Prometheus 监控服务

操作指南

腾讯云

文档目录

操作指南
实例
创建实例

最近更新时间：2024-12-05 16:07:33

本文以自定义配置方式为例，指导您如何创建 Prometheus 实例。

前提条件

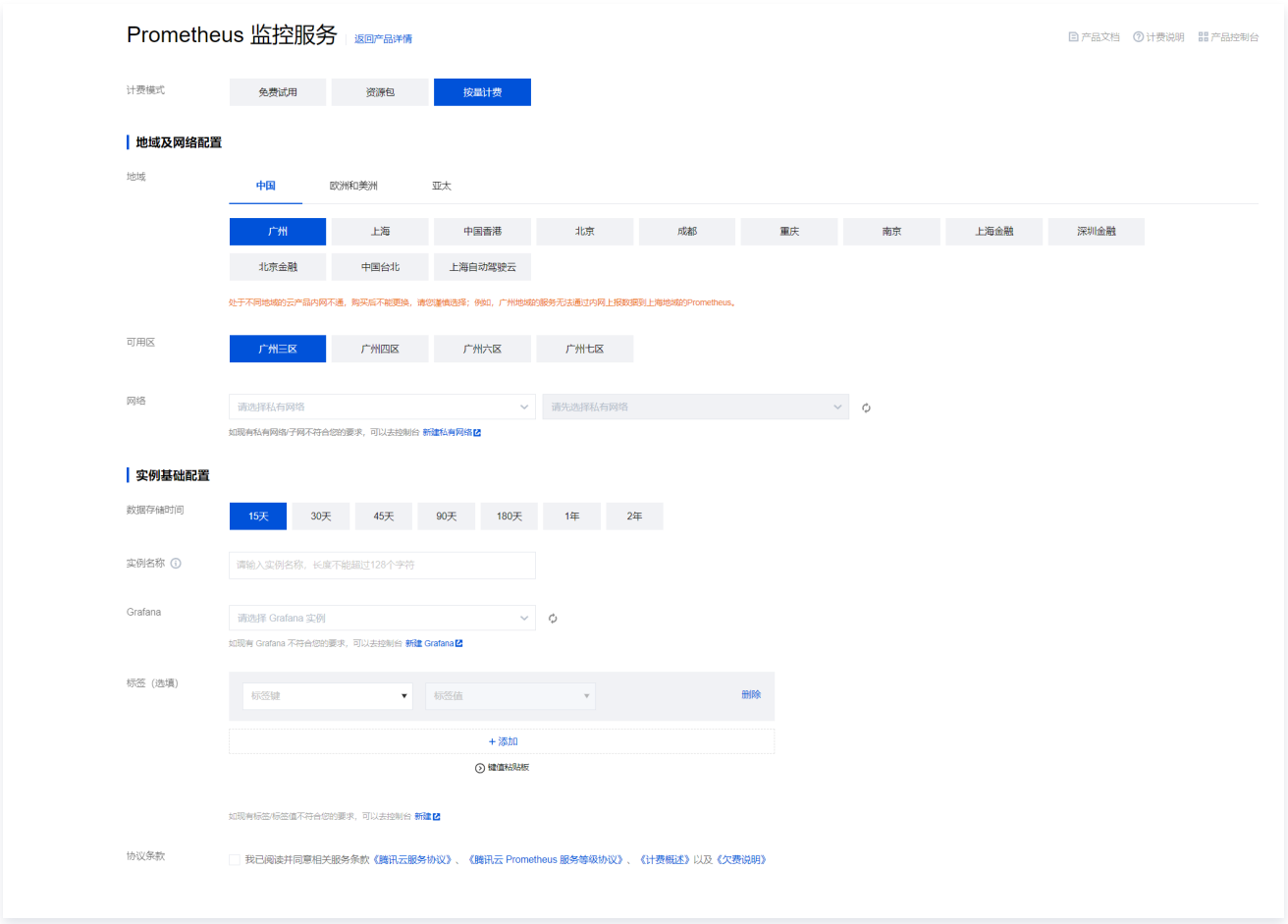
在创建 Prometheus 实例前，您需要完成以下工作：

- 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 腾讯云可观测平台 。
2. 在左侧菜单栏中单击 **Prometheus 监控**。
3. 在 Prometheus 监控页面单击**新建**，根据页面提示，配置以下信息：

类别	必选/可选	配置说明
计费模式	必选	目前支持资源包和按量付费模式，新用户支持免费试用15天。资源包具有较高的折扣优惠，目前资源包支持境内外多个城市，您可以自行选择购买。
地域	必选	处于不同地域的云产品内网不同，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	根据您实际需求选择。Prometheus 存储层目前并不区分可用区，同地域购买不同可用区的实例并不能作为多可用区灾备方案（建议使用云联网等方式上报到其它地域作为替代）。
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。选择后只展示在同一私有网络的 Grafana 。
套餐包选择（仅套餐包类型可选）	必选	支持基础版年包、升级版年包和豪华版年包。具体价格以购买页为准。
数据存储时间	必选	资源包只能选择存储时长为30天和180天的时长；按量付费不同的存储时长价格有差异，详情请参见 按量付费产品定价 。
实例名称	必选	用户自定义 Prometheus 实例名称。
Grafana	可选	选择对应的 Grafana 实例（仅展示与所选网络同一 VPC 的 Grafana 实例，若没有符合的 Grafana ，请参见 操作指引 创建）
标签	可选	设置标签之后可以用于从不同维度对资源分类管理。具体可参见 标签说明。
到期或用完处理方式（仅套餐包类型可选）	可选	勾选后，资源包到期或资源包内任一种用量额度用完后，自动转为按量付费。



4. 配置完后，勾选服务条款并单击**立即购买**支付即可。

搜索实例

最近更新时间：2024-12-05 16:07:33

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 腾讯云可观测平台 。
2. 在左侧菜单栏中单击 **Prometheus 监控**。
3. 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。

4. 支持用户根据不同的维度来过滤实例列表，目前支持以下几种维度：

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	支持用户通过 IPv4 地址进行精准查询。
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：2024-12-05 16:07:33

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 腾讯云可观测平台 。
2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 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**。
4. 在弹出的“**修改名称**”窗口中，输入新实例名称，单击**确定**即可。

销毁实例

最近更新时间：2024-12-05 16:07:33

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服状态。对于停服状态的实例，您可以根据不同场景和需求进行重建实例。

相关影响

当实例进入停服状态时，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为准。

操作步骤

1. 登录 腾讯云可观测平台 。
2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 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/退还**。

4. 由于销毁操作属于高危操作，需在弹出的**销毁**窗口中，完成销毁操作步骤，单击**确定**即可。

强制释放

当您实例进行销毁/退还处理，实例7天内处于停服状态，在此期间的数据将继续保留。实例仍存在 Prometheus 监控服务列表中，您可以执行强制释放，释放后将不会在列表中展示，释放后该实例下所有数据将被清除且不可恢复。

1. 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 > 强制释放**。
2. 由于释放操作属于高危操作，实例下所有数据将被清除且不可恢复。请您确认后在弹出的**强制释放**窗口中，完成释放操作步骤，单击**确定**即可。

重建实例

最近更新时间：2024-12-05 16:07:33

如果您需要对停服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 腾讯云可观测平台 。
2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 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**。
4. 在弹出的“重建操作须知”窗口中，单击**确定**即可。

> **说明：**
> 如果当前实例状态为运行中，无法进行重建操作。

修改存储时长

最近更新时间：2024-12-05 16:07:33

> **说明：**
> 资源包不支持修改存储时长。

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 腾讯云可观测平台 。
2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 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**。

4. 在弹框中选择需要修改的存储时长，单击**确定**即可。

> **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。

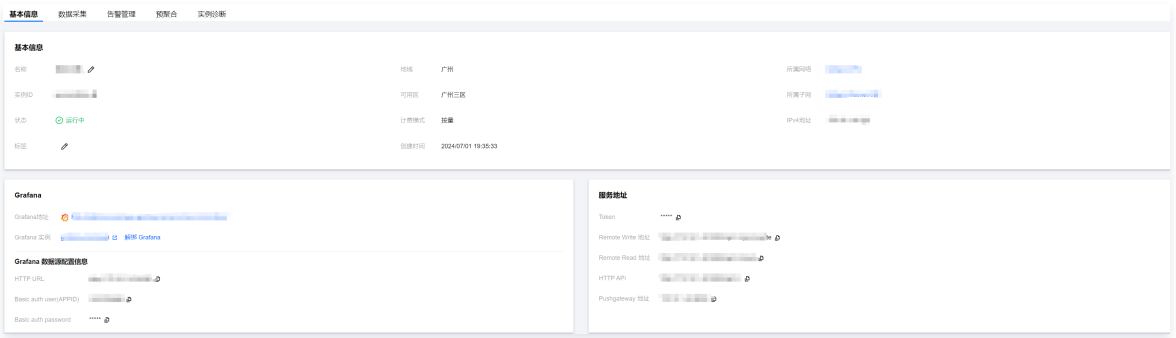
查看实例基本信息

最近更新时间：2024-12-05 16:07:33

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 腾讯云可观测平台 。
2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 在实例列表中，选择需要查看的 Prometheus 实例，单击实例的 ID/名称。



4. 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。
   - 绑定 / 解绑 Grafana 实例。

容器监控
集成容器服务

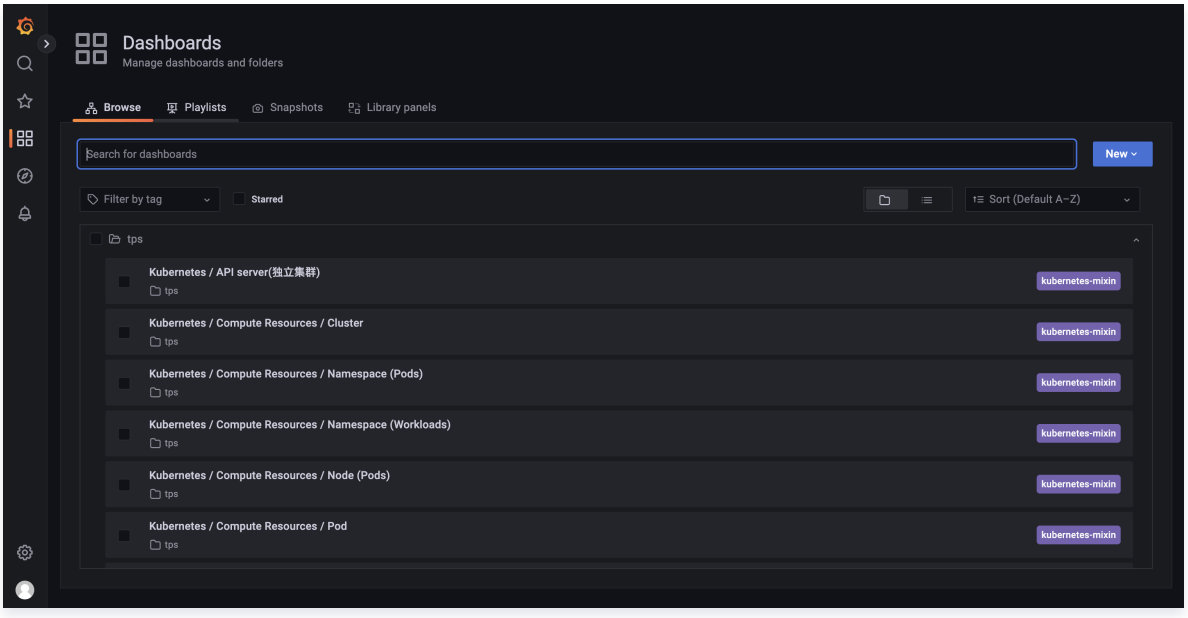
最近更新时间：2025-06-20 15:22:42

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。作为容器领域实施的标准，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，可以大大简化用户通过 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，在顶部导航栏中单击**数据采集**。
4. 在集成容器服务页面进行下列操作：
   - 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 精简监控指标 。
5. 完成以上操作后，在 Prometheus 实例列表点击对应的 Grafana 图标，查看您容器服务的监控数据。



关联集群

> **注意：**
> 关联集群成功后将在集群中安装监控数据采集插件，该插件在解除关联的同时会被删除。当前支持跨 VPC 关联，支持在同一个监控实例内监控不同地域不同 VPC 下的集群。

