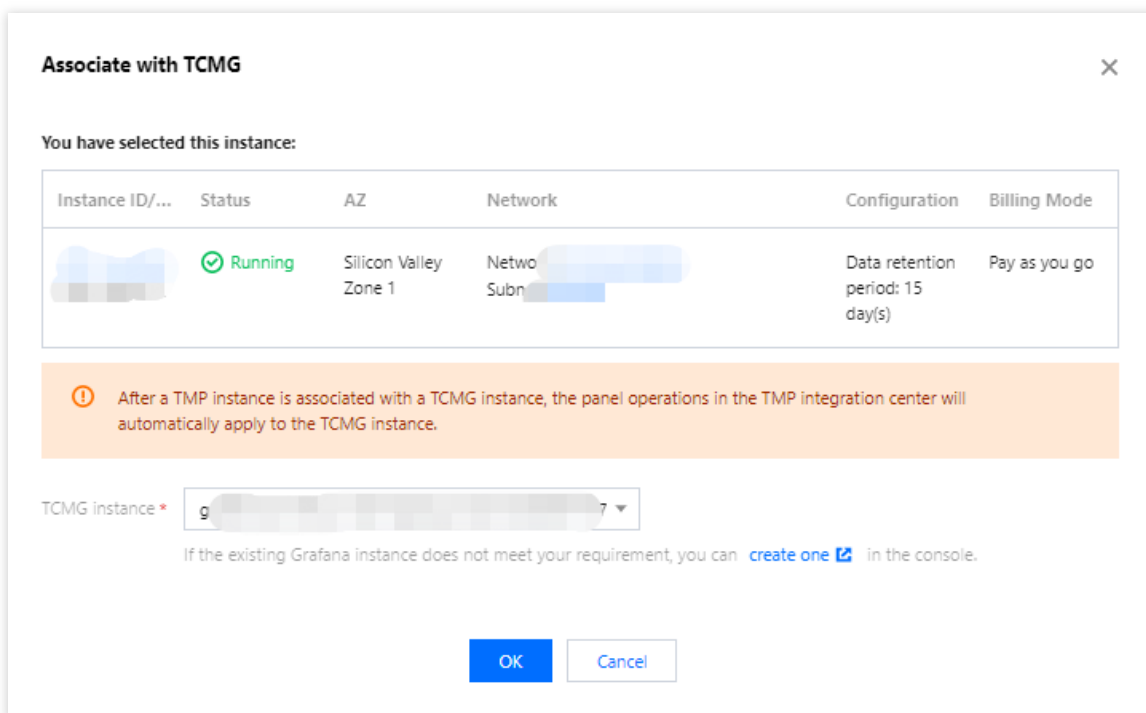
Prometheus 监控服务与 [Grafana 可视化服务](#) 高度集成，一个 Grafana 实例可同时被多个 Prometheus 实例绑定，用于实现 Prometheus 数据的统一可视化。

操作步骤

Prometheus 监控服务支持您在 [创建实例](#) 时绑定 Grafana 实例，若您在购买 Prometheus 时未绑定实例可参考下列操作指引绑定。

绑定 Grafana 实例

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在 Prometheus 列表中找到对应的 Prometheus 实例，在操作列单击 **更多>Grafana>绑定 Grafana**。
3. 在弹框中选择 Grafana 实例，并单击确定即可。

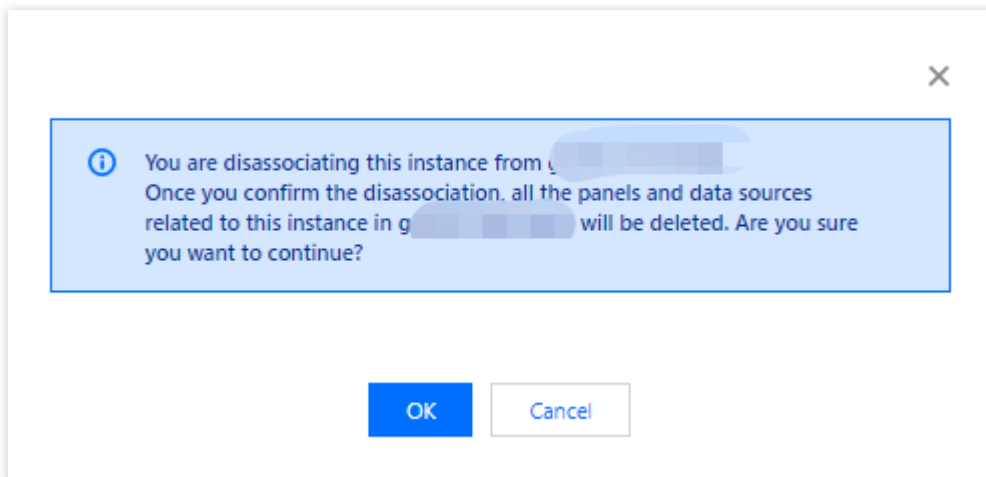


说明：

仅支持选择与 Prometheus 实例同一VPC（私有网络）的 Grafana 实例，若没有符合的 Grafana，请参考 [操作指引](#) 创建。

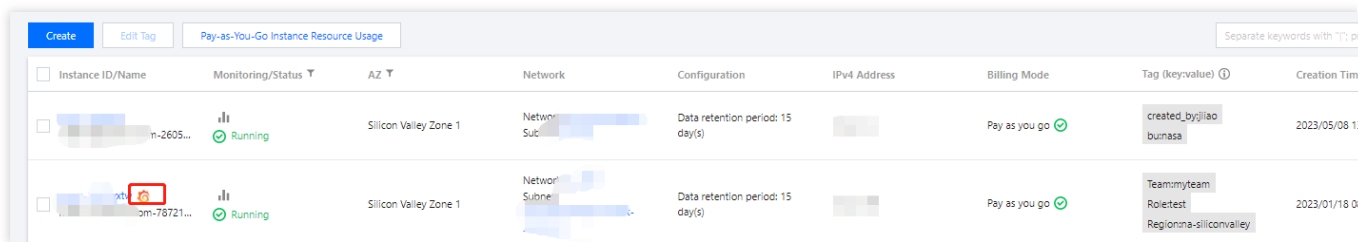
解绑 Grafana 实例

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在 Prometheus 列表中找到对应的 Prometheus 实例，在操作列单击**更多 > Grafana > 解绑 Grafana**。
3. 在弹框中确定解绑即可。



登录 Grafana 实例

1. 登录 [Prometheus 监控服务控制台](#)。
2. 在 Prometheus 列表中找到对应的 Prometheus 实例，单击实例 ID/名称右侧的 Grafana 图标。



Instance ID/Name	Monitoring/Status	AZ	Network	Configuration	IPv4 Address	Billing Mode	Tag (key:value)	Creation Time
[redacted]-2605...	Running	Silicon Valley Zone 1	Network Subnet [redacted]	Data retention period: 15 day(s)	[redacted]	Pay as you go	created_by@iao burnasa	2023/05/08 11:...
[redacted]-78721...	Running	Silicon Valley Zone 1	Network Subnet [redacted]	Data retention period: 15 day(s)	[redacted]	Pay as you go	Teamumy/team Roletest Regionna-siliconvalley	2023/01/18 0...

3. 进入 Grafana 界面输入账号和密码即可登录。

说明：

更多 Grafana 操作请参考 [Grafana 可视化服务](#)，如 Grafana 配置、图片渲染等操作。

API 使用指南

API 概览

最近更新时间：2024-01-29 16:01:49

HTTP API

Prometheus 所有稳定的 HTTP API 都在 `/api/v1` 路径下。当我们有数据查询需求时，可以通过查询 API 请求监控数据，提交数据可以使用 [remote write](#) 协议或者 [Pushgateway](#) 的方式。

支持的 API

API	说明	需要认证	方法
<code>/api/v1/query</code>	查询接口	是	GET/POST
<code>/api/v1/query_range</code>	范围查询	是	GET/POST
<code>/api/v1/series</code>	series 查询	是	GET/POST
<code>/api/v1/labels</code>	labels 查询	是	GET/POST
<code>/api/v1/label/<label_name>/values</code>	label value 查询	是	GET
<code>/api/v1/prom/write</code>	remote write 数据提交	是	remote write
Pushgateway	pushgateway 数据提交	是	SDK

认证方法

默认开启认证，因此所有的接口都需要认证，且所有的认证方式都支持 [Bearer Token](#)和 [Basic Auth](#)。

Bearer Token

Bearer Token 随着实例产生而生成，可以通过控制台进行查询。了解 Bearer Token 更多信息，请参见 [Bearer Authentication](#)。

Basic Auth

Basic Auth 兼容原生 Prometheus Query 的认证方式，用户名为用户的 APPID，密码为 bearer token（实例产生时生成），可以通过控制台进行查询。了解 Basic Auth 更多信息，请参见 [Basic Authentication](#)。