前提条件

- 已登录 腾讯云可观测平台控制台，并创建集群。
- 已创建 Prometheus 实例。

操作步骤

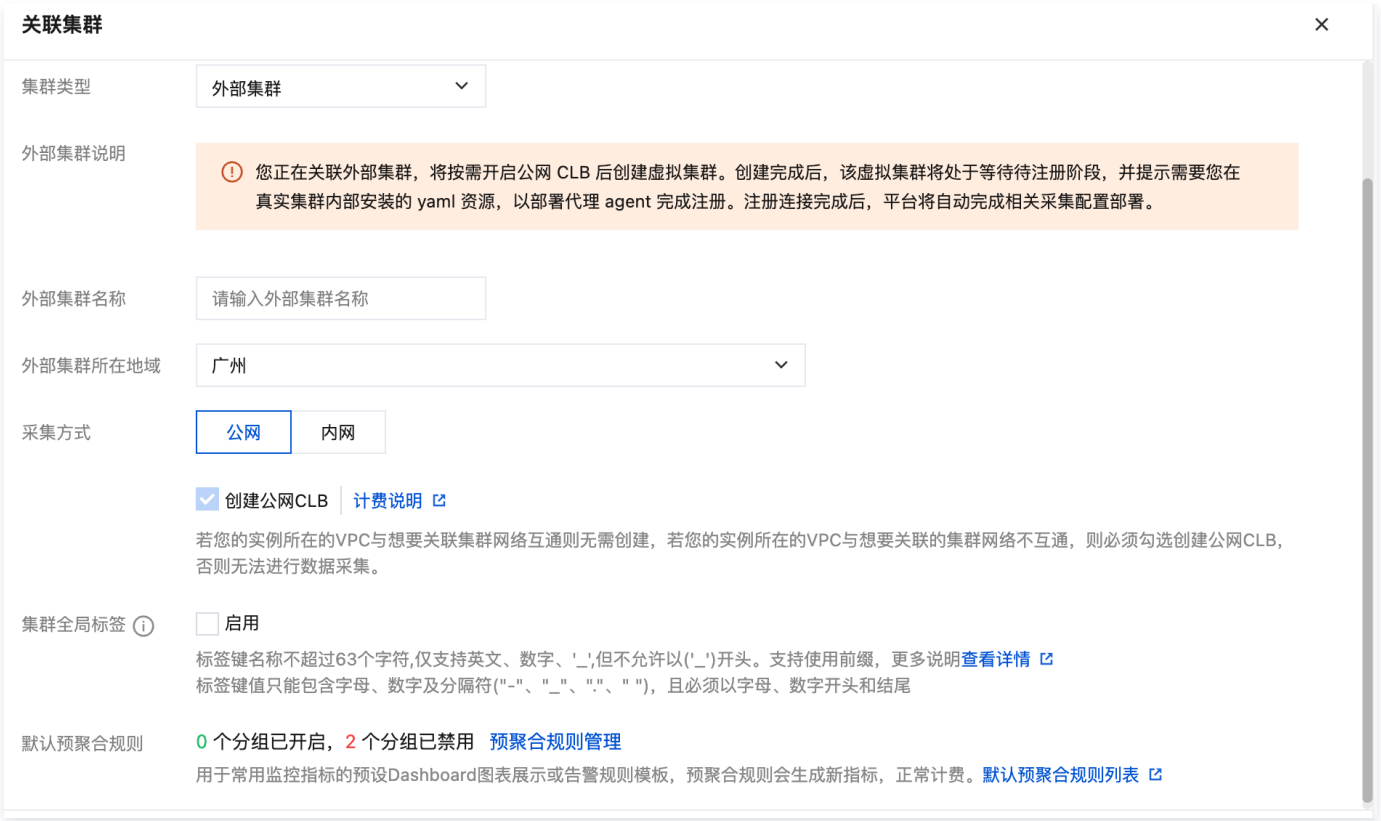
关联腾讯云上 Kubernetes 集群

1. 登录 腾讯云可观测平台控制台，选择左侧导航栏中的 **Prometheus 监控**。
2. 在监控实例列表页，选择需要关联集群操作的实例名称，进入该实例详情页。
3. 在顶部导航栏中单击**数据采集 > 集成容器服务 > 关联集群**。
4. 在弹出的“关联集群”窗口，选择相关集群。

   - **集群类型**：容器服务的标准集群、 弹性集群、注册集群、边缘集群。
   - **跨 VPC 关联**：开启后支持在同一个监控实例内监控不同地域不同VPC下的集群。
     - **创建公网 CLB**：若您的实例所在的 VPC 与想要关联集群网络互通则无需创建；若您的实例所在的 VPC 与想要关联的集群网络不互通，则必须勾选创建公网 CLB，否则无法进行跨 VPC 集群的数据采集。例如：若您实例所在的 VPC 与想要关联集群所在的 VPC 已经通过 云联网 打通，则不需要创建公网 CLB。
     - **集群所在地域**：选择集群所在地域。
   - **集群**：选择需要关联的集群，支持多选。
   - **集群全局标签**：用于给每个监控指标打上相同的键值对。
   - **默认预聚合规则**：默认安装预聚合规则，如果已经安装会提示开启状态。
5. 单击**确定**即可将所选集群和当前监控实例关联。

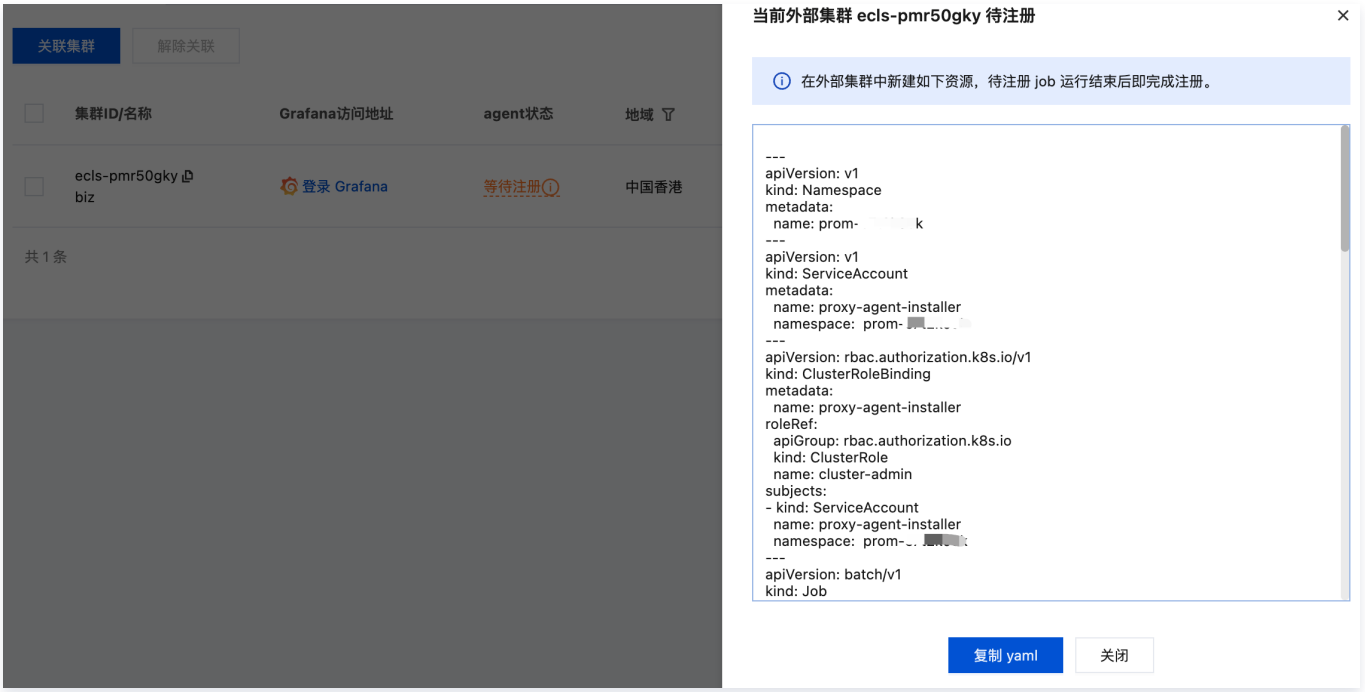
关联外部 Kubernetes 集群

1. 登录 腾讯云可观测平台控制台，选择左侧导航栏中的 **Prometheus 监控**。
2. 在监控实例列表页，选择需要关联集群操作的实例名称，进入该实例详情页。
3. 在顶部导航栏中单击**数据采集 > 集成容器服务 > 关联集群**。
4. 在弹出的“关联集群”窗口，**集群类型**选择**外部集群**。



- **外部集群名称**：给集群取个名称。
- **外部集群所在地域**：选择要注册集群所在地域或临近地域。
- **采集方式**：如果外部集群已经和公有云 VPC 打通选择内网接入，否则选择公网。公网接入需要付额外的公网 CLB 费用。
- **集群全局标签**：用于给每个监控指标打上相同的键值对。
- **默认预聚合规则**：默认安装预聚合规则，如果已经安装会提示开启状态。

5. **单击确定开始初始化，会生成一个集群 ID。**
6. **注册外部集群。**



- **初始化成功后，点击等待注册弹出 Job 安装 YAML。Job 执行 Helm 命令安装采集相关组件，组件和权限详细说明请参见 TKE 集群内安装组件说明。Job 主要参数如下：**

Job 名	参数	说明
proxy-agent-installer（安装 proxy-agent）	timeout	Helm 安装等待的超时时间，超时未完成会自动回滚
	instanceId	实例 ID
	instanceToken	实例 Token
	clusterId	集群 ID
	clusterType	集群类型
	serverAddress	proxy-agent 注册地址，只有正常访问这个地址才能注册成功
	image	proxy-agent 镜像
exporter-installer（安装 kube-state-metrics/node-exporter 基础采集组件）	kubeStateMetrics.enabled	是否安装 kube-state-metrics 组件，如果不需要或者已安装可以设置为 false
	kubeStateMetrics.repo	kube-state-metrics 镜像仓库地址（不包含版本，版本根据当前集群版本自动设置）
	nodeExporter.enabled	是否安装 node-exporter 组件，如果不需要或者已安装可以设置为 false
	nodeExporter.image	node-exporter 镜像

- 创建并查看 Job 运行结果。

```
# 安装注册任务
kubectl apply -f <yaml>
# 设置 namespace
export KUBE_NS=<实例 ID>
# 查看 Job 执行状态
kubectl get job proxy-agent-installer -n ${KUBE_NS}
# 查看 Job POD，有失败可以查看出错日志
kubectl get pods -l job-name=proxy-agent-installer  -n ${KUBE_NS}
```

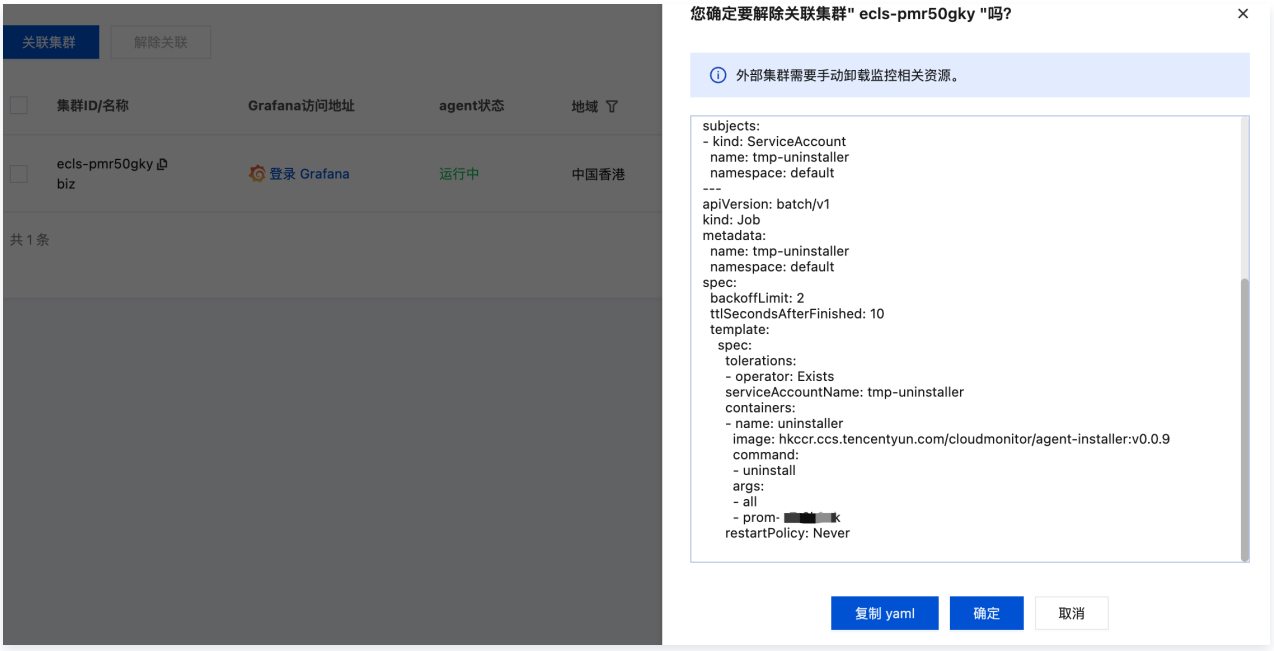
- 查看 proxy-agent 日志，日志应该包含 conn is active，否则用户要检查集群内能否访问 serverAddress 指定的 IP:Port。

```
# 获取 proxy-agent pod 名
export KUBE_POD=`kubectl  get pods -l k8s-app=proxy-agent -n ${KUBE_NS}|sed '1d'|head -1|awk
'{print $1}'`
# 查看 proxy-agent 日志
kubectl logs ${KUBE_POD} -n ${KUBE_NS}
```