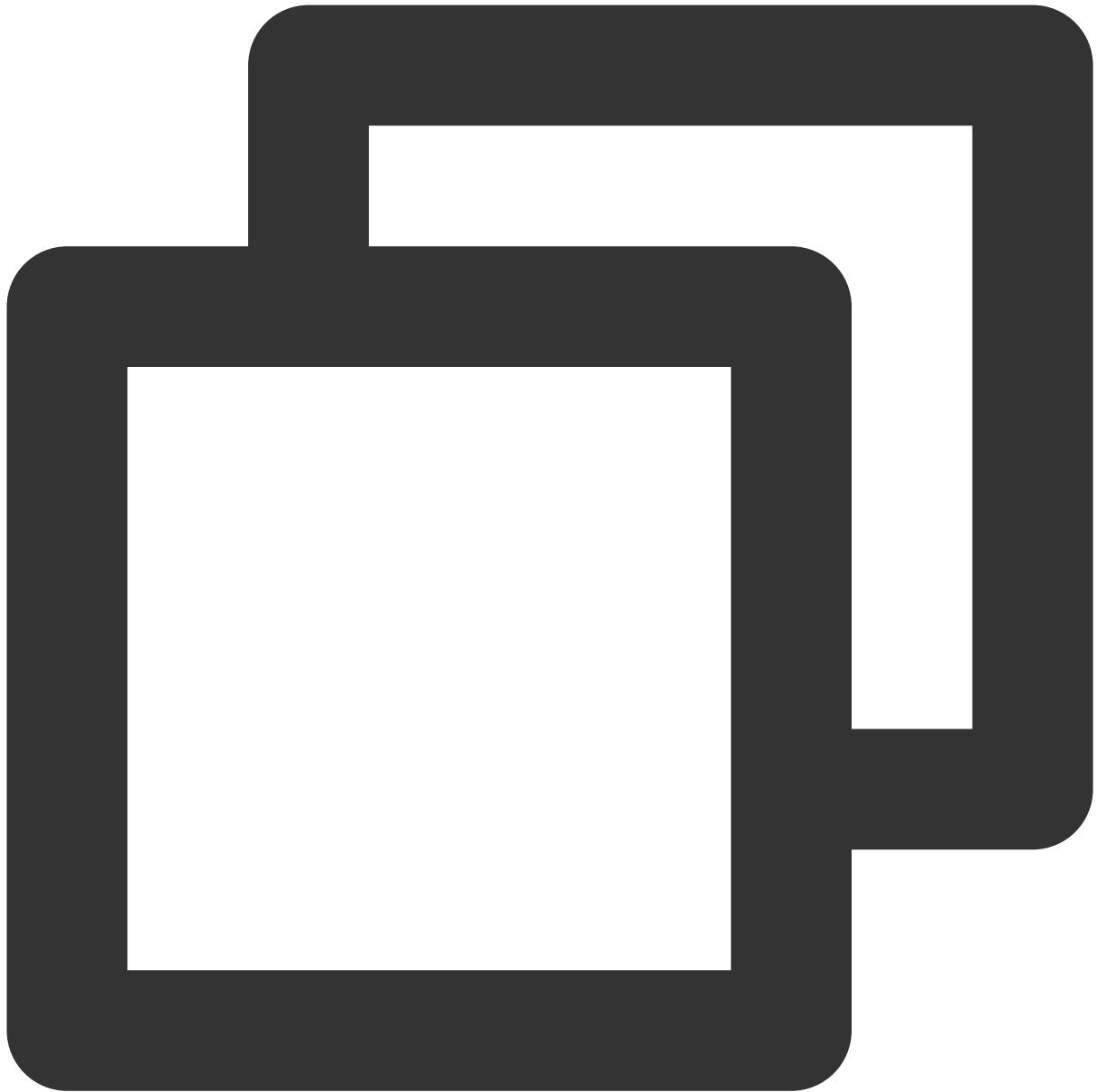
数据返回格式

所有 API 的响应数据格式都为 JSON。每一次成功的请求会返回 `2xx` 状态码。

无效的请求会返回一个包含错误对象的 JSON 格式数据，同时也将包含一个如下表格的状态码：

状态码	含义
401	认证失败
400	当参数缺失或错误时返回无效的请求状态码
422	当一个无效的表达式无法被指定时 (RFC4918)
503	当查询不可用或者被取消时返回服务不可用状态码

无效请求响应返回模板如下：



```
{
  "status": "success" | "error",
  "data": <data>,

  // 当 status 状态为 error 时, 下面的数据将被返回
  "errorType": "<string>",
  "error": "<string>",

  // 当执行请求时有警告信息时, 该字段将被填充返回
  "warnings": ["<string>"]
}
```

数据写入

最近更新时间：2024-01-29 16:01:49

操作场景

类似 flink job 上报数据的场景，我们需要通过 API 直接将数据写入 Prometheus，因为这些 job 的生命周期可能会很短，来不及等待 Prometheus 来拉取数据。写入数据可以直接使用 [Remote Write](#) 协议或者 [Pushgateway](#) 的方式。

Remote Write



```
POST /api/v1/prom/write
```

Remote Write 是 Prometheus 标准的协议，更多介绍可以参见 [Prometheus-remote write](#)。使用 remote write 我们可以把 VPC 内其他 Prometheus 的数据写入到 Prometheus 监控服务中，对于数据的稳定性提升和迁移，都不失为一种不错的方案。

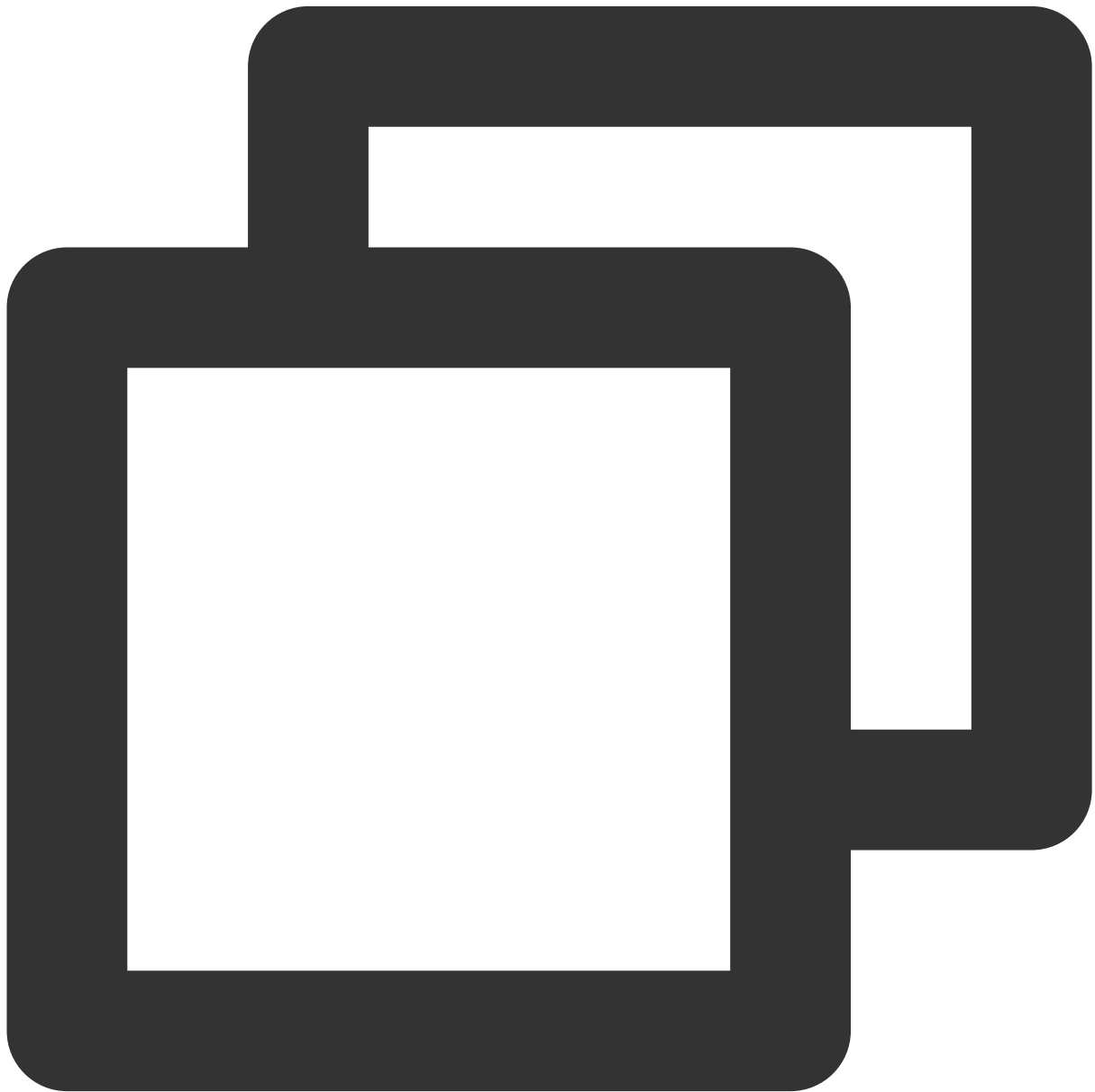
Pushgateway

虽然 Prometheus 场景主张以拉为主，但是有些场景我们仍然需要使用推的方式，可以参见 [Prometheus - Pushgateway 说明](#)。

Prometheus 监控服务原生集成了 Pushgateway 模块，可以直接推送数据到服务中。下面是一个使用 Go 语言推送数据的简单的例子，需要将 `$IP`、`$PORT`、`$APPID`、`$TOKEN` 这些变量改为自己实例的认证链接信息，这些链接信息可以从控制台进行查询。

说明：

强烈建议替换 `$INSTANCE` 为当前机器的标识，例如 `IP/Hostname`。如果没有设置这个 `groupingKey`，当多台机器一起上报时，数据会互相覆盖。



```
package main
```

```
import (
    "fmt"
    "time"

    "github.com/prometheus/client_golang/prometheus"
    "github.com/prometheus/client_golang/prometheus/push"
    "github.com/prometheus/common/expfmt"
)

var completionTime = prometheus.NewGauge(prometheus.GaugeOpts{
    Name: "db_backup_last_completion_timestamp_seconds",
    Help: "The timestamp of the last successful completion of a DB backup.",
})

func do() {
    completionTime.SetToCurrentTime()
}

func ExamplePusher_Push() {
    if err := push.New("http://$IP:$PORT", "db_backup").
        BasicAuth("$APPID", "$TOKEN").
        Collector(completionTime).
        Grouping("instance", "$INSTANCE").
        Grouping("db", "customers").
        Format(expfmt.FmtText).
        Push(); err != nil {
        fmt.Println("Could not push completion time to Pushgateway:", err)
    }
}

func main() {
    do()
    ticker := time.NewTicker(2 * time.Second)
    done := make(chan bool)
    for {
        select {
        case <-done:
            return
        case <-ticker.C:
            ExamplePusher_Push()
        }
    }
}
```

注意：

`push.New` 生成的对象可以通过 `Client` 方法自定义 HTTP Client，我们推荐设置一个合适的超时时间，同时 `push` 数据我们建议使用异步的方式调用，以免阻塞业务主流程。