7. 注册成功后等待1 – 2分钟，控制台上 **agent 状态**会变成**运行中**。接下来就可以像云上 Kubernetes 集群一样操作外部集群。

解除关联

1. 登录 腾讯云可观测平台控制台，选择左侧导航栏中的 **Prometheus 监控**。
2. 在监控实例列表页，选择解除关联的实例名称，进入该实例详情页。
3. 在**数据采集 > 集成容器服务**页面，单击实例右侧的**更多 > 解除关联**。
4. 在弹出的“解除关联集群”窗口，单击**确定**即可解除关联。
5. 对于外部集群，需要用户执行弹出 YAML 清理集群内监控相关资源。



说明：

如需在容器服务控制台关联 Prometheus 集群，请参见 容器服务关联 Prometheus 。

容器服务关联 Prometheus

最近更新时间：2024-11-27 11:46:22

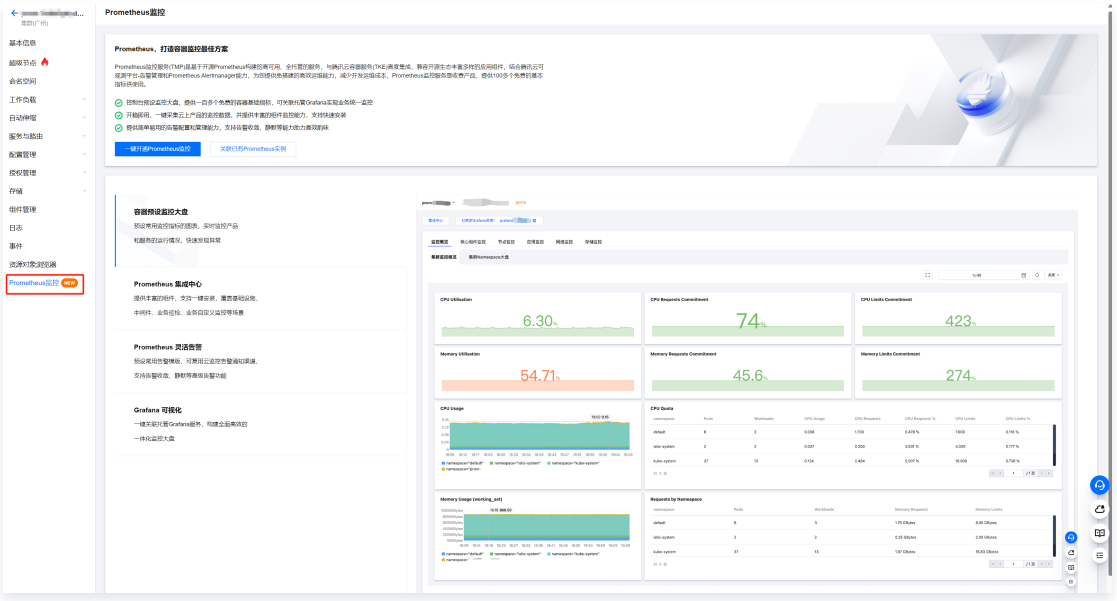
容器场景监控是 Prometheus 监控的主要用户场景，本文将为您介绍集群如何在容器场景监控下关联 Prometheus 监控，且关联 Prometheus 监控后，可以直接在集群页面查看监控数据图表，大大提高了用户的使用体验。

前提条件

账户下已有集群且暂未关联 Prometheus 监控。

操作步骤

1. 登录 容器服务控制台 。
2. 在左侧导航栏中选择**集群**，单击一个**集群名称/ID** 进入集群详情页。
3. 在集群详情页的左侧导航栏选择 **Prometheus 监控**，如下图所示。



4. 在弹出的 Prometheus 监控页面用户可选择 一键开通 Prometheus 监控 或者 关联已有 Prometheus 实例 。

方式一：一键开通 Prometheus 监控

> **说明：**
> 一键开通仅支持默认的按量付费计费模式和默认的存储时间15天。如需创建其他规格的 Prometheus 实例，请前往 Prometheus 控制台 创建实例 。

在弹出的 Prometheus 监控页面选择**一键开通 Prometheus 监控**，选择子网并勾选相关协议后点击**确定**。

方式二：关联已有 Prometheus 实例

在弹出的 Prometheus 监控页面选择关联 Prometheus 实例，会出现以下几种情况，请根据自身情况选择对应处理方案。

- **该账号下没有任何 Prometheus 实例，请按照下图提示去 新建 Prometheus 实例 。**

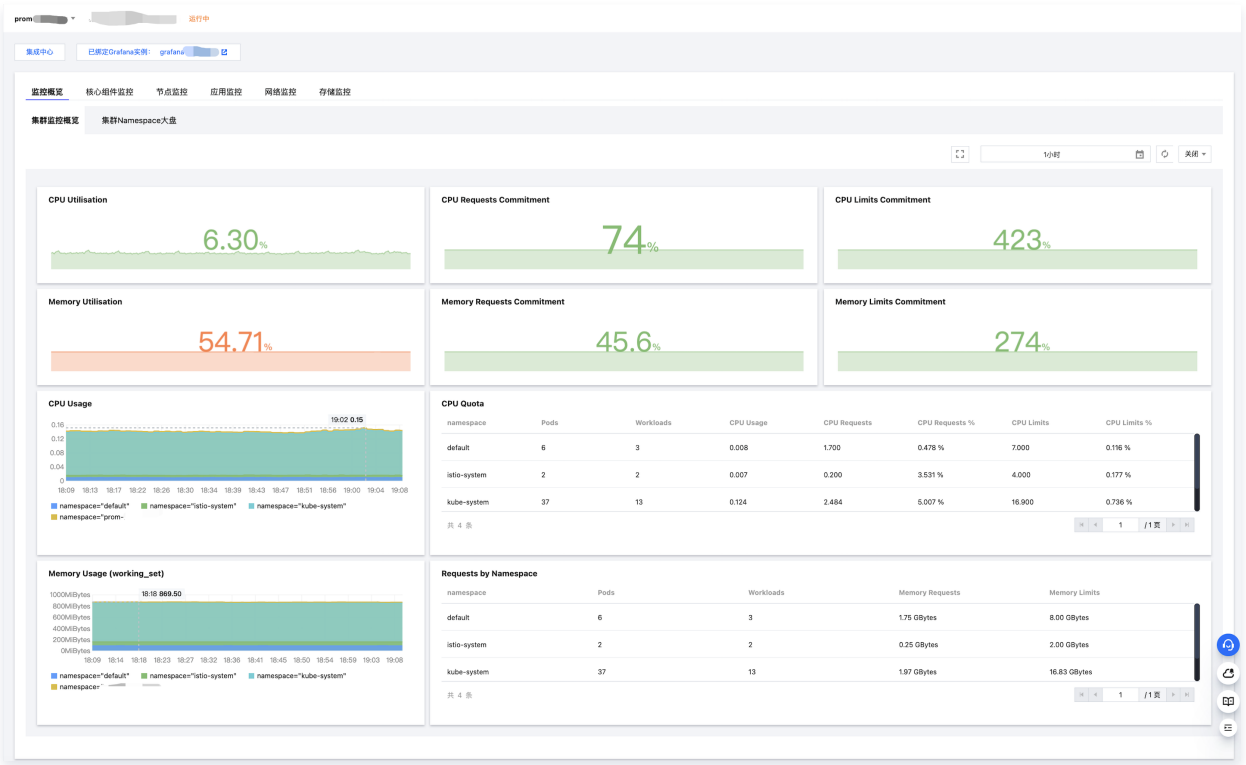
- **该账号下有 Prometheus 实例，但当前 VPC 下无实例，用户可选择 创建实例 或开启跨 VPC 关联。**

- **该账号当前 VPC 下有 Prometheus 实例，客户选择当前 VPC 内的 Prometheus 实例。如下图所示：**

- **用户选择开启跨 VPC 关联，会展示创建公网 CLB 实例功能，同时，可选择的实例列表是当前账号下的所有 Prometheus 实例。如下图：**

容器指标监控图表

1. **当成功关联 Prometheus 监控以后，展示容器常用指标的监控图表。**



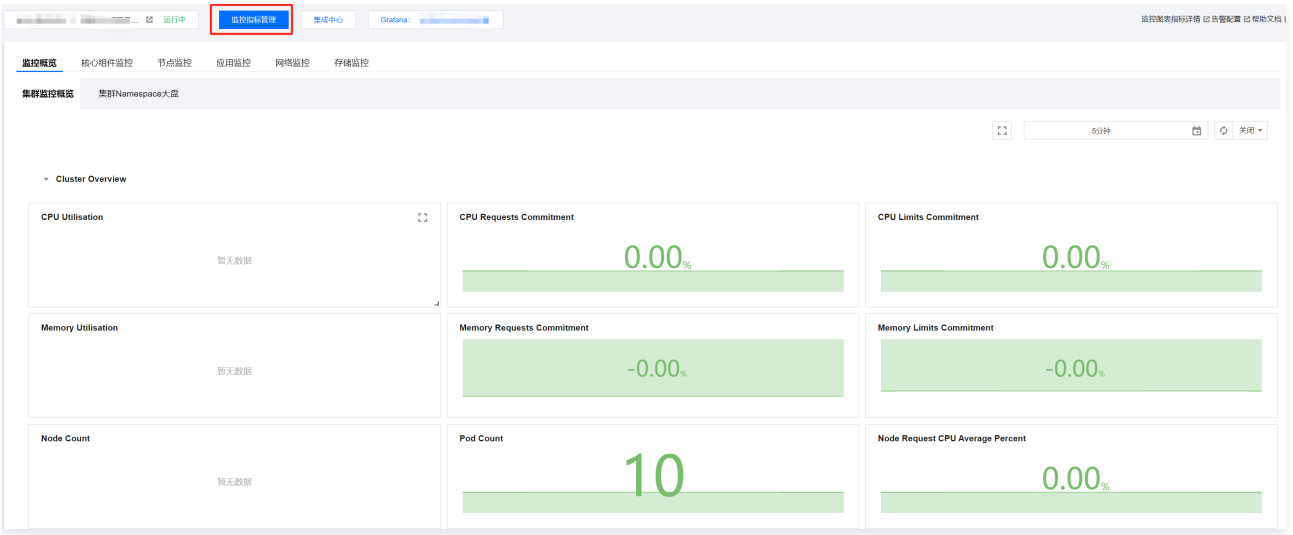
2. **若需要监控容器中运行的其他组件，可以点击集成中心，跳转到 Prometheus 监控的 集成中心 安装其他类型的监控组件。若当前图表不满足需求，可以点击绑定 Grafana 服务，在 绑定 Grafana 服务 后前往 Grafana平台配置更多指标的监控图表。**

> **说明：**

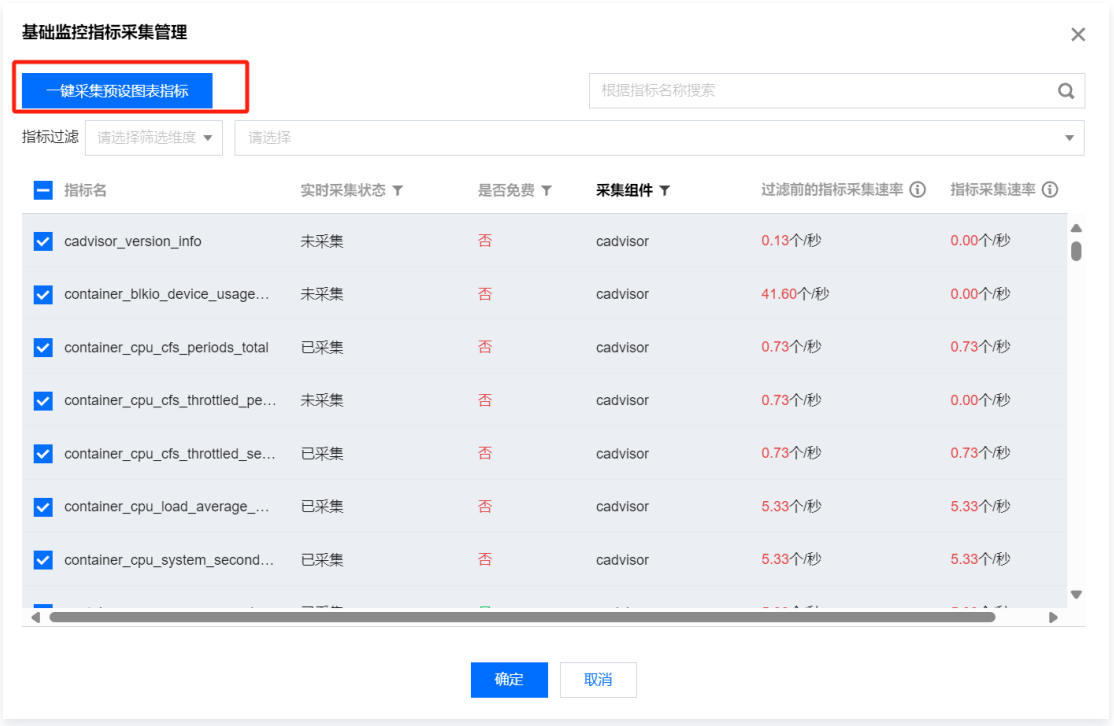
> 如需在 Prometheus 控制台关联容器集群，可参考 集成容器服务 。

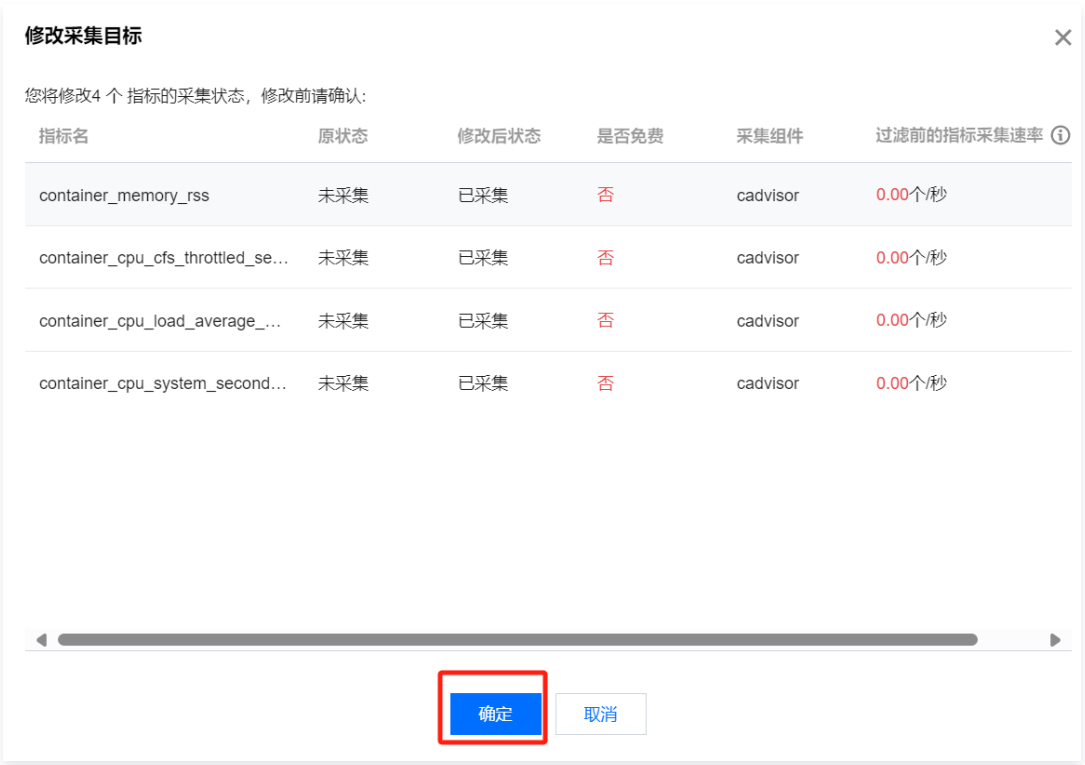
配置指标数据

进入监控图表页面后，我们会发现部分图表有数据，部分图表显示**暂无数据**。这是由于免费的指标会自动接入，免费指标可参考 按量付费免费指标；而无数据的指标可以参考 容器监控图表 。配置指标的步骤如下：
点击监控图表页面上方的**监控指标管理**，即可跳转到基础监控指标采集管理页面。

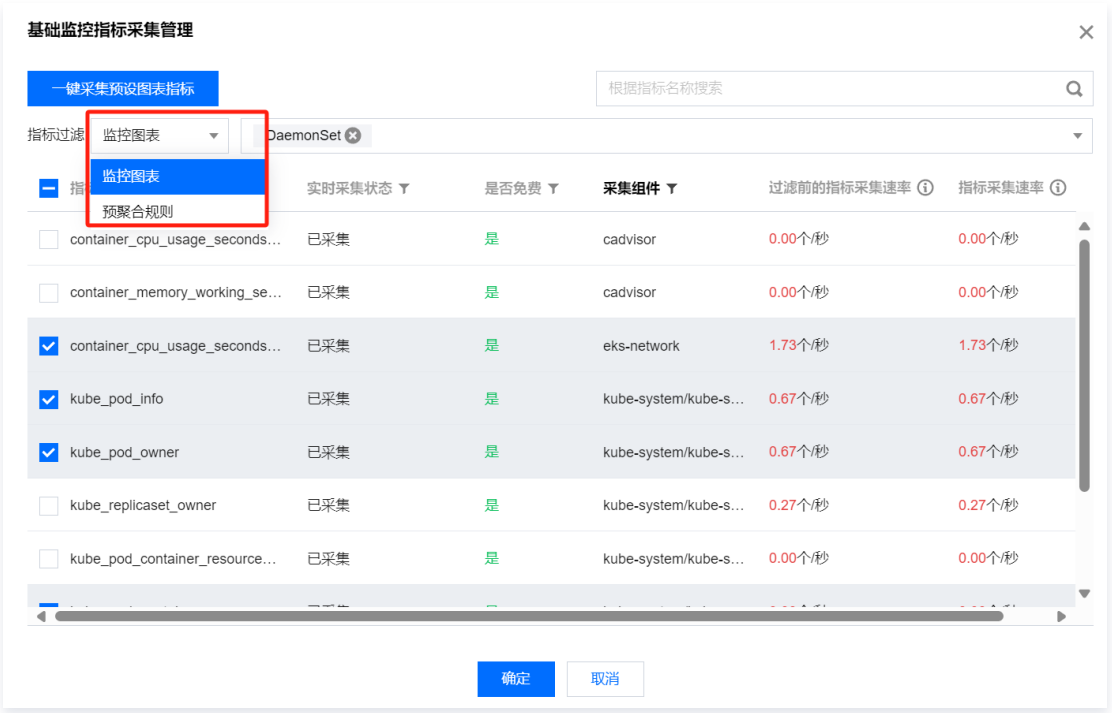


- 若需要所有内嵌图表都显示有数据，可以在弹出的方框中选择**一键采集预设图表指标**后会再弹出一个确认窗口，点击**确定**将会采集 Prometheus 监控预设容器图表相关的全部指标。





- **若只需要采集部分指标，可以通过监控图表和预聚合规则进行指标筛选，勾选自己需要的指标后再点击确定即可。如下图所示：**



说明：

由于有些指标属于预聚合指标，此类指标需要添加预聚合规则，若默认预聚合规则是关闭状态，我们需要前往 Prometheus 控制台，进入实例 ID > 预聚合 > 开启默认预聚合规则。

集成中心

最近更新时间：2025-01-15 14:09:12

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、组件监控、应用层监控、业务监控等监控场景，方便您快速接入并使用。

支持服务列表

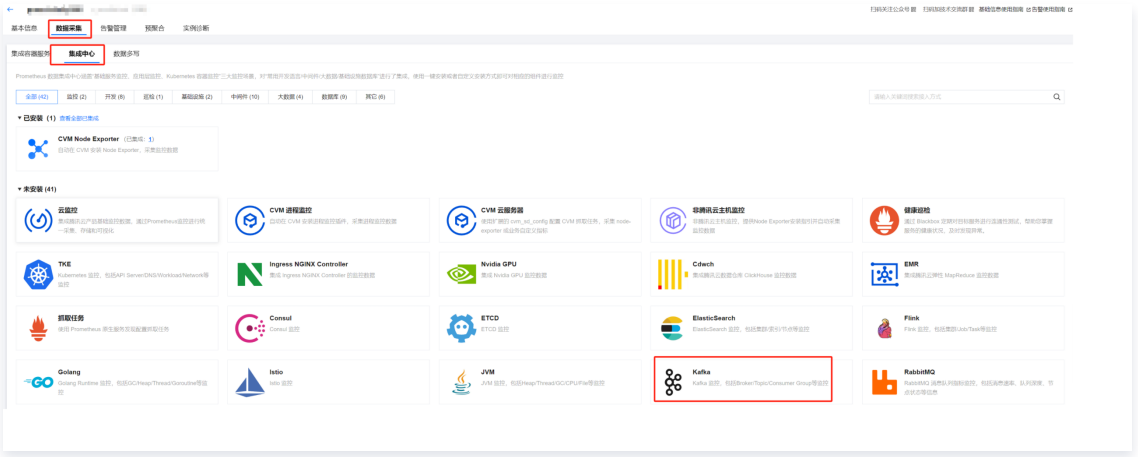
服务类型	服务名称	监控项	是否支持一键安装	接入文档
监控	云监控	集成腾讯云产品基础监控数据，通过 Prometheus 监控进行统一采集、存储和可视化	支持	云监控
	PTS	云压测监控，提供压测任务 RPS、响应时间、错误率、压测节点内存/CPU 等监控	不支持	使用 Prometheus 观测性能压测指标
开发	CVM Node Exporter	自动在 CVM 安装 Node Exporter，采集监控数据	支持	批量安装 Node Exporter
	CVM 进程监控	自动在 CVM 安装进程监控插件，采集进程监控数据	支持	CVM 进程监控
	CVM 云服务器	使用扩展的 cvm_sd_config 配置 CVM 抓取任务，采集 node-exporter 或业务自定义指标	支持	云服务器场景下自定义接入
	非腾讯云主机监控	非腾讯云主机监控，提供 Node Exporter 安装指引并自动采集监控数据	支持	非腾讯云主机监控
	抓取任务	使用原生 static_config 配置抓取任务	支持	抓取配置说明
	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
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
基础设施	TKE	Kubernetes 监控，包括 API Server/DNS/Workload/Network 等监控	不支持	集成容器服务
	Nvidia GPU	集成 Nvidia GPU 监控数据	支持	Nvidia Gpu Exporter 接入
中间件	Ingress NGINX Controller	Ingress NGINX Controller 监控	支持	Ingress NGINX Controller Exporter 接入
	Consul	Consul 监控	支持	Consul Exporter 接入
	Etcd	Etcd 监控	不支持	配置 Prometheus 监控
	Istio	Istio 监控	不支持	-
	Kafka	包括 Broker/Topic/Consumer Group 等监控	支持	Kafka Exporter 接入
	RabbitMQ	RabbitMQ 消息队列指标监控，包括消息速率、队列深度、节点状态等信息	支持	Rabbitmq Exporter 接入
	RocketMQ	RocketMQ 监控，包括 Broker/Producer/Consumer Group 等监	支持	-

		控		
	Fluentd	fluentd 性能监控，包括组件状态、缓冲区使用情况和重试次数等	不支持	Fluentd/FluentBit 接入
	Nginx	Nginx 服务指标监控，包括健康状况、性能和负载情况等	支持	Nginx Exporter 接入
	Kong	Kong 监控	不支持	使用 Prometheus 监控
大数据	Cdwch	集成腾讯云数据仓库 ClickHouse 监控数据	支持	-
	EMR	集成腾讯云弹性 MapReduce 监控数据	支持	Prometheus 采集 EMR 组件
	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
数据库	云数据库 Memcached	Memcached 监控	支持	Memcached Exporter 接入
	云数据库 MongoDB	包括文档数/读写性能/网络流量等	支持	MongoDB Exporter 接入
	云数据库 MySQL	包括网络/连接数/慢查询等	支持	MySQL Exporter 接入
	MSSQL	Microsoft SQL Server 指标监控，包括 SQL Server 性能和健康状况等	支持	SQL Server 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
	Aerospike	aerospike 数据库指标监控，包括集群的健康状况、性能表现和负载情况	支持	Aerospike Exporter 接入
	Ceph	Ceph 监控，包括集群、OSD、Pools 的状态、性能等	支持	Ceph Exporter 接入
	OracleDB	Oracle 数据库指标监控，包括数据库性能、负载、健康状况等	支持	Oracle DB Exporter 接入
其它	Docker	Docker daemon 监控，包括 docker、container、engine 等信息	不支持	Docker 接入
	Pushgateway	Prometheus Pushgateway，用于接收 Prometheus 服务发现无法直接监控的短期任务指标数据	支持	Pushgateway 接入
	Thanos Sidecar	Thanos Sidecar 仅支持查询当前实例功能，不支持写入功能，可以供 Thanos Query 查询实例数据。	支持	-
	Apache	Apache HTTP 服务监控，包括请求速率、连接数、响应代码等	支持	Apache Exporter 接入
	Graphite	Graphite 指标转换成 Prometheus 指标	支持	-
	免鉴权代理	无鉴权代理，安装后可获得无需 basic auth 鉴权的 prometheus 内网地址	支持	-