其它上报场景请参见 [自定义监控说明](#)。

监控数据查询

最近更新时间：2024-01-29 16:01:49

操作场景

当我们有数据查询需求时，可以通过查询 API 请求监控数据。

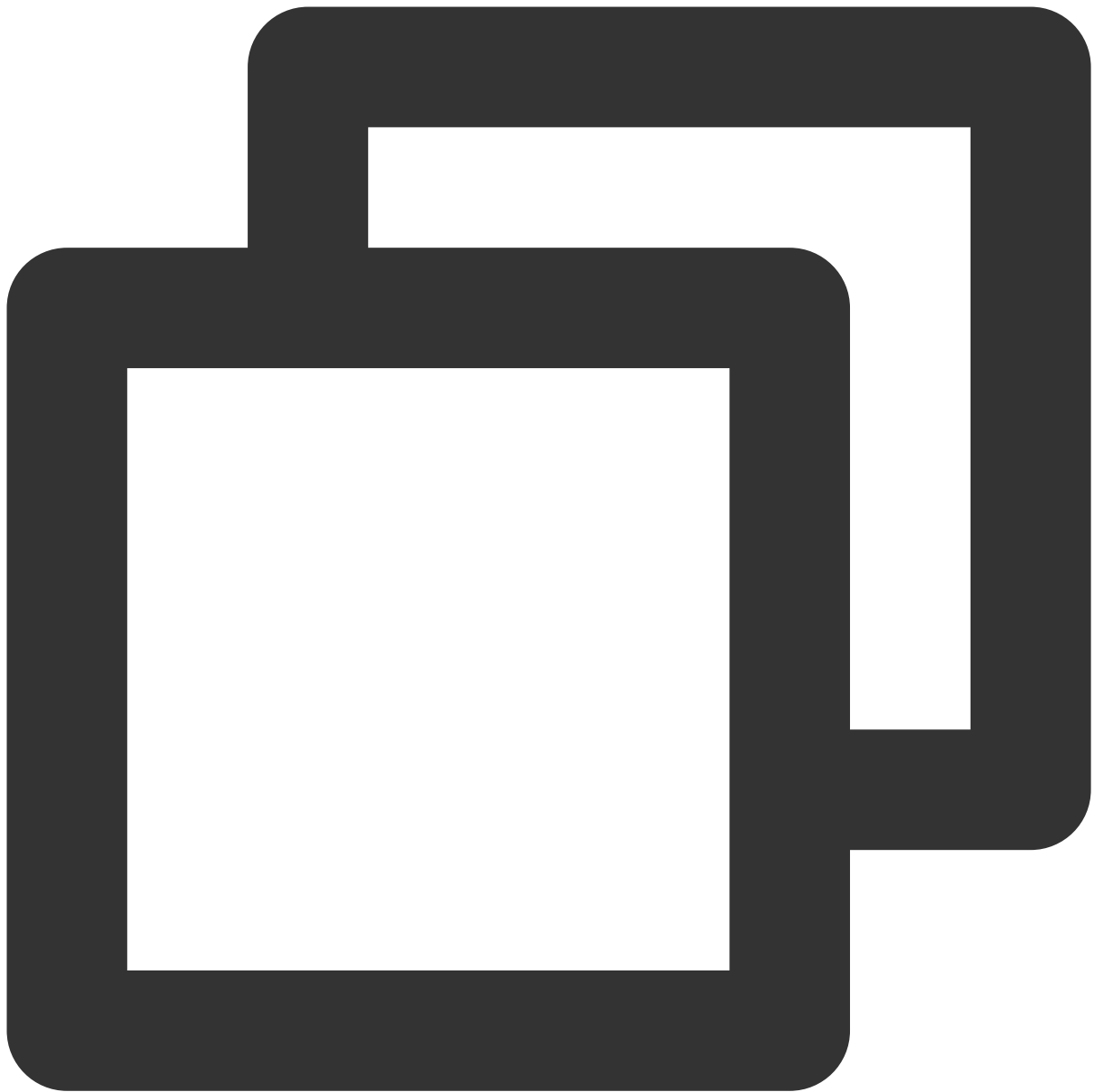
APPID/Token 获取方式

托管 Prometheus API 使用需要通过 APPID + Token 的方式进行鉴权访问。

[APPID 获取地址](#)。

Token 从对应 Prometheus 实例的基本信息中获取。

查询 API 接口



```
GET /api/v1/query  
POST /api/v1/query
```

查询参数

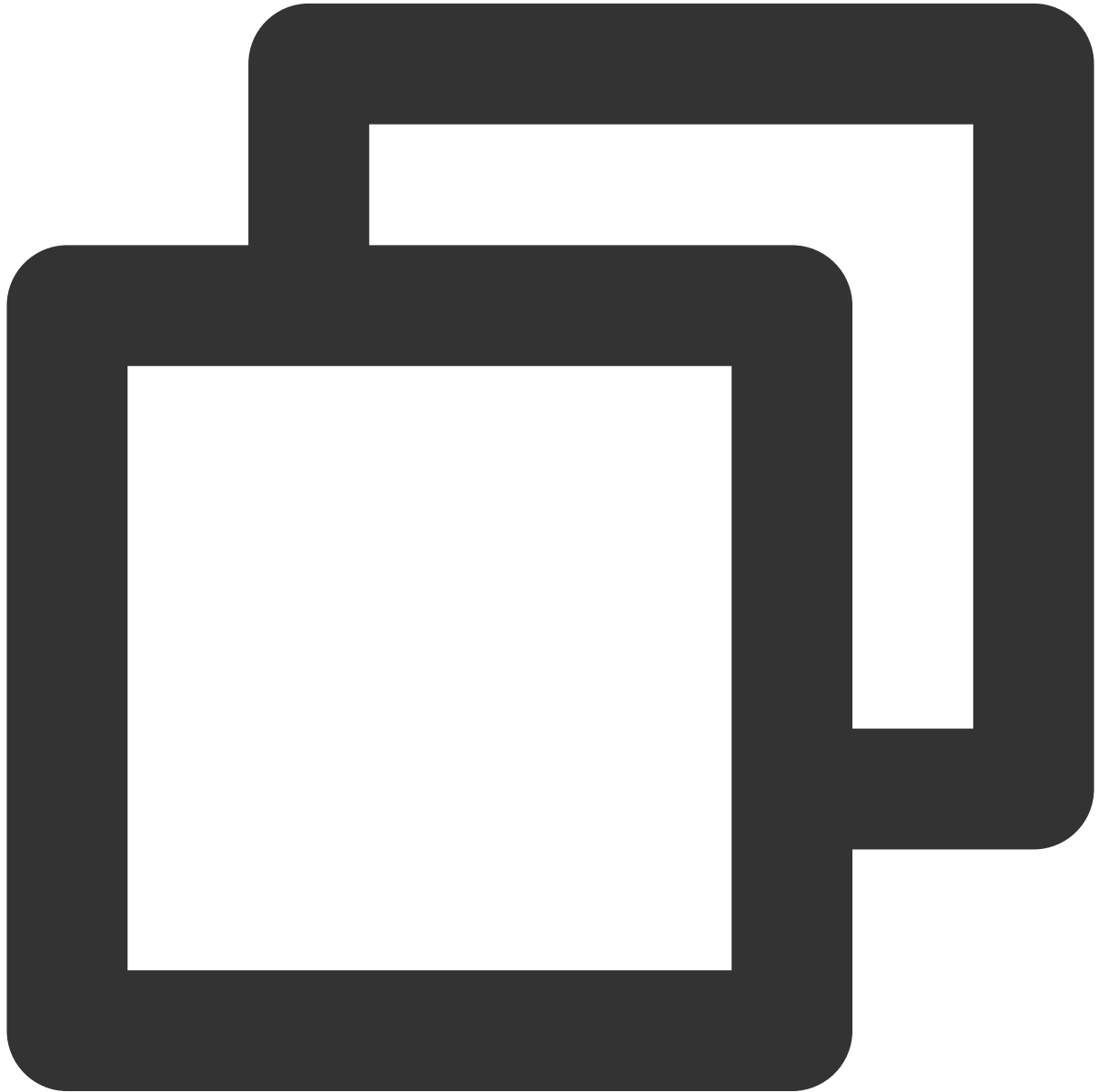
query=<string>: Prometheus : 查询表达式。

time=<rfc3339 | unix_timestamp> : 时间戳, 可选。

timeout=<duration> : 检测超时时间, 可选。默认由 `-query.timeout` 参数指定。

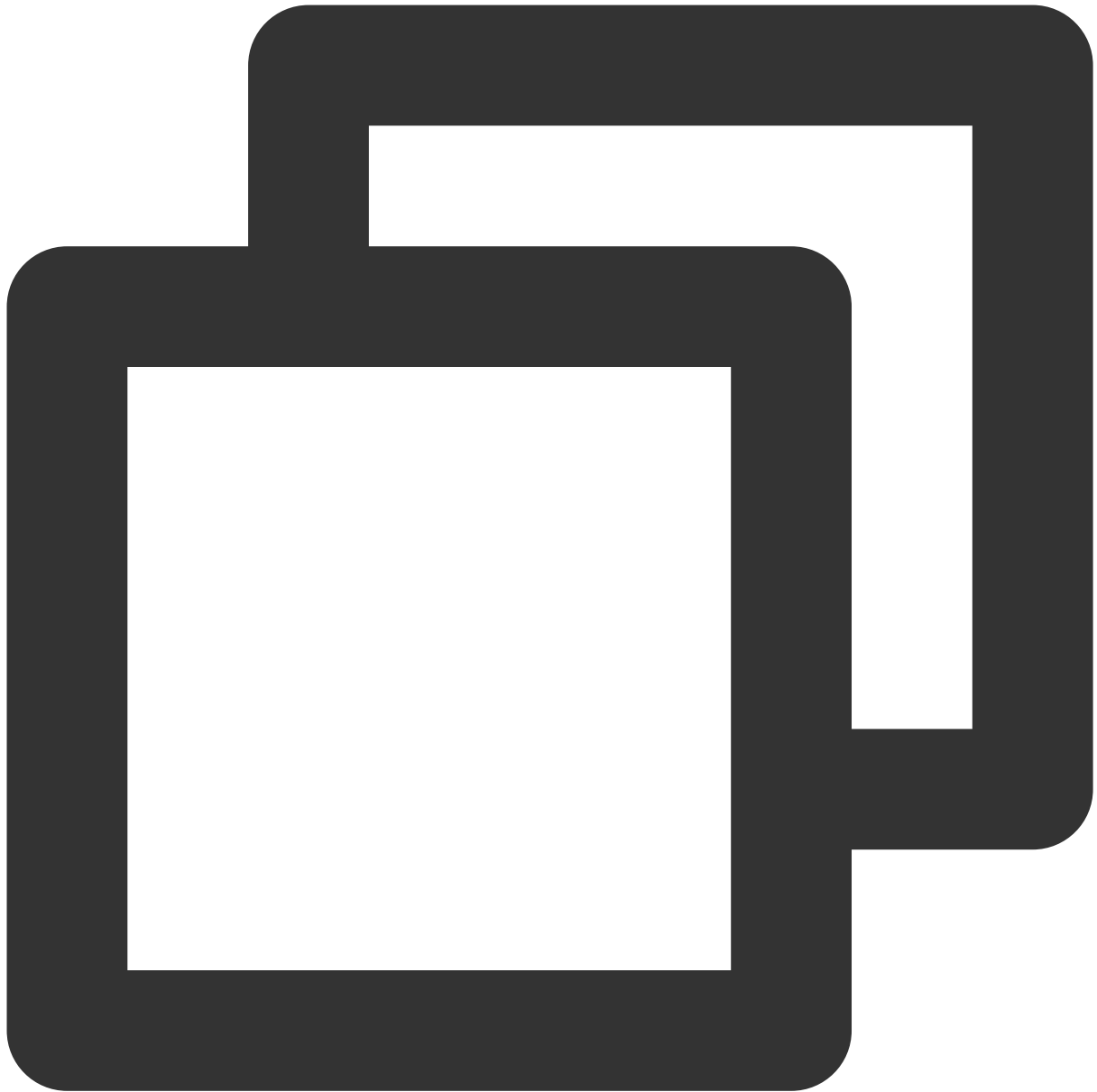
简单查询示例

我们可以使用下面的例子进行 API 数据查询, 查询服务地址和认证信息可以在相应实例控制台查看:



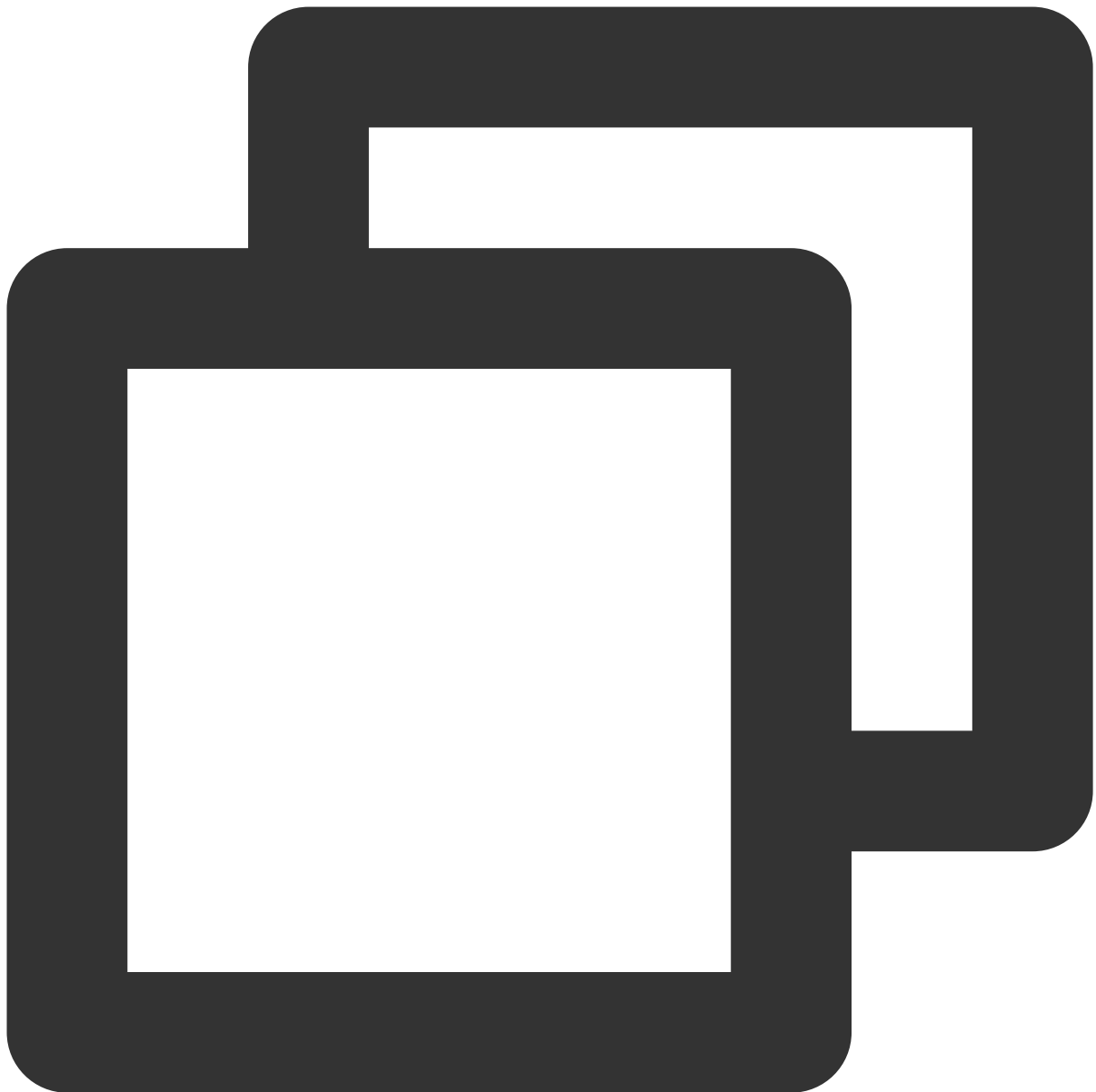
```
curl -u "appid:token" 'http://IP:PORT/api/v1/query?query=up'
```

如果返回状态码为 401 请检查认证信息是否正确。



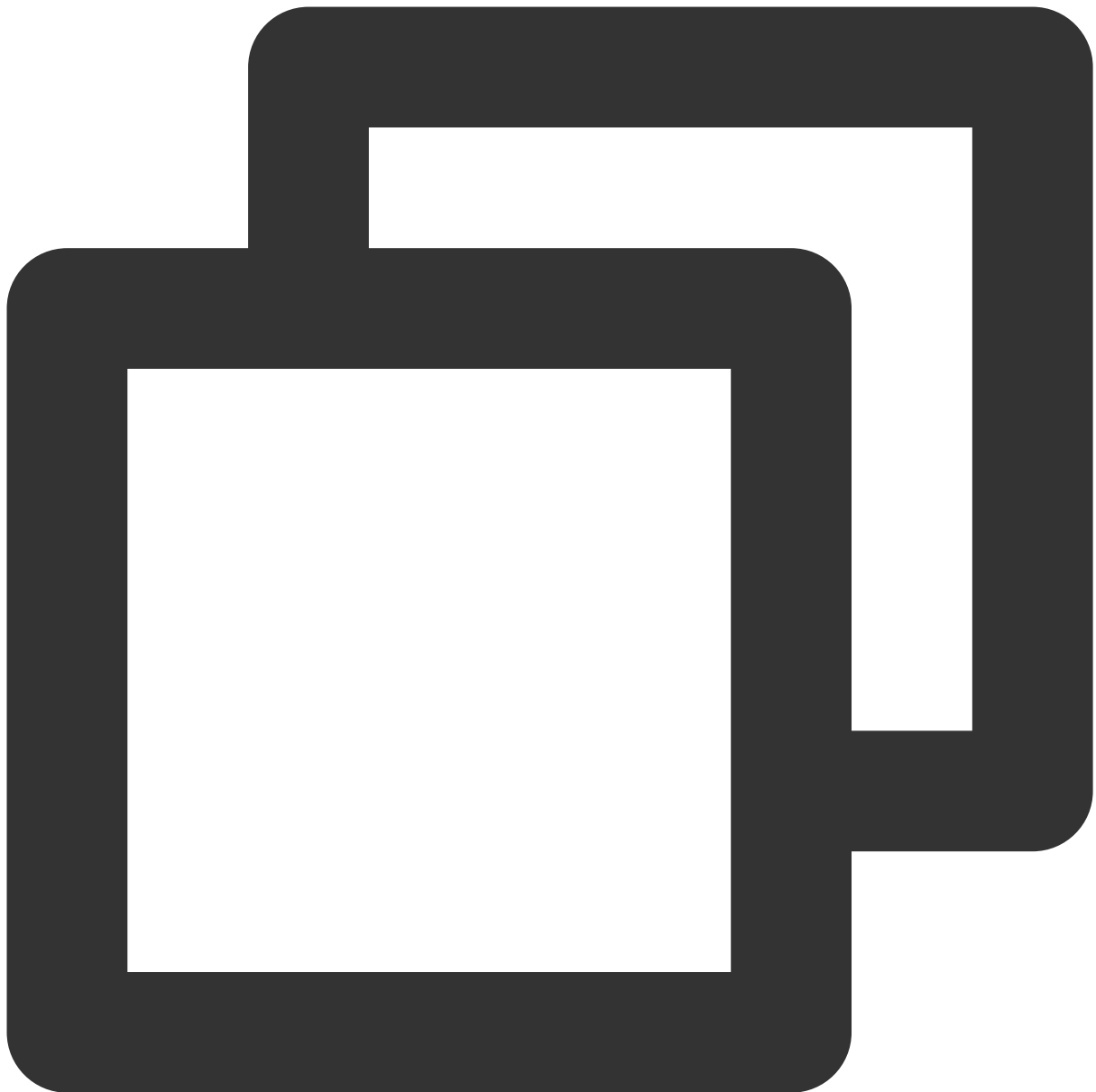
```
< HTTP/1.1 401 Unauthorized  
< Content-Length: 0
```