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 Prometheus 监控服务控制台。
2. 在实例列表中，选择并进入对应的 Prometheus 实例。
3. 在实例详情页，选择**数据采集 > 集成中心**。
4. 在集成中心选择支持一键安装的服务，下图以 Kafka 为例。

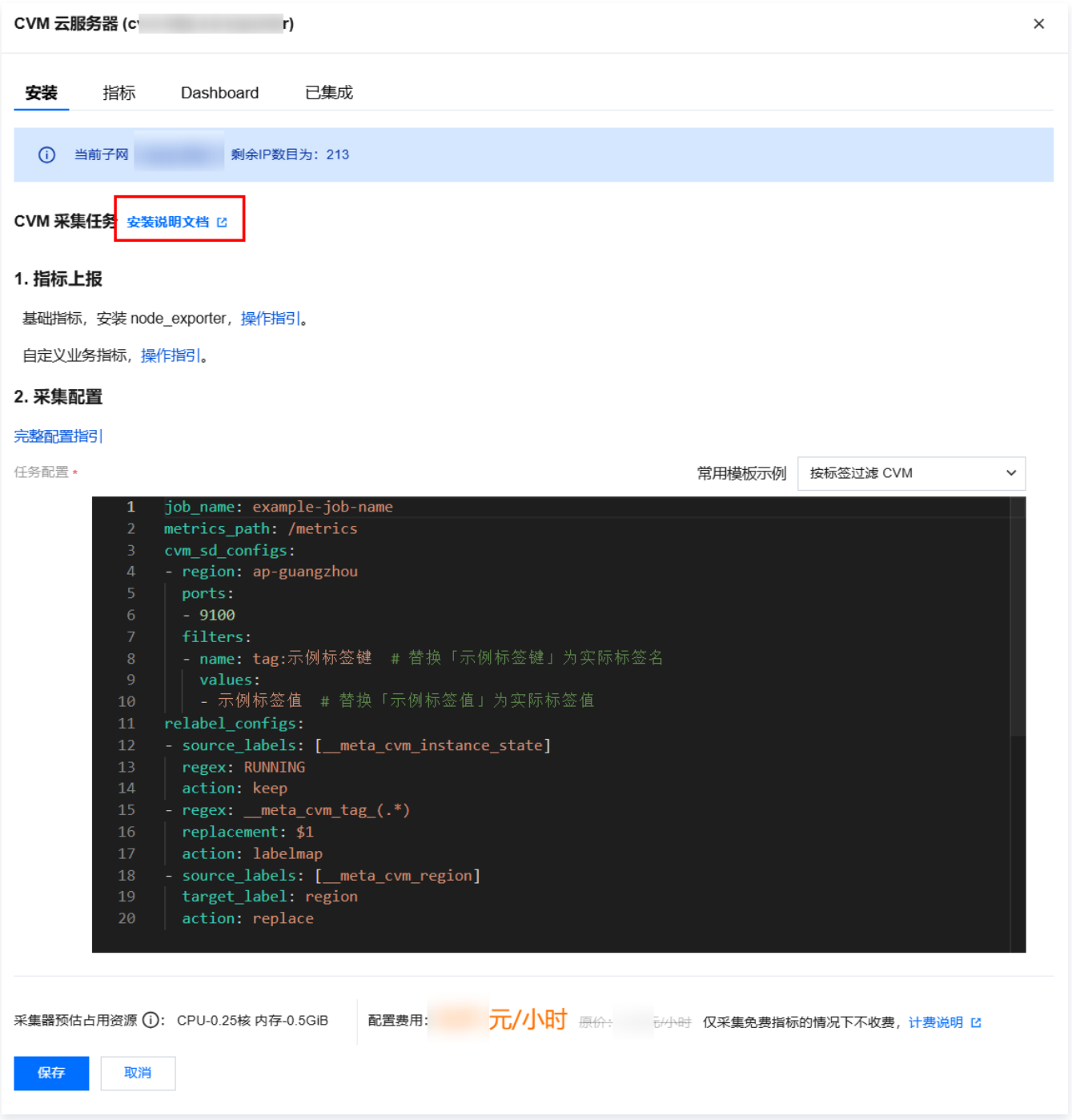


5. 单击 **Kafka** 即会弹出一个安装窗口，填写指标采集名称和地址等信息，并单击**保存**即可。

自定义安装

1. 登录 Prometheus 监控服务控制台。

2. 在实例列表中，选择并进入对应的 Prometheus 实例。
3. 在实例详情页，选择**数据采集 > 集成中心**。
4. 在集成中心选择对应的服务，下图以 **CVM 云服务器**为例。单击它即会弹出一个窗口，单击**安装说明文档**查看接入指引（仅支持部分产品），接入成功后即可实时监控对应的服务。您还可以单击 **Dashboard 操作**，安装或升级该服务的 Grafana Dashboard。



Metric Relabel 配置示例

在支持 Metric Relabel 配置的集成中，可以添加 Prometheus 原生的 metric_relabel_configs 配置，下面是常用的 metric_relabe_configs 示例：

```
metric_relabel_configs:
- action: labeldrop  # 去掉名为 labelA 的 label。regex是正则表达式，多个正则表达式用 | 分隔
  regex: labelA
- regex: ins-(.*) # 新增一个名为 id 的 label，其值通过名为 instance_id 的 label 的值经过正则处理后得到。例如 instance_id="ins-a"，新得到的 id="a"
```

```
  replacement: $1
  source_labels:
  - instance_id
  target_label: id
- target_label: region # 新增一个 region="ap-guangzhou" 的 label
  replacement: ap-guangzhou
- action: drop # 去掉名为 metricA 或 metricB 的指标
  source_labels:
  - __name__
  regex: metricA|metricB
```

数据多写

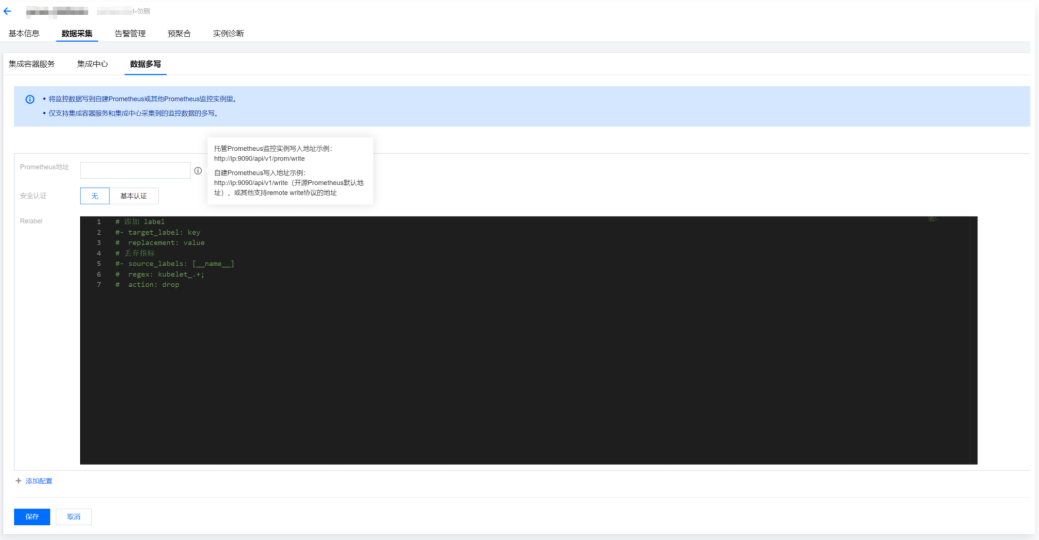
最近更新时间：2024-12-05 16:07:33

数据多写功能支持您将监控数据写到自建 Prometheus 或其他 Prometheus 监控实例中。

> **说明**
> 目前仅支持集成容器服务和集成中心采集到的监控数据的多写(通过预聚合规则计算得到的指标不支持多写)，其它暂不支持。

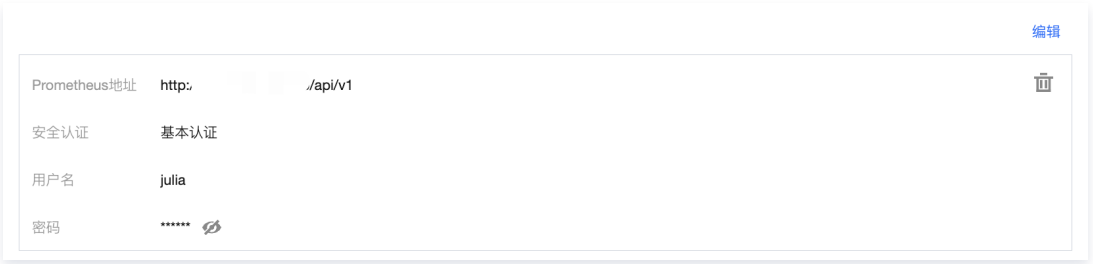
操作指南

1. 登录 腾讯云可观测平台 。
2. 在左侧导航栏中单击 **Prometheus 监控**。
3. 单击对应的实例名称，进入实例详情页，在顶部导航栏点击**数据采集** > **数据多写**。
4. 在数据多写子窗口单击**添加**，参见下列描述配置数据多写信息。