范围查询



```
GET /api/v1/query_range  
POST /api/v1/query_range
```

根据时间范围查询我们需要的数据是我们面临的最多的场景，这时我们我们需要用到 `/api/v1/query_range` 接口，示例如下：



```
$ curl -u "appid:token" 'http://IP:PORT/api/v1/query_range?query=up&start=2015-07-0'
{
  "status" : "success",
  "data" : {
    "resultType" : "matrix",
    "result" : [
      {
        "metric" : {
          "__name__" : "up",
          "job" : "prometheus",
          "instance" : "localhost:9090"
        }
      }
    ]
  }
}
```

```
    },
    "values" : [
      [ 1435781430.781, "1" ],
      [ 1435781445.781, "1" ],
      [ 1435781460.781, "1" ]
    ]
  },
  {
    "metric" : {
      "__name__" : "up",
      "job" : "node",
      "instance" : "localhost:9091"
    },
    "values" : [
      [ 1435781430.781, "0" ],
      [ 1435781445.781, "0" ],
      [ 1435781460.781, "1" ]
    ]
  }
]
}
```

自建 Grafana 添加数据源

我们可以通过自己部署的 Grafana 添加托管的 Prometheus 为数据源，方便我们在自己的 Grafana 中查看数据，前提是需要保证它们在同一 VPC 内，保证网络是可以互相访问的。

开启 BasicAuth 认证方法，并填写相应的认证信息即可，如下图配置。

The screenshot shows the configuration interface for a Prometheus data source named 'hosted-prometheus'. The page is divided into sections for general settings, HTTP, and authentication (Auth). The 'Auth' section is highlighted with a red box, showing that 'Basic auth' is enabled. Below it, the 'Basic Auth Details' section is also highlighted, showing the 'User' field set to 'APPID' and the 'Password' field set to 'configured', with a 'Reset' button.

Data Sources / hosted-prometheus
Type: Prometheus

Settings | Dashboards

Name: hosted-prometheus Default

HTTP

URL: http://IP:PORT

Access: Server (default) Help >

Whitelisted Cookies: Add Name Add

Auth

Basic auth With Credentials

TLS Client Auth With CA Cert

Skip TLS Verify

Forward OAuth Identity

Basic Auth Details

User: APPID

Password: configured Reset

容器服务指标

按量付费免费指标

最近更新时间：2024-01-29 16:01:49

按量付费模式部分指标基础免费存储15天，存储时长超过15天的实例，将按照超出的天数，收取免费指标的存储费用。

所属配置文件	指标名
node-exporter	node_boot_time_seconds
node-exporter	node_context_switches_total
node-exporter	node_cpu_seconds_total
node-exporter	node_disk_io_now
node-exporter	node_disk_io_time_seconds_total
node-exporter	node_disk_io_time_weighted_seconds_total
node-exporter	node_disk_read_bytes_total
node-exporter	node_disk_read_time_seconds_total
node-exporter	node_disk_reads_completed_total
node-exporter	node_disk_write_time_seconds_total
node-exporter	node_disk_writes_completed_total
node-exporter	node_disk_written_bytes_total
node-exporter	node_filefd_allocated
node-exporter	node_filesystem_avail_bytes
node-exporter	node_filesystem_free_bytes
node-exporter	node_filesystem_size_bytes
node-exporter	node_load1
node-exporter	node_load15
node-exporter	node_load5

node-exporter	node_memory_Buffers_bytes
node-exporter	node_memory_Cached_bytes
node-exporter	node_memory_MemAvailable_bytes
node-exporter	node_memory_MemFree_bytes
node-exporter	node_memory_MemTotal_bytes
node-exporter	node_netstat_TcpExt_ListenDrops
node-exporter	node_netstat_Tcp_ActiveOpens
node-exporter	node_netstat_Tcp_CurrEstab
node-exporter	node_netstat_Tcp_InSegs
node-exporter	node_netstat_Tcp_OutSegs
node-exporter	node_netstat_Tcp_PassiveOpens
node-exporter	node_network_receive_bytes_total
node-exporter	node_network_transmit_bytes_total
node-exporter	node_sockstat_TCP_alloc
node-exporter	node_sockstat_TCP_inuse
node-exporter	node_sockstat_TCP_tw
node-exporter	node_sockstat_UDP_inuse
node-exporter	node_sockstat_sockets_used
node-exporter	node_uname_info
cadvisor	container_cpu_usage_seconds_total
cadvisor	container_fs_limit_bytes
cadvisor	container_fs_reads_bytes_total
cadvisor	container_fs_usage_bytes
cadvisor	container_fs_writes_bytes_total
cadvisor	container_memory_working_set_bytes

cadvisor	container_network_receive_bytes_total
cadvisor	container_network_receive_packets_dropped_total
cadvisor	container_network_receive_packets_total
cadvisor	container_network_transmit_bytes_total
cadvisor	container_network_transmit_packets_dropped_total
cadvisor	container_network_transmit_packets_total
cadvisor	machine_cpu_cores
cadvisor	machine_memory_bytes
kubelet	kubelet_cgroup_manager_duration_seconds_count
kubelet	kubelet_node_config_error
kubelet	kubelet_node_name
kubelet	kubelet_pleg_relist_duration_seconds_bucket
kubelet	kubelet_pleg_relist_duration_seconds_count
kubelet	kubelet_pleg_relist_interval_seconds_bucket
kubelet	kubelet_pod_start_duration_seconds_count
kubelet	kubelet_pod_worker_duration_seconds_count
kubelet	kubelet_running_containers
kubelet	kubelet_running_pods
kubelet	kubelet_runtime_operations_duration_seconds_bucket
kubelet	kubelet_runtime_operations_errors_total
kubelet	kubelet_runtime_operations_total
kubelet	process_cpu_seconds_total
kubelet	process_resident_memory_bytes
kubelet	rest_client_request_duration_seconds_bucket
kubelet	rest_client_requests_total

kubelet	storage_operation_duration_seconds_bucket
kubelet	storage_operation_duration_seconds_count
kubelet	storage_operation_errors_total
kubelet	volume_manager_total_volumes
kube-state-metrics	kube_job_status_succeeded
kube-state-metrics	kube_job_status_failed
kube-state-metrics	kube_job_status_active
kube-state-metrics	kube_node_status_capacity_cpu_cores
kube-state-metrics	kube_node_status_capacity_memory_bytes
kube-state-metrics	kube_node_status_allocatable_cpu_cores
kube-state-metrics	kube_node_status_allocatable_memory_bytes
kube-state-metrics	kube_pod_info
kube-state-metrics	kube_pod_owner
kube-state-metrics	kube_pod_status_phase
kube-state-metrics	kube_pod_container_status_waiting
kube-state-metrics	kube_pod_container_status_running
kube-state-metrics	kube_pod_container_status_terminated
kube-state-metrics	kube_pod_container_status_restarts_total
kube-state-metrics	kube_pod_container_resource_requests_cpu_cores
kube-state-metrics	kube_pod_container_resource_requests_memory_bytes
kube-state-metrics	kube_pod_container_resource_limits_cpu_cores
kube-state-metrics	kube_pod_container_resource_limits_memory_bytes
kube-state-metrics	kube_replicaset_owner
kube-state-metrics	kube_statefulset_status_replicas
kube-controller-manager	rest_client_request_duration_seconds_bucket

kube-controller-manager	rest_client_requests_total
kube-controller-manager	workqueue_adds_total
kube-controller-manager	workqueue_depth
kube-controller-manager	workqueue_queue_duration_seconds_bucket
kube-apiserver	apiserver_current_inflight_requests
kube-apiserver	apiserver_current_inqueue_requests
kube-apiserver	apiserver_init_events_total
kube-apiserver	apiserver_longrunning_gauge
kube-apiserver	apiserver_registered_watchers
kube-apiserver	apiserver_request_duration_seconds_bucket
kube-apiserver	apiserver_request_duration_seconds_sum
kube-apiserver	apiserver_request_duration_seconds_count
kube-apiserver	apiserver_request_filter_duration_seconds_bucket
kube-apiserver	apiserver_request_filter_duration_seconds_sum
kube-apiserver	apiserver_request_filter_duration_seconds_count
kube-apiserver	apiserver_request_total
kube-apiserver	apiserver_requested_deprecated_apis
kube-apiserver	apiserver_response_sizes_bucket
kube-apiserver	apiserver_response_sizes_sum
kube-apiserver	apiserver_response_sizes_count
kube-apiserver	apiserver_selfrequest_total
kube-apiserver	apiserver_tls_handshake_errors_total
kube-apiserver	apiserver_watch_events_sizes
kube-apiserver	apiserver_watch_events_sizes_bucket
kube-apiserver	apiserver_watch_events_sizes_sum

kube-apiserver	apiserver_watch_events_sizes_count
kube-apiserver	apiserver_watch_events_total

容器常用指标推荐

最近更新时间：2024-01-29 16:01:49

基于大多数用户使用情况，专家建议配置如下常用的容器指标：

注意：

以下指标都是付费指标，指标的计费方式请参见 [相关计费说明](#)。

所属配置文件	指标名	指标含义
kubelet	kubelet_running_container_count	kubelet_running_container_count Number of containers currently running
kubelet	kubelet_running_pod_count	kubelet_running_pod_count Number of pods currently running
kube-state-metrics	kube_pod_container_info	Information about a container in pod.
kube-state-metrics	kube_deployment_status_replicas	The number of replicas per deployment.
kube-state-metrics	kube_deployment_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_pod_start_time	Start time in unix timestamp for pod.
kube-state-metrics	kube_pod_status_ready	Describes whether the pod is ready to serve requests.
kube-state-metrics	kube_node_info	Information about a cluster node
kube-state-metrics	kube_node_status_condition	The condition of a cluster node.
kube-	kube_deployment_status_replicas_updated	The number of updated replicas

state-metrics		per deployment.
kube-state-metrics	kube_deployment_status_replicas_available	The number of available replicas per deployment.
kube-state-metrics	kube_node_status_capacity_pods	The total pod resources of the node.
kube-state-metrics	kube_pod_container_status_ready	Describes whether the container readiness check succeeded.
kube-state-metrics	kube_deployment_spec_replicas	Number of desired pods for a deployment.
kube-state-metrics	kube_pod_status_scheduled_time	Unix timestamp when pod move into scheduled status
kube-state-metrics	kube_node_status_allocatable_pods	The pod resources of a node that are available for scheduling.
kube-state-metrics	kube_pod_container_resource_limits	The number of requested limit resource by a container.
kube-state-metrics	node_filefd_maximum	File descriptor statistics: maximum.
kube-state-metrics	kube_pod_container_resource_requests	The number of requested request resource by a container.
kube-state-metrics	kube_namespace_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_deployment_status_replicas_unavailable	The number of unavailable replicas per deployment.
kube-	kube_pod_created	Unix creation timestamp

state-metrics		
kube-state-metrics	kube_pod_container_status_waiting_reason	Describes the reason the container is currently in waiting state.
kube-state-metrics	kube_daemonset_status_desired_number_scheduled	The number of nodes that should be running the daemon pod.
kube-state-metrics	kube_pod_restart_policy	Describes the restart policy in use by this pod.
kube-state-metrics	kube_deployment_metadata_generation	Sequence number representing specific generation of the desired state.
kube-state-metrics	kube_statefulset_status_update_revision	Indicates the version of the StatefulSet used to generate Pods in the sequence [replicas-updatedReplicas
kube-state-metrics	kube_node_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_statefulset_replicas	Number of desired pods for a StatefulSet.
kube-state-metrics	kube_statefulset_status_observed_generation	The generation observed by the StatefulSet controller.
kube-state-metrics	kube_pod_container_status_last_terminated_reason	Describes the last reason the container was in terminated state.
kube-state-metrics	kube_replicaset_spec_replicas	Number of desired pods for a ReplicaSet.
kube-state-metrics	kube_statefulset_created	Unix creation timestamp

kube-state-metrics	kube_statefulset_status_replicas_current	The number of current replicas per StatefulSet.
kube-state-metrics	kube_statefulset_status_current_revision	Indicates the version of the StatefulSet used to generate Pods in the sequence [0
kube-state-metrics	kube_statefulset_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_deployment_created	Unix creation timestamp
kube-state-metrics	kube_namespace_created	Unix creation timestamp
kube-state-metrics	kube_daemonset_status_number_ready	The number of nodes that should be running the daemon pod and have one or more of the daemor pod running and ready.
kube-state-metrics	kube_deployment_status_observed_generation	The generation observed by the deployment controller.
kube-state-metrics	kube_endpoint_info	Information about endpoint.
kube-state-metrics	kube_statefulset_status_replicas_updated	The number of updated replicas per StatefulSet.
kube-state-metrics	kube_statefulset_metadata_generation	Sequence number representing specific generation of the desire state for the StatefulSet.
kube-state-metrics	kube_secret_created	Unix creation timestamp
kube-state-	kube_endpoint_address_not_ready	Number of addresses not ready in endpoint

metrics		
kube-state-metrics	kube_secret_type	Type about secret.
kube-state-metrics	kube_deployment_spec_paused	Whether the deployment is paused and will not be processed by the deployment controller.
kube-state-metrics	kube_pod_container_status_terminated_reason	Describes the reason the container is currently in terminated state.
kube-state-metrics	kube_statefulset_status_replicas_ready	The number of ready replicas per StatefulSet.
kube-state-metrics	kube_endpoint_address_available	Number of addresses available endpoint.
kube-state-metrics	kube_secret_info	Information about secret.
kube-state-metrics	kube_service_info	Information about service.
kube-state-metrics	kube_node_status_allocatable	The allocatable for different resources of a node that are available for scheduling.
kube-state-metrics	kube_endpoint_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_deployment_status_condition	The current status conditions of deployment.
kube-state-metrics	kube_endpoint_created	Unix creation timestamp
kube-state-	kube_replicaset_labels	Kubernetes labels converted to Prometheus labels.

metrics		
kube-state-metrics	kube_replicaset_metadata_generation	Sequence number representing specific generation of the desired state.
kube-state-metrics	kube_namespace_status_phase	kubernetes namespace status phase.
kube-state-metrics	kube_service_created	Unix creation timestamp
kube-state-metrics	kube_configmap_created	Unix creation timestamp
kube-state-metrics	kube_secret_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_deployment_spec_strategy_rollingupdate_max_surge	Maximum number of replicas that can be scheduled above the desired number of replicas during a rolling update of a deployment
kube-state-metrics	kube_configmap_metadata_resource_version	Resource version representing a specific version of the configmap
kube-state-metrics	kube_pod_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_replicaset_status_replicas	The number of replicas per ReplicaSet.
kube-state-metrics	kube_node_created	Unix creation timestamp
kube-state-metrics	kube_service_spec_type	Type about service.
kube-	kube_secret_metadata_resource_version	Resource version representing a

state-metrics		specific version of secret.
kube-state-metrics	kube_configmap_info	Information about configmap.
kube-state-metrics	kube_replicaset_status_observed_generation	The generation observed by the ReplicaSet controller.
kube-state-metrics	kube_service_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_replicaset_created	Unix creation timestamp
kube-state-metrics	kube_deployment_spec_strategy_rollingupdate_max_unavailable	Maximum number of unavailable replicas during a rolling update of a deployment.
kube-state-metrics	kube_replicaset_status_ready_replicas	The number of ready replicas per ReplicaSet.
kube-state-metrics	kube_replicaset_status_fully_labeled_replicas	The number of fully labeled replicas per ReplicaSet.
kube-state-metrics	kube_pod_status_scheduled	Describes the status of the scheduling process for the pod.
kube-state-metrics	kube_storageclass_created	Unix creation timestamp
kube-state-metrics	kube_daemonset_status_number_misscheduled	The number of nodes running a daemon pod but are not supposed to.
kube-state-metrics	kube_storageclass_labels	Kubernetes labels converted to Prometheus labels.
kube-	kube_node_status_capacity	The capacity for different

state-metrics		resources of a node.
kube-state-metrics	kube_daemonset_status_current_number_scheduled	The number of nodes running at least one daemon pod and are supposed to.
kube-state-metrics	kube_storageclass_info	Information about storageclass.
kube-state-metrics	kube_node_spec_unschedulable	Whether a node can schedule new pods.
kube-state-metrics	kube_daemonset_status_number_available	The number of nodes that should be running the daemon pod and have one or more of the daemon pod running and available
kube-state-metrics	kube_daemonset_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_daemonset_created	Unix creation timestamp
kube-state-metrics	kube_daemonset_status_number_unavailable	The number of nodes that should be running the daemon pod and have none of the daemon pod running and available
kube-state-metrics	kube_daemonset_metadata_generation	Sequence number representing specific generation of the desired state.
kube-state-metrics	kube_mutatingwebhookconfiguration_info	Information about the MutatingWebhookConfiguration
kube-state-metrics	kube_mutatingwebhookconfiguration_created	Unix creation timestamp.
kube-state-	kube_mutatingwebhookconfiguration_metadata_resource_version	Resource version representing a specific version of the

metrics		MutatingWebhookConfiguration
kube-state-metrics	kube_daemonset_updated_number_scheduled	The total number of nodes that are running updated daemon po
node-exporter	node_filesystem_files_free	Filesystem total free file nodes.
node-exporter	node_filesystem_files	Filesystem total file nodes.
node-exporter	node_sockstat_UDP_mem_bytes	Number of UDP sockets in state mem_bytes.
node-exporter	node_nf_contrack_entries_limit	Maximum size of connection tracking table.
node-exporter	node_memory_Shmem_bytes	Memory information field Shmem_bytes.
node-exporter	node_netstat_Tcp_RetransSegs	Statistic TcpRetransSegs.
node-exporter	node_sockstat_TCP_mem_bytes	Number of TCP sockets in state mem_bytes.
node-exporter	node_network_info	Non-numeric data from /sys/class/net/<iface>
node-exporter	node_filesystem_readonly	Filesystem read-only status.
node-exporter	node_exporter_build_info	A metric with a constant '1' value labeled by version
node-exporter	node_network_iface_link_mode	iface_link_mode value of /sys/class/net/<iface>.
node-exporter	node_network_receive_packets_total	Network device statistic receive_packets.
node-exporter	node_network_transmit_packets_total	Network device statistic transmit_packets.
node-exporter	node_memory_Mlocked_bytes	Memory information field Mlocked_bytes.

node-exporter	node_network_iface_id	iface_id value of /sys/class/net/<iface>.
node-exporter	node_memory_WritebackTmp_bytes	Memory information field WritebackTmp_bytes.
node-exporter	kube_service_status_load_balancer_ingress	Service load balancer ingress status
node-exporter	node_vmstat_pgpgout	/proc/vmstat information field pgpgout.
node-exporter	node_nf_contrack_entries	Number of currently allocated flow entries for connection tracking.
node-exporter	node_memory_Inactive_file_bytes	Memory information field Inactive_file_bytes.
node-exporter	node_memory_SwapFree_bytes	Memory information field SwapFree_bytes.
node-exporter	node_sockstat_TCP_mem	Number of TCP sockets in state mem.
node-exporter	node_memory_Slab_bytes	Memory information field Slab_bytes.
node-exporter	node_network_transmit_errs_total	Network device statistic transmit_errs.
node-exporter	node_memory_Active_bytes	Memory information field Active_bytes.
node-exporter	node_procs_blocked	Number of processes blocked waiting for I/O to complete.
node-exporter	node_sockstat_UDP_mem	Number of UDP sockets in state mem.
node-exporter	node_timex_maxerror_seconds	Maximum error in seconds.
node-exporter	node_memory_Inactive_bytes	Memory information field Inactive_bytes.
node-exporter	node_network_receive_errs_total	Network device statistic receive_errs.