- Prometheus 地址：在实例基本信息中获取 HTTP API 地址。

- **安全验证：是否开启安全认证，开启后需要自定义用户名和密码。**



预聚合
预聚合概述

最近更新时间：2024-12-05 16:07:33

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> ]
```

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：2024-12-05 16:07:33

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。

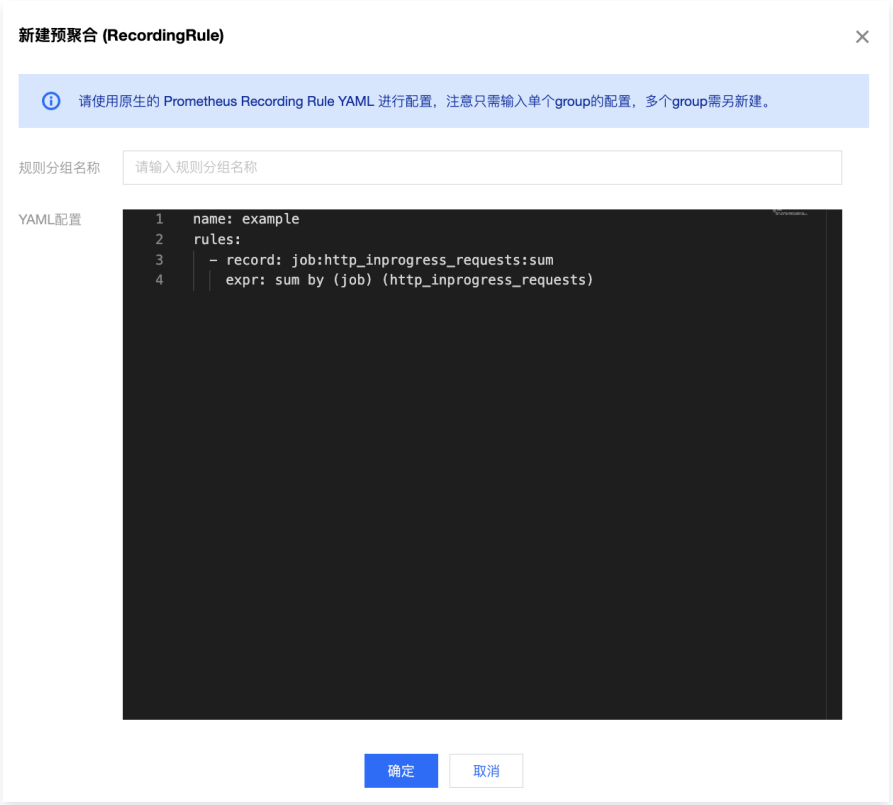
准备工作

1. 登录 腾讯云可观测平台 。
2. 在左侧菜单栏中单击 **Prometheus 监控**。
3. 创建 Prometheus 托管实例，详情请参见 创建实例 。
4. 通过实例列表进入到 Prometheus 实例的管理页面。
5. 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. 在实例管理页面，选择需要预聚合操作的实例名称，进入该实例详情页。
2. 在顶部导航栏中单击**预聚合** > **新建预聚合**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 。



以下为一个简单预聚合规则例子：

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

3. 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：2024-12-05 16:07:33

开启默认预聚合规则的情况下，会自动生成默认预聚合规则指标。默认预聚合规则指标用于常用监控指标的预设 Dashboard 图表展示与告警规则模板，正常计费。默认预聚合规则的启用状态是 Prometheus 实例维度的，可前往 Prometheus 监控 的实例详情页，在预聚合页面修改启用状态。

默认预聚合规则指标列表如下：

metric（指标名称）	容器控制台预设 Dashboard	Grafana 平台预设 Dashboard	告警规则模板
:node_memory_MemAvailable_bytes:sum	集群监控概览-Memory Utilisation	Kubernetes / Compute Resources / Cluster	-
node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	集群监控概览-CPU Usage	Kubernetes / Compute Resources / Cluster Kubernetes / Compute Resources / Workload Kubernetes / Compute Resources / Pod Kubernetes / Compute Resources / Node (Pods) Kubernetes / Compute Resources / Namespace (Workloads) Kubernetes / Compute Resources / Namespace (Pods) Kubernetes / Networking / Workload Kubernetes / Networking / Namespace (Workload)	-
	集群监控概览-CPU Quota		-
	集群 Namespace大盘-CPU Quota		-
	节点 Pod 监控-CPU Usage		-
	节点 Pod 监控-CPU Quota		-
	工作负载监控概览-CPU Usage		-
	工作负载监控概览-CPU Quota		-
	DaemonSet-CPU Usage		-
	DaemonSet-CPU Quota		-
	集群 Pod 监控-CPU Quota		-
	StatefulSet-CPU Usage		-
	StatefulSet-CPU Quota		-
	Deployment-CPU Usage		-
	Deployment-CPU Quota		-
node_namespace_pod_container:container_memory_working_set_bytes	节点 Pod 监控-Memory Usage		-
	节点 Pod 监控-Memory Quota		-
node_namespace_pod_container:container_memory_rss	-		-
node_namespace_pod_container:container_memory_cache	-		-

node_namespace_pod_container:container_memory_swap	-		-
namespace_workload_pod:kube_pod_owner:relabel	集群监控概览-CPU Quota		-
	集群监控概览-Memory Requests		-