node-exporter	node_memory_Unevictable_bytes	Memory information field Unevictable_bytes.
node-exporter	node_memory_KernelStack_bytes	Memory information field KernelStack_bytes.
node-exporter	node_procs_running	Number of processes in runnable state.
node-exporter	node_memory_SwapTotal_bytes	Memory information field SwapTotal_bytes.
node-exporter	node_netstat_IpExt_OutOctets	Statistic IpExtOutOctets.
node-exporter	node_memory_Active_file_bytes	Memory information field Active_file_bytes.
node-exporter	node_memory_SwapCached_bytes	Memory information field SwapCached_bytes.
node-exporter	node_netstat_Icmp_InMsgs	Statistic IcmpInMsgs.
node-exporter	node_forks_total	Total number of forks.
node-exporter	node_sockstat_RAW_inuse	Number of RAW sockets in state inuse.
node-exporter	node_time_seconds	System time in seconds since epoch (1970).
node-exporter	node_vmstat_pgpgin	/proc/vmstat information field pgpgin.
node-exporter	node_memory_Mapped_bytes	Memory information field Mapped_bytes.
node-exporter	node_memory_SUnreclaim_bytes	Memory information field SUnreclaim_bytes.
node-exporter	node_memory_HardwareCorrupted_bytes	Memory information field HardwareCorrupted_bytes.
node-exporter	node_memory_PageTables_bytes	Memory information field PageTables_bytes.

node-exporter	node_netstat_Udp6_InDatagrams	Statistic Udp6InDatagrams.
node-exporter	node_netstat_Icmp_OutMsgs	Statistic IcmpOutMsgs.
node-exporter	node_netstat_Udp6_NoPorts	Statistic Udp6NoPorts.
node-exporter	node_memory_AnonPages_bytes	Memory information field AnonPages_bytes.
node-exporter	node_memory_Committed_AS_bytes	Memory information field Committed_AS_bytes.
node-exporter	node_netstat_TcpExt_ListenOverflows	Statistic TcpExtListenOverflows
node-exporter	node_netstat_UdpLite_InErrors	Statistic UdpLiteInErrors.
node-exporter	node_entropy_available_bits	Bits of available entropy.
node-exporter	node_memory_Inactive_anon_bytes	Memory information field Inactive_anon_bytes.
node-exporter	node_vmstat_pswpin	/proc/vmstat information field pswpin.
node-exporter	node_memory_AnonHugePages_bytes	Memory information field AnonHugePages_bytes.
node-exporter	node_memory_SReclaimable_bytes	Memory information field SReclaimable_bytes.
node-exporter	node_netstat_IpExt_InOctets	Statistic IpExtInOctets.
node-exporter	node_netstat_Udp_NoPorts	Statistic UdpNoPorts.
node-exporter	node_timex_sync_status	Is clock synchronized to a reliable server (1 = yes)
node-exporter	node_memory_CommitLimit_bytes	Memory information field CommitLimit_bytes.

node-exporter	node_memory_VmallocChunk_bytes	Memory information field VmallocChunk_bytes.
node-exporter	node_netstat_Udp_InDatagrams	Statistic UdpInDatagrams.
node-exporter	node_netstat_Icmp6_InErrors	Statistic Icmp6InErrors.
node-exporter	node_netstat_Icmp6_OutMsgs	Statistic Icmp6OutMsgs.
node-exporter	node_netstat_UdpLite6_InErrors	Statistic UdpLite6InErrors.
node-exporter	node_netstat_TcpExt_SyncookiesSent	Statistic TcpExtSyncookiesSent
node-exporter	node_netstat_Tcp_InErrs	Statistic TcpInErrs.
node-exporter	node_intr_total	Total number of interrupts serviced.
node-exporter	node_timex_offset_seconds	Time offset in between local system and reference clock.
node-exporter	node_memory_Bounce_bytes	Memory information field Bounce_bytes.
node-exporter	node_memory_Writeback_bytes	Memory information field Writeback_bytes.
node-exporter	node_netstat_Udp_OutDatagrams	Statistic UdpOutDatagrams.
node-exporter	node_netstat_Icmp6_InMsgs	Statistic Icmp6InMsgs.
node-exporter	node_netstat_Ip6_OutOctets	Statistic Ip6OutOctets.
node-exporter	node_netstat_Ip_Forwarding	Statistic IpForwarding.
node-exporter	node_sockstat_TCP_orphan	Number of TCP sockets in state orphan.

node-exporter	node_netstat_Ip6_InOctets	Statistic Ip6InOctets.
node-exporter	node_netstat_TcpExt_SyncookiesFailed	Statistic TcpExtSyncookiesFailed.
node-exporter	node_netstat_Udp_InErrors	Statistic UdpInErrors.
node-exporter	node_vmstat_pgmajfault	/proc/vmstat information field pgmajfault.
node-exporter	node_network_transmit_drop_total	Network device statistic transmit_drop.
node-exporter	node_vmstat_pswpout	/proc/vmstat information field pswpout.
node-exporter	node_network_up	Value is 1 if operstate is 'up'
node-exporter	node_memory_NFS_Unstable_bytes	Memory information field NFS_Unstable_bytes.
node-exporter	node_memory_VmallocTotal_bytes	Memory information field VmallocTotal_bytes.
node-exporter	node_sockstat_FRAG_inuse	Number of FRAG sockets in sta inuse.
node-exporter	node_memory_Dirty_bytes	Memory information field Dirty_bytes.
node-exporter	node_netstat_Udp6_InErrors	Statistic Udp6InErrors.
node-exporter	node_netstat_TcpExt_SyncookiesRecv	Statistic TcpExtSyncookiesRecv.
node-exporter	node_netstat_Udp6_OutDatagrams	Statistic Udp6OutDatagrams.
node-exporter	node_memory_HugePages_Rsvd	Memory information field HugePages_Rsvd.
node-exporter	node_arp_entries	ARP entries by device

node-exporter	node_network_carrier	carrier value of /sys/class/net/<iface>.
node-exporter	node_timex_pps_stability_exceeded_total	Pulse per second count of stability limit exceeded events.
node-exporter	node_network_receive_compressed_total	Network device statistic receive_compressed.
node-exporter	node_network_transmit_carrier_total	Network device statistic transmit_carrier.
node-exporter	node_memory_DirectMap2M_bytes	Memory information field DirectMap2M_bytes.
node-exporter	node_memory_Hugepagesize_bytes	Memory information field Hugepagesize_bytes.
node-exporter	node_network_address_assign_type	address_assign_type value of /sys/class/net/<iface>.
node-exporter	node_network_receive_multicast_total	Network device statistic receive_multicast.
node-exporter	node_network_transmit_compressed_total	Network device statistic transmit_compressed.
node-exporter	node_memory_DirectMap4k_bytes	Memory information field DirectMap4k_bytes.
node-exporter	node_network_transmit_queue_length	transmit_queue_length value of /sys/class/net/<iface>.
node-exporter	node_memory_HugePages_Free	Memory information field HugePages_Free.
node-exporter	node_network_receive_frame_total	Network device statistic receive_frame.
node-exporter	node_memory_HugePages_Total	Memory information field HugePages_Total.
node-exporter	node_network_flags	flags value of /sys/class/net/<iface>.
node-exporter	node_network_receive_fifo_total	Network device statistic receive_fifo.

node-exporter	node_scrape_collector_duration_seconds	node_exporter: Duration of a collector scrape.
node-exporter	node_network_speed_bytes	speed_bytes value of /sys/class/net/<iface>.
node-exporter	node_sockstat_UDPLITE_inuse	Number of UDPLITE sockets in state inuse.
node-exporter	node_cpu_guest_seconds_total	Seconds the cpus spent in gues (VMs) for each mode.
node-exporter	node_filesystem_device_error	Whether an error occurred while getting statistics for the given device.
node-exporter	node_scrape_collector_success	node_exporter: Whether a collector succeeded.
node-exporter	node_network_transmit_fifo_total	Network device statistic transmit_fifo.
node-exporter	node_vmstat_pgfault	/proc/vmstat information field pgfault.
node-exporter	node_network_device_id	device_id value of /sys/class/net/<iface>.
node-exporter	node_network_protocol_type	protocol_type value of /sys/class/net/<iface>.
node-exporter	node_network_receive_drop_total	Network device statistic receive_drop.
node-exporter	node_timex_estimated_error_seconds	Estimated error in seconds.
node-exporter	node_disk_writes_merged_total	The number of writes merged.
node-exporter	node_network_transmit_colls_total	Network device statistic transmit_colls.
node-exporter	node_timex_tick_seconds	Seconds between clock ticks.
node-exporter	node_textfile_scrape_error	1 if there was an error opening c reading a file

node-exporter	node_network_iface_link	iface_link value of /sys/class/net/<iface>.
node-exporter	node_disk_reads_merged_total	The total number of reads merged.
node-exporter	node_timex_status	Value of the status array bits.
node-exporter	node_netstat_Icmp_InErrors	Statistic IcmpInErrors.
node-exporter	node_memory_Active_anon_bytes	Memory information field Active_anon_bytes.
node-exporter	node_timex_pps_frequency_hertz	Pulse per second frequency.
node-exporter	node_network_mtu_bytes	mtu_bytes value of /sys/class/net/<iface>.
node-exporter	node_timex_tai_offset_seconds	International Atomic Time (TAI) offset.
node-exporter	node_timex_pps_jitter_total	Pulse per second count of jitter limit exceeded events.
node-exporter	node_timex_pps_jitter_seconds	Pulse per second jitter.
node-exporter	node_network_net_dev_group	net_dev_group value of /sys/class/net/<iface>.
node-exporter	node_network_dormant	dormant value of /sys/class/net/<iface>.
node-exporter	node_timex_pps_calibration_total	Pulse per second count of calibration intervals.
node-exporter	node_timex_pps_shift_seconds	Pulse per second interval duration.
node-exporter	node_timex_pps_error_total	Pulse per second count of calibration errors.
node-exporter	node_memory_VmallocUsed_bytes	Memory information field VmallocUsed_bytes.

node-exporter	node_timex_frequency_adjustment_ratio	Local clock frequency adjustment.
node-exporter	node_sockstat_FRAG_memory	Number of FRAG sockets in sta memory.
node-exporter	node_memory_HugePages_Surp	Memory information field HugePages_Surp.
node-exporter	node_timex_loop_time_constant	Phase-locked loop time constan
node-exporter	node_timex_pps_stability_hertz	Pulse per second stability