	工作负载监控概览-CPU Usage		-
	工作负载监控概览-CPU Quota		-
	工作负载监控概览-Memory Usage		-
	工作负载监控概览-Memory Quota		-
	DaemonSet-CPU Usage		-
	DaemonSet-CPU Quota		-
	DaemonSet-Memory Usage		-
	DaemonSet-Memory Quota		-
	命名空间工作负载网络监控-Current Rate of Bytes Received		-
	命名空间工作负载网络监控-Current Rate of Bytes Transmitted		-
	命名空间工作负载网络监控-Current Status		-
	命名空间工作负载网络监控-Average Rate of Bytes Received		-
	命名空间工作负载网络监控-Average Rate of Bytes Transmitted		-
	命名空间工作负载网络监控-Receive Bandwidth		-
	命名空间工作负载网络监控-Transmit Bandwidth		-
	命名空间工作负载网络监控-Rate of Received Packets		-
	命名空间工作负载网络监控-Rate of Transmitted Packets		-
	命名空间工作负载网络监控-Rate of Received Packets Dropped		-
	命名空间工作负载网络监控-Rate of Transmitted Packets Dropped		-
	工作负载网络监控-Current Rate of Bytes Received		-
	工作负载网络监控-Current Rate of Bytes Transmitted		-
	工作负载网络监控-Average Rate of Bytes Received		-

	工作负载网络监控-Average Rate of Bytes Transmitted		-
	工作负载网络监控-Receive Bandwidth		-
	工作负载网络监控-Transmit Bandwidth		-
	工作负载网络监控-Rate of Received Packets		-
	工作负载网络监控-Rate of Transmitted Packets		-
	工作负载网络监控-Rate of Received Packets Dropped		-
	工作负载网络监控-Rate of Transmitted Packets Dropped		-
	StatefulSet-CPU Usage		-
	StatefulSet-CPU Quota		-
	StatefulSet-Memory Usage		-
	StatefulSet-Memory Quota		-
	Deployment-CPU Usage		-
	Deployment-CPU Quota		-
	Deployment-Memory Usage		-
	Deployment-Memory Quota		-
namespace:kube_pod_container_resource_requests_memory_bytes:sum	_	_	Kubernetes 资源
namespace:kube_pod_container_resource_requests_cpu_cores:sum	-	_	Kubernetes 资源

实例诊断

最近更新时间：2024-12-05 16:46:32

背景

为了提升客户在使用 Prometheus 监控采集端的体验，现提供了新的采集端架构，升级后的采集端新架构支持实例诊断、系统健康检查，并提升了采集 Agent 资源利用率和指标采集稳定性。本文将指引用户通过控制台实例诊断页面将旧采集架构升级到新采集架构，并获取关于当前实例采集和存储的细节信息，以获得更优质的使用体验。

操作步骤

升级新架构

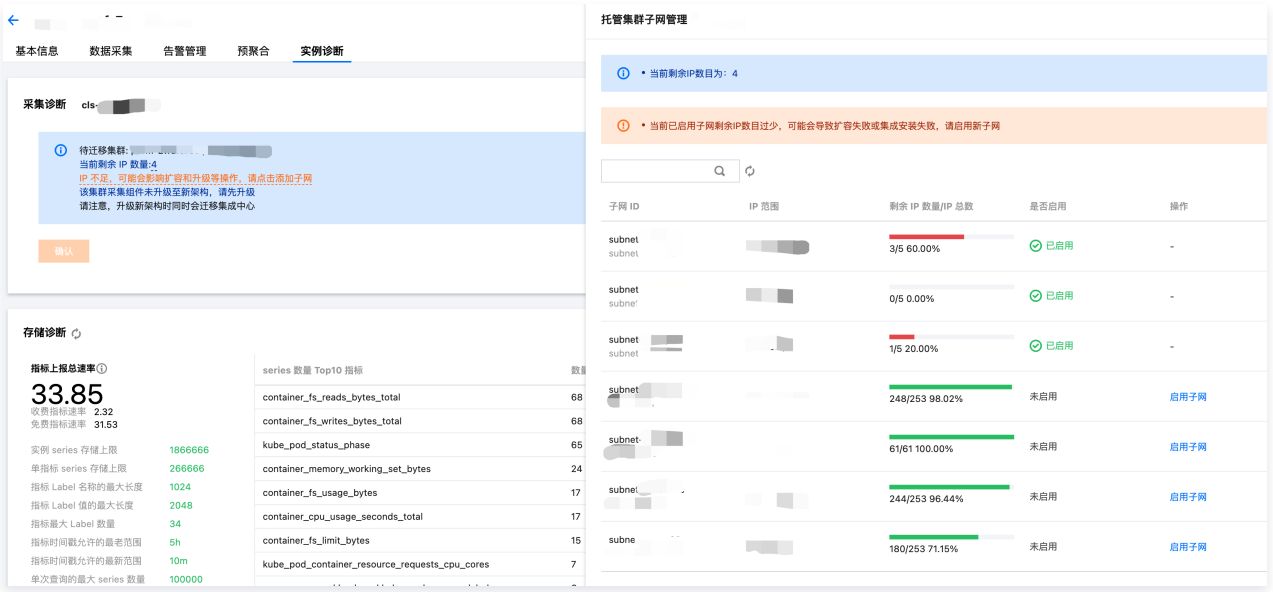
1. 登录 Prometheus 监控服务控制台。
2. 在 Prometheus 实例列表中，单击**实例 ID/名称**。
3. 进入 Prometheus 管理中心，在顶部导航栏中单击**实例诊断**，并选择对应的**采集集群**。
- 点击**确认**即可升级新架构。

> **注意：**
> 升级过程预计需要5分钟，可能存在1-2分钟的指标断点。

- 若实例的托管集群中 IP 数量少于10个会出现风险提示，为防止由于 IP 数量不足问题导致组件无法正常升级，请根据下文指引增加可用 IP 数量。

添加子网

1. 在实例诊断页面中点击子网信息，进入**托管集群子网管理**页面。



2. 页面当前 VPC 内已经添加到托管集群内的子网和未添加的子网，可根据规划点击**启用子网**启用剩余 IP 足够的子网。
3. 若没有可用子网，请先 添加子网 后再在当前页面中启用。

实例诊断

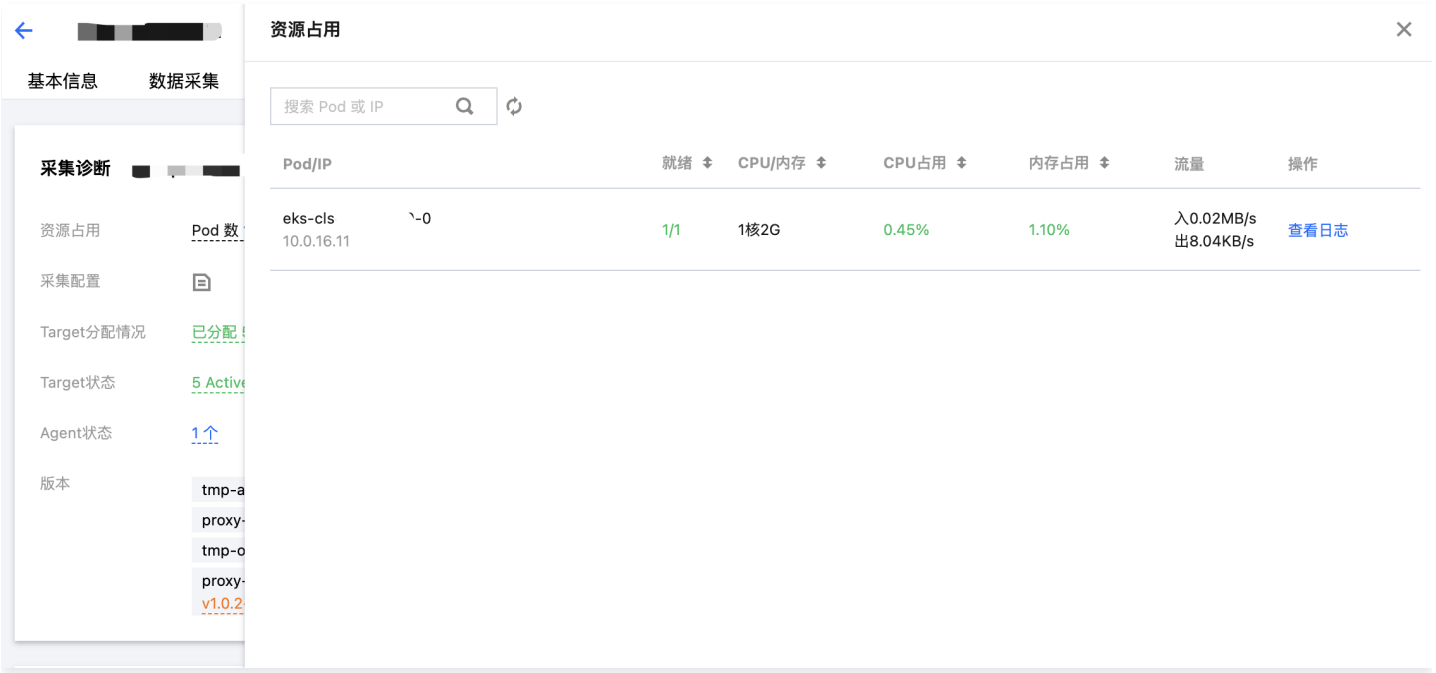
架构升级后，实例诊断页面包括采集和存储两个部分的内容，有助于用户了解 Prometheus 采集和存储的运行情况，从而更加快速地定位存在的问题。