相关资源使用及计费说明

最近更新时间：2024-01-29 16:01:49

当您使用 Prometheus 监控服务（TMP）服务时，可能会使用到 **TKE Serverless 集群**、**Grafna 可视化服务和负载均衡 CLB** 资源，本文将为您介绍这些资源使用场景以及相关费用。

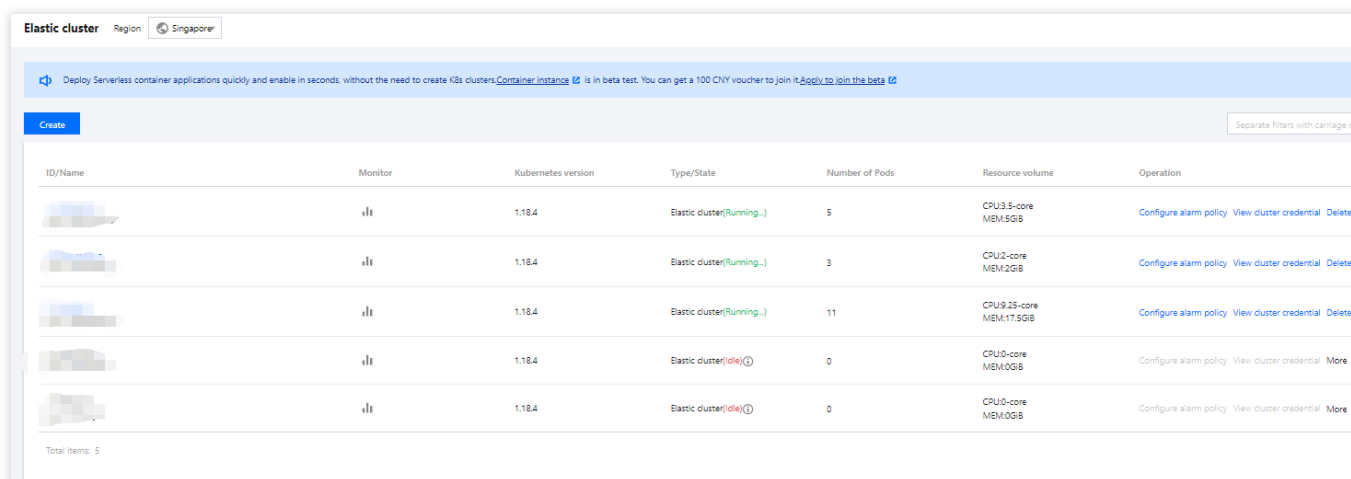
EKS 集群

使用场景

使用 Prometheus 监控服务关联集群来监控容器服务，需要创建一个 TKE Serverless 集群。

在集成中心安装集成插件时会自动创建 TKE Serverless 集群用于数据采集。

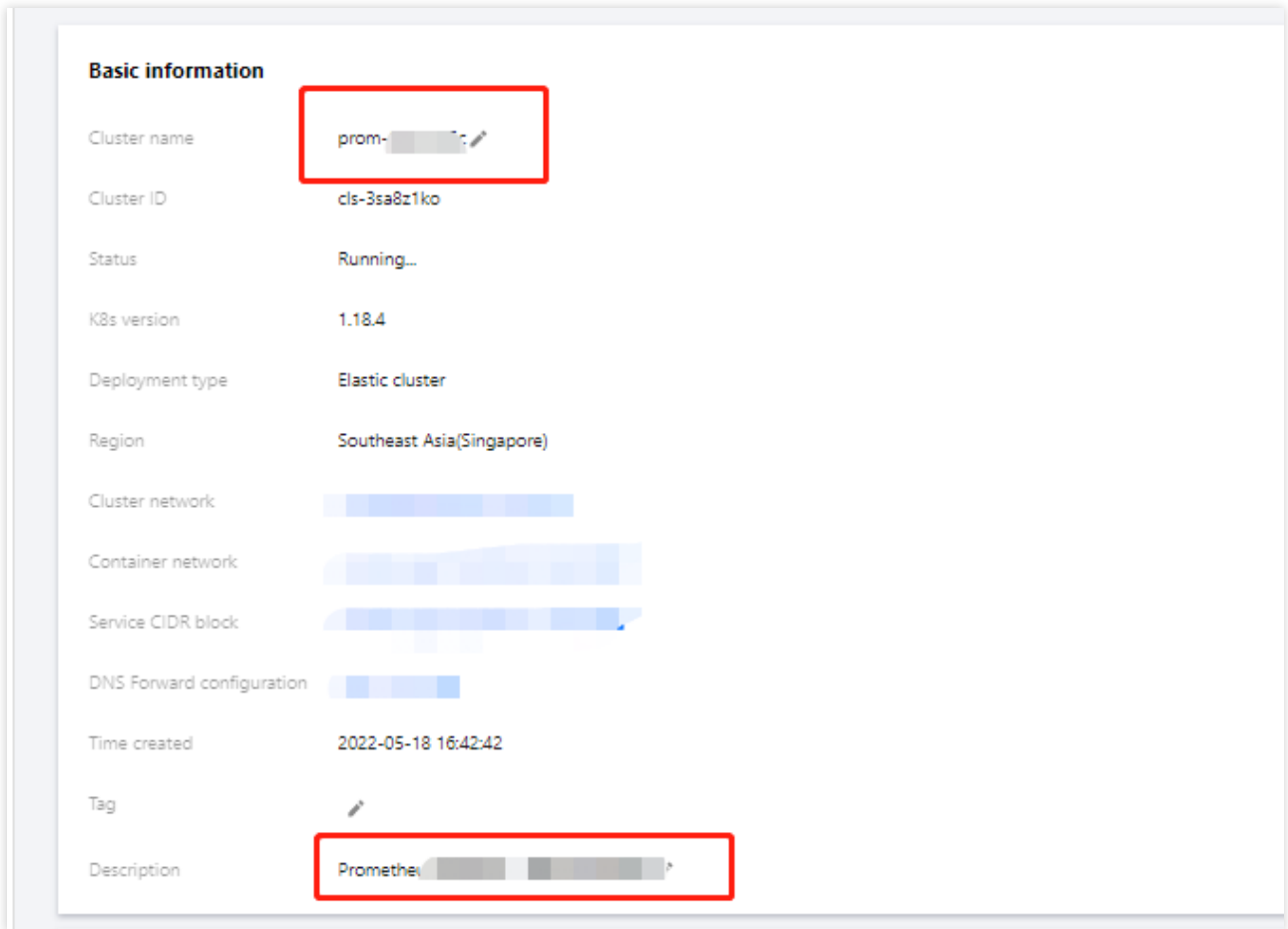
上述两种情况下会创建 TKE Serverless 集群，若两种情况都在使用，只会创建并共用一个 TKE Serverless 集群。在 [集群列表页](#) 可查看已创建的 TKE Serverless 集群，如下图所示：



ID/Name	Monitor	Kubernetes version	Type/State	Number of Pods	Resource volume	Operation
[blurred]	[monitor icon]	1.18.4	Elastic cluster(Running-)	5	CPU3.5-core MEM4.5GIB	Configure alarm policy View cluster credential Delete
[blurred]	[monitor icon]	1.18.4	Elastic cluster(Running-)	3	CPU2-core MEM2GIB	Configure alarm policy View cluster credential Delete
[blurred]	[monitor icon]	1.18.4	Elastic cluster(Running-)	11	CPU9.25-core MEM17.5GIB	Configure alarm policy View cluster credential Delete
[blurred]	[monitor icon]	1.18.4	Elastic cluster(idle)	0	CPU0-core MEM0GIB	Configure alarm policy View cluster credential More
[blurred]	[monitor icon]	1.18.4	Elastic cluster(idle)	0	CPU0-core MEM0GIB	Configure alarm policy View cluster credential More

注意事项

该 TKE Serverless 集群的名称为 Prometheus 监控服务实例的 ID，集群描述里面说明为 **Prometheus 监控专用**，请勿修改或删除。



计费说明

计费方式为**按量计费**，计费详情请参见 [EKS 产品定价](#)。

EKS 集群会按照监控量进行自动扩缩容，监控规模和 EKS 集群费用的关系可参考：

用户上报的瞬时 Series 量级	预估需要的 EKS 资源	对应的刊例价费用/日
<50万	1.25核 1.6GiB	0.35美元
100万	0.5核1.5GiB*2	1.46美元
500万	1核3GiB*3	2.93美元
2000万	1核6GiB*5	7.98美元
3000万	1核6GiB*8	12.77美元

EKS 集群成本示例如下：

一个新初始化的 Prometheus 实例所用 EKS 集群消耗了：CPU:1.25 核、内存:1.5GiB。预计一天刊例价费用为：
 $0.0319 \times 24 + 0.0132 \times 24 = 1.0824$ 美元。

Grafana 可视化服务

使用场景

当您创建 Prometheus 实例时，需要绑定一个地域相同的 Grafana 实例，用于可视化展示 Prometheus 采集的监控数据。详细实例价格可参见 [Grafana 计费说明](#)。

负载均衡 CLB

使用 Prometheus 服务监控关联集群监控容器服务，常规情况下会在用户账户下创建一个内网 CLB 用于打通采集器与用户集群的网络。

若用户关联了边缘集群，或跨集群关联了未打通网络的集群，支持创建公网的 CLB 进行网络联通，此时会创建一个公网 CLB。

若要使用通过外网访问 Grafana 服务，则需要创建一个相应的公网 CLB。

这些 CLB 资源会收取费用，创建的公网 LB 可在 [负载均衡控制台](#) 查看资源信息。

该资源按实际使用量计费，计费详情请参见负载均衡 [标准账户类型计费说明](#) 文档。

资源销毁

在 [Prometheus 监控服务控制台](#) 销毁监控实例，对应的所有资源将会一并销毁。腾讯云不主动回收用户的监控实例，若您不再使用 Prometheus 监控服务，请务必及时删除监控实例，以免发生资源的额外扣费。如需销毁 Prometheus 实例可参见 [销毁实例](#) 文档。