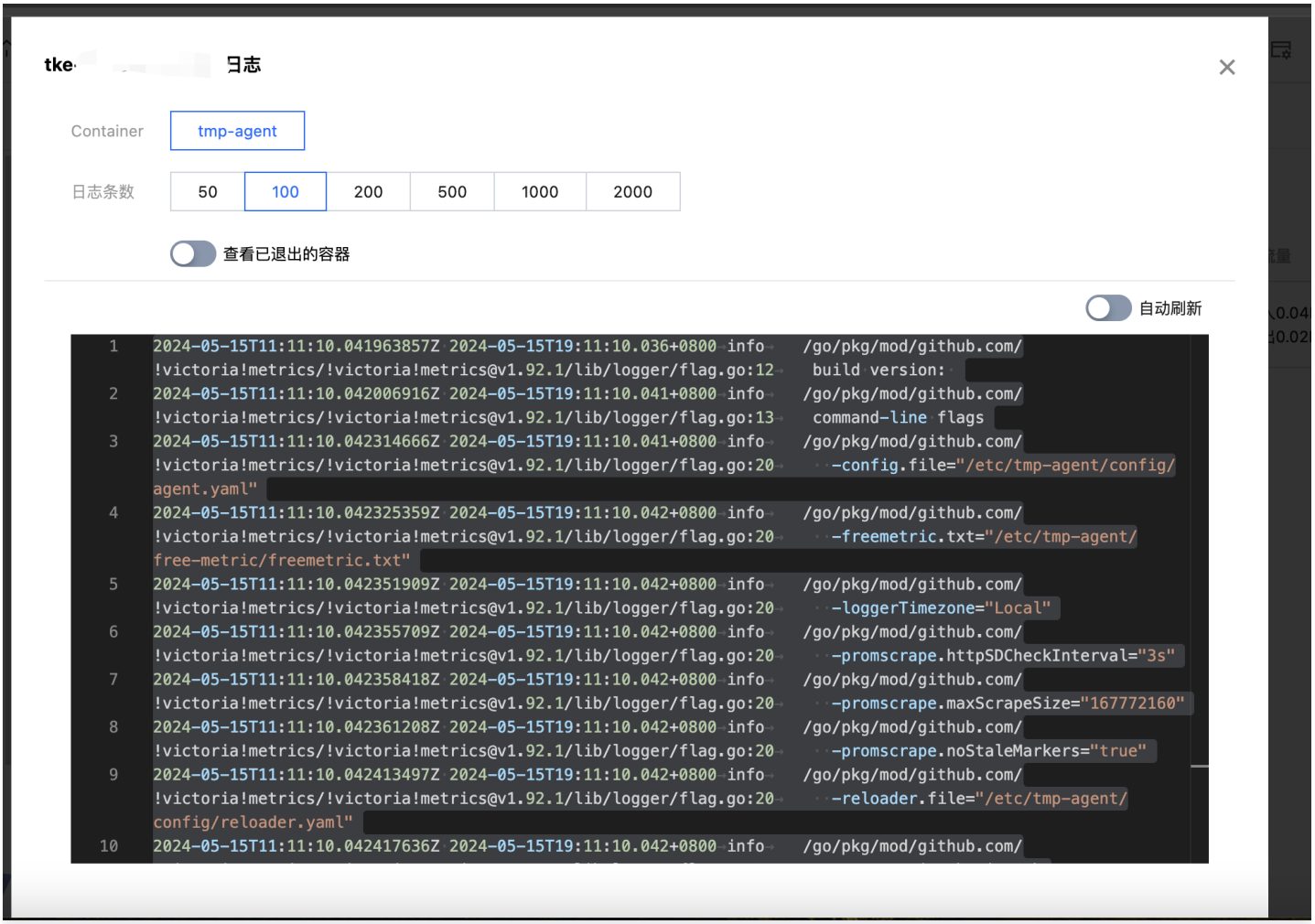
采集诊断

如图所示，采集诊断包括对应采集的资源占用、采集配置、Target 分配情况、Target 状态、Agent 状态、组件版本以及采集的架构图。

资源占用

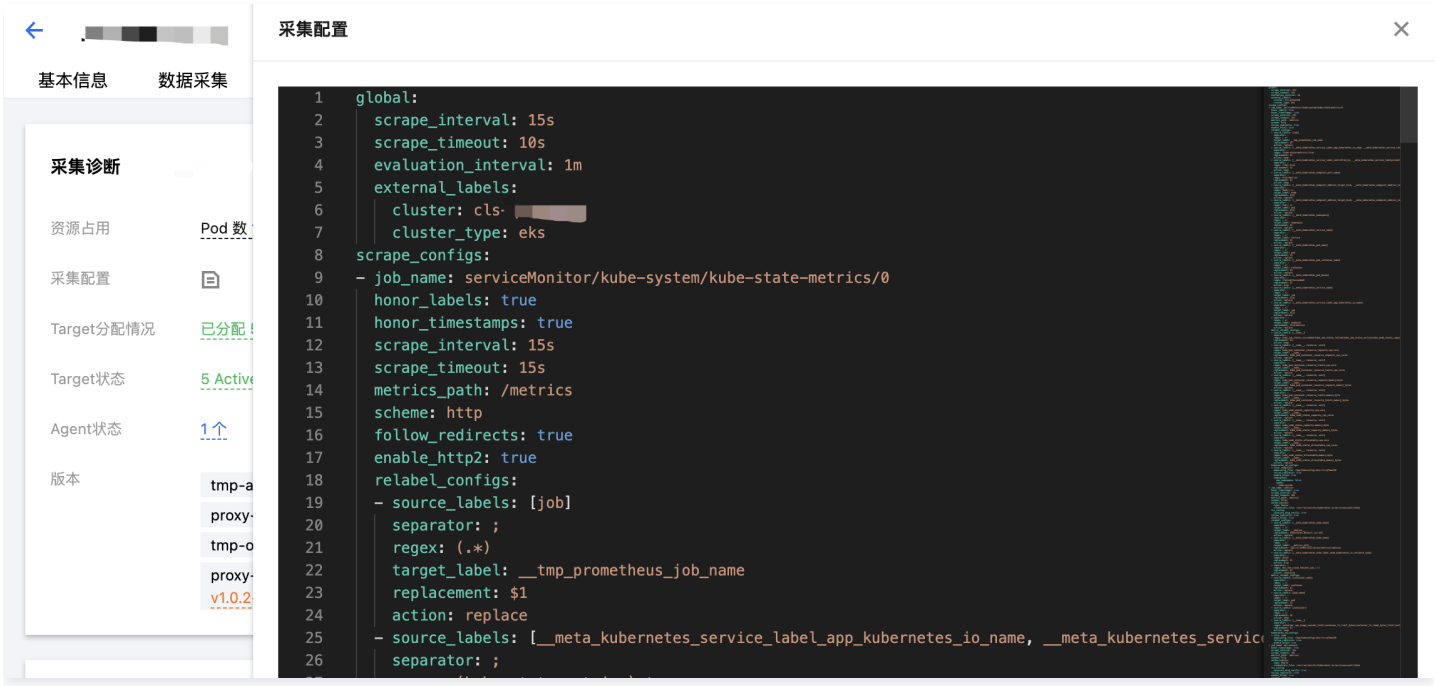
采集分片的资源占用情况，展示了包括 CPU、内存上限和占用以及出入流量等信息；点击**查看日志**可以看到该 Pod 的日志，便于查看运行细节和排查异常问题。





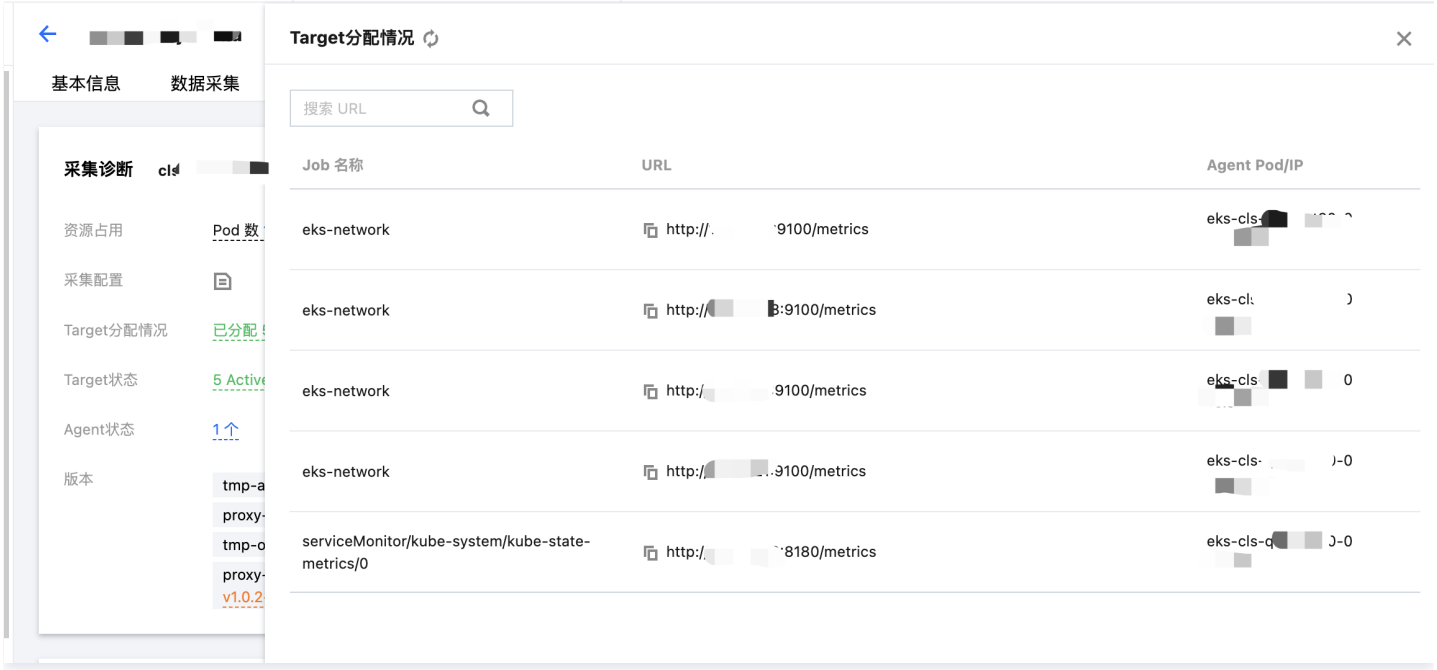
采集配置

展示了当前采集的具体采集配置。



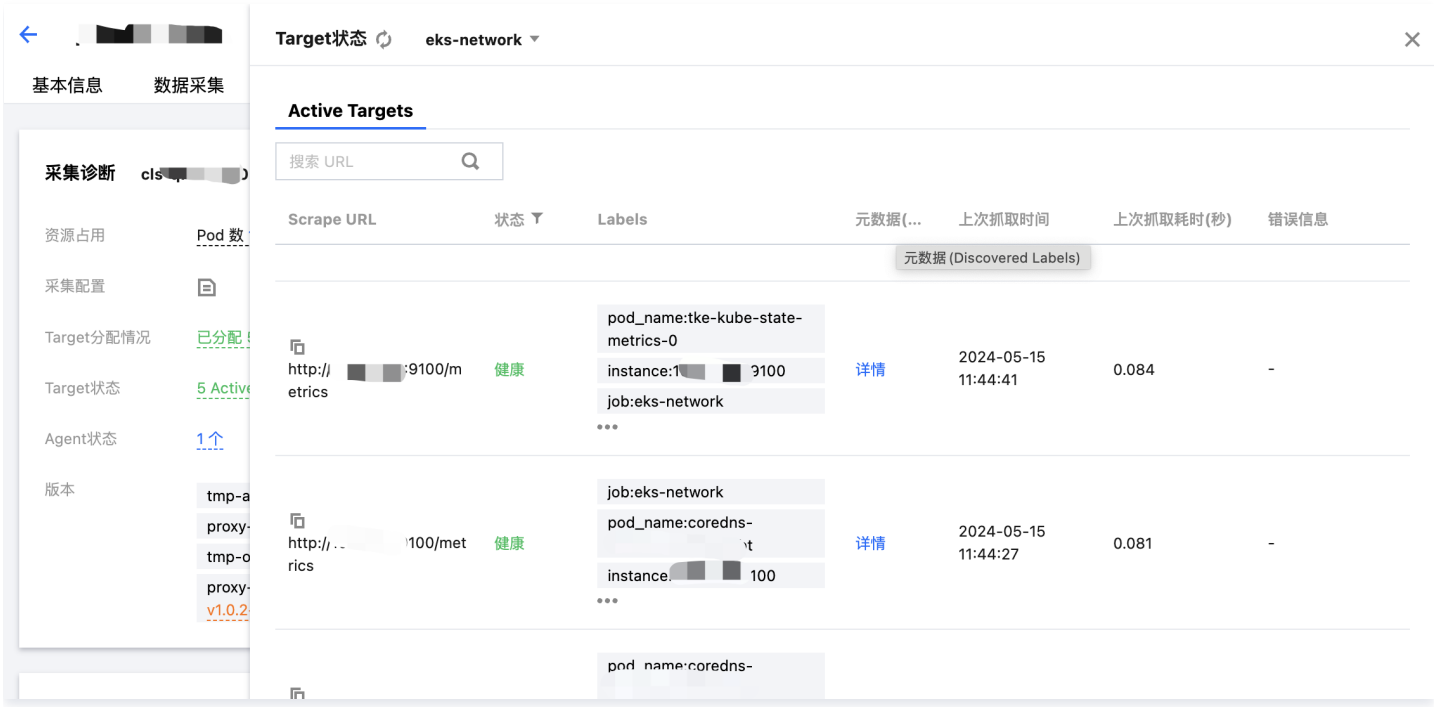
Target 分配情况

展示了采集目标 Target 的 URL、Target 所属的采集任务 Job 名称，以及目前由哪个采集分片进行采集。



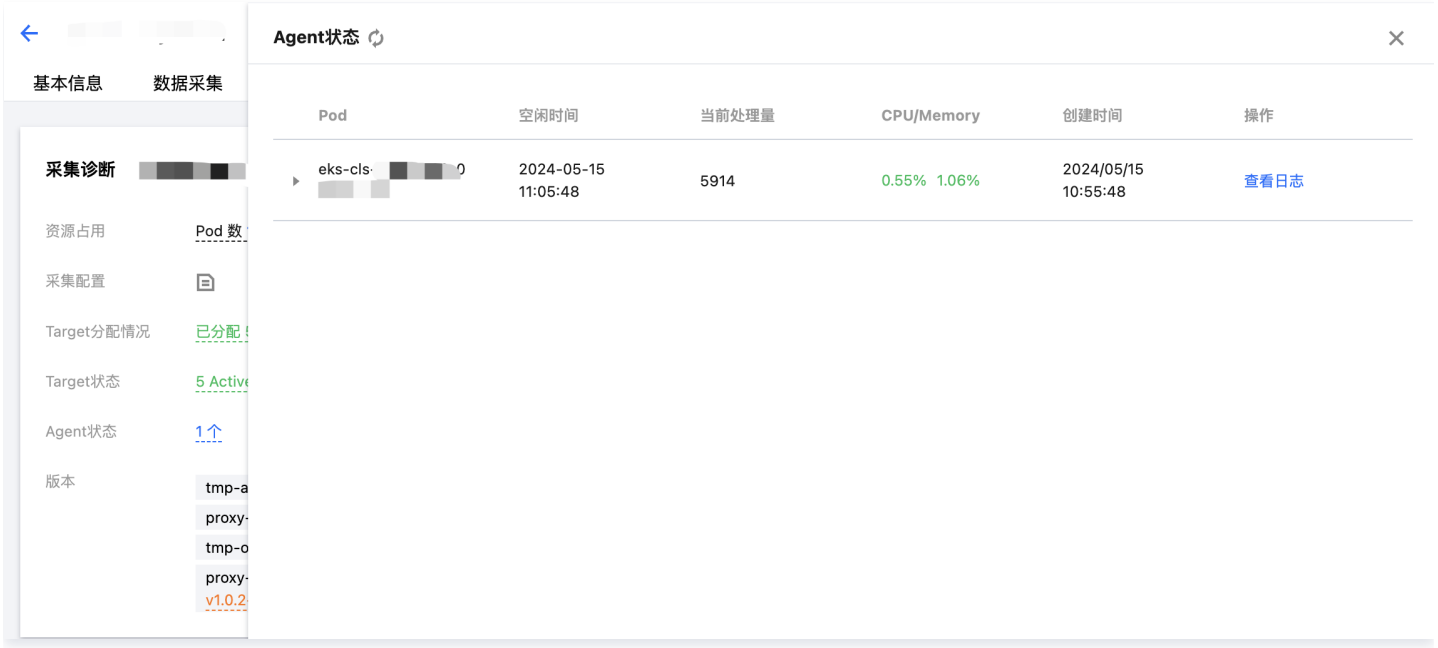
Target 状态

可根据采集任务 Job 筛选活跃的采集目标 Targets，并获取对应 Target 的状态、Labels、Discovered Labels 等信息。Target 状态在正常情况下是“健康”，若为“异常”且已经有过上次抓取时间，可根据最右边的错误信息进行排查，常见的问题可能为 Target 本身不健康、权限配置错误、网络错误等等。



Agent 状态

展示了采集分片 Agent 的运行状态；点击**查看日志**可以看到该 Pod 的日志以了解运行细节和排查异常问题。



组件版本

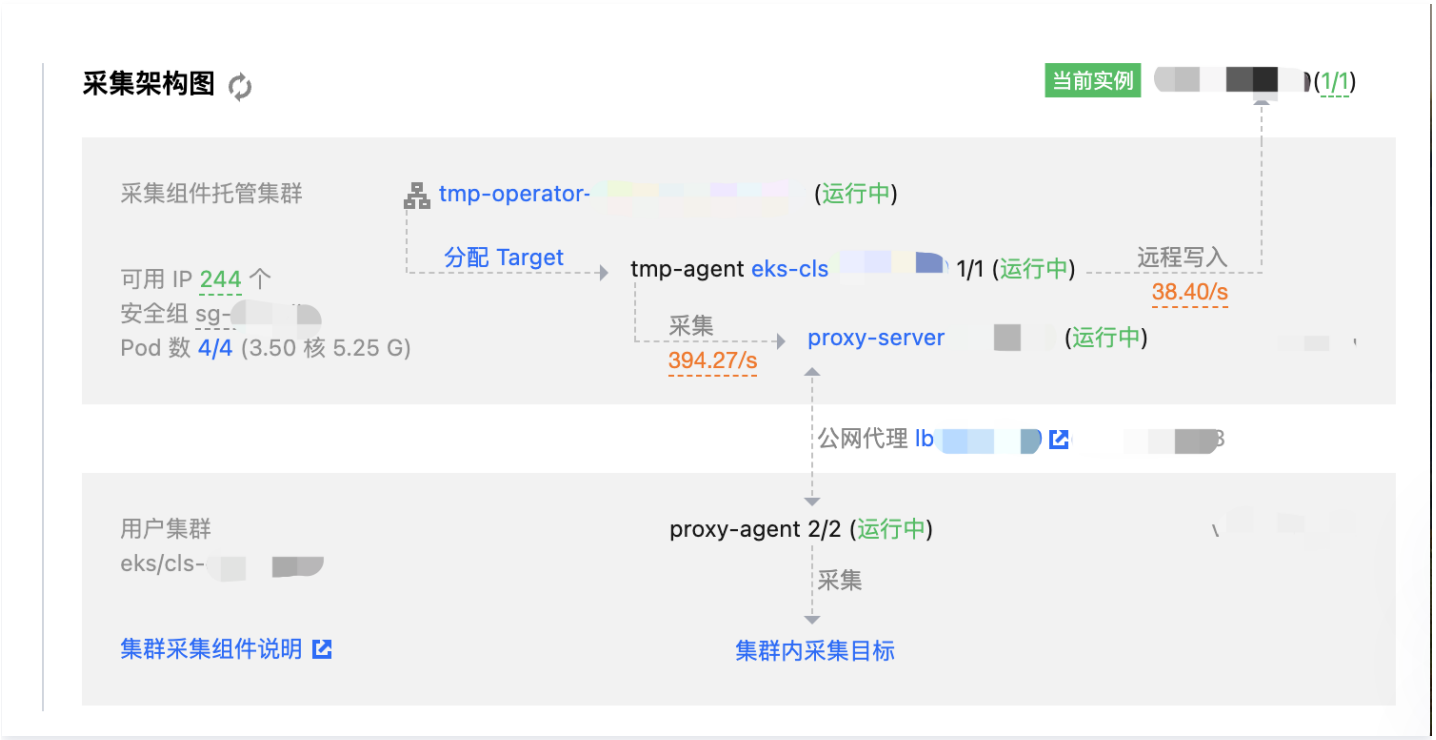
当前采集的组件版本信息，点击进入**组件版本**页面，包括 IP 数量检查和各组件版本的当前版本、最新版本和组件描述、升级描述等信息。

说明：

- 采集组件请尽量保持**最新版本**，可通过对应组件的**升级**进行升级；
- 组件 tmp-operator、tmp-agent、proxy-agent 正常情况下能够无影响升级；
- 组件 proxy-server 在升级过程中会出现采集断点，断点时长为 eks 拉起组件 Pod 的时间，并且影响范围是整个 Prometheus 实例的采集（包括集成中心和容器集群），请谨慎操作。

采集架构图

通过采集架构图可以了解当前的采集架构信息。



采集组件托管集群：

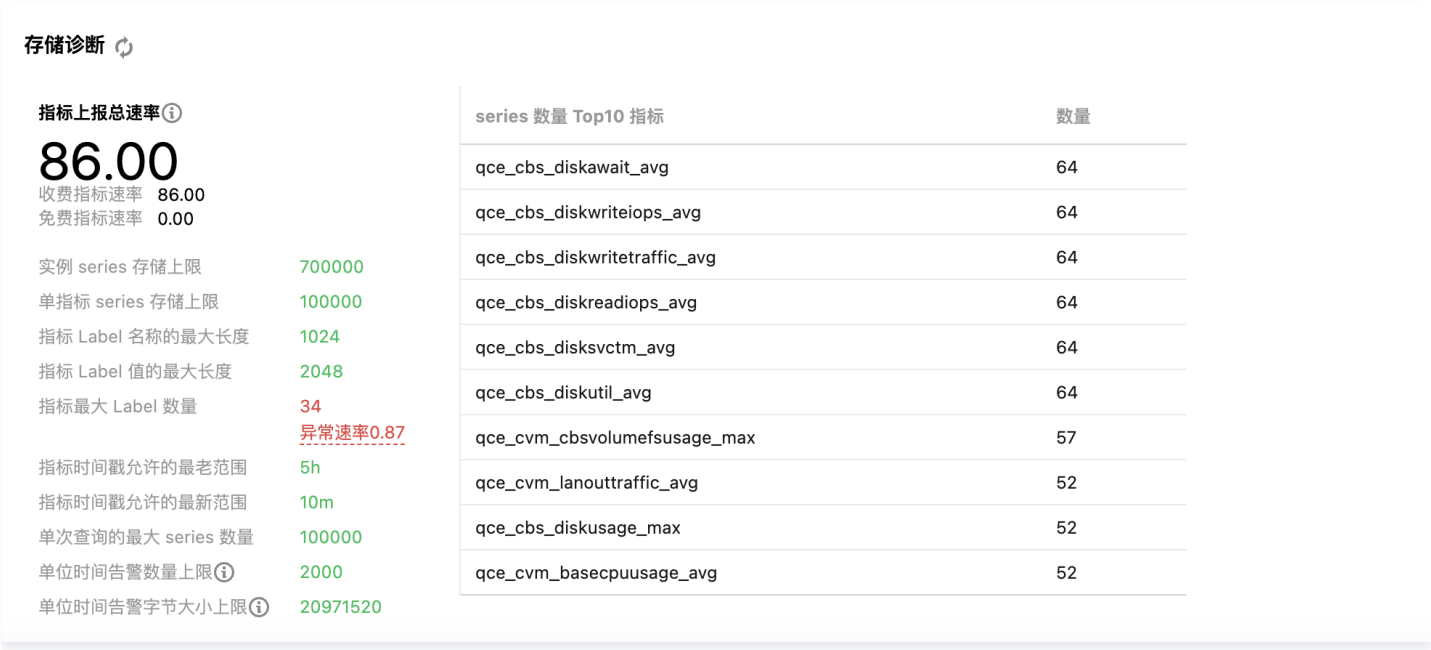
- 可用 IP 数量，当 IP 数量不足时会出现 IP 数量不足的提示，点击进入**托管集群子网管理**页面，具体操作可参考前面 添加子网；
- 托管集群的安全组，及其正常运行的放通要求，采集出现的部分网络问题可能是因为安全组未放通；
- 当前采集相关的 Pod 数量及资源使用情况；
- 采集调度组件 tmp-operator 的运行状态；
- 采集目标 Target 分配情况；
- 采集分片组件 tmp-agent 的运行状态；
- 指标采集速率及入带宽；
- 代理组件 proxy-server 的运行状态；
- 指标写入速率及出带宽；
- 写入目标状态，包括当前实例对应的存储和用户配置的数据多写。

用户集群：

- 公网代理 CLB 的状态（若开启）；
- 代理组件 proxy-agent 的运行状态；
- 集群内采集目标 Target 的状态。

存储诊断

如图所示，存储诊断包括了存储相关的状态和限制，具体限制可参考 相关限制 。



参数说明

参数	说明
指标上报总速率	包括免费指标和收费指标。
实例 series 存储上限	活跃的 series 数量超过该值会因为超限丢弃。
单指标 series 存储上限	相同指标名不同 label 是不同的 series，超过该值会因为超限丢弃。
指标标签 Label 名称的最大长度	超过该值会因为不合法而丢弃。
指标最大 Label 数量	超过该值会因为不合法而丢弃。
指标时间戳允许的最老范围	单条 series 中可以接受的最老时间戳（不允许乱序）。

指标时间戳允许的最新范围	单条 series 中可以接受的最新时间戳（不允许乱序）。
单次查询的最大 series 数量	查询数据时最大涉及的 series 数量，建议缩短范围查询 Range Query 的时间，或在合适的场景使用即时查询 Instant Query。
单位时间告警数量上限	5分钟内触发告警数量上限。
单位时间告警字节大小上限	5分钟内触发告警的标签、Annotation 等字段总的大小上限。
series 数量 Top10 指标	同一指标名称，不同标签 Label 的键值都是不同的 series，存储收到单指标 series 存储上限限制，数量过大会产生高基数问题。

归档存储

最近更新时间：2025-05-19 15:41:32

归档存储是一种数据存储解决方案，专门用于长期保存不常访问的数据，它通常用于存储那些需要保留但不需要频繁访问的信息。为了满足用户对数据的长期保留和业务需求，并在最大程度上为用户减少成本，Prometheus 特推出了归档存储功能，下面将详细介绍该功能的使用方法。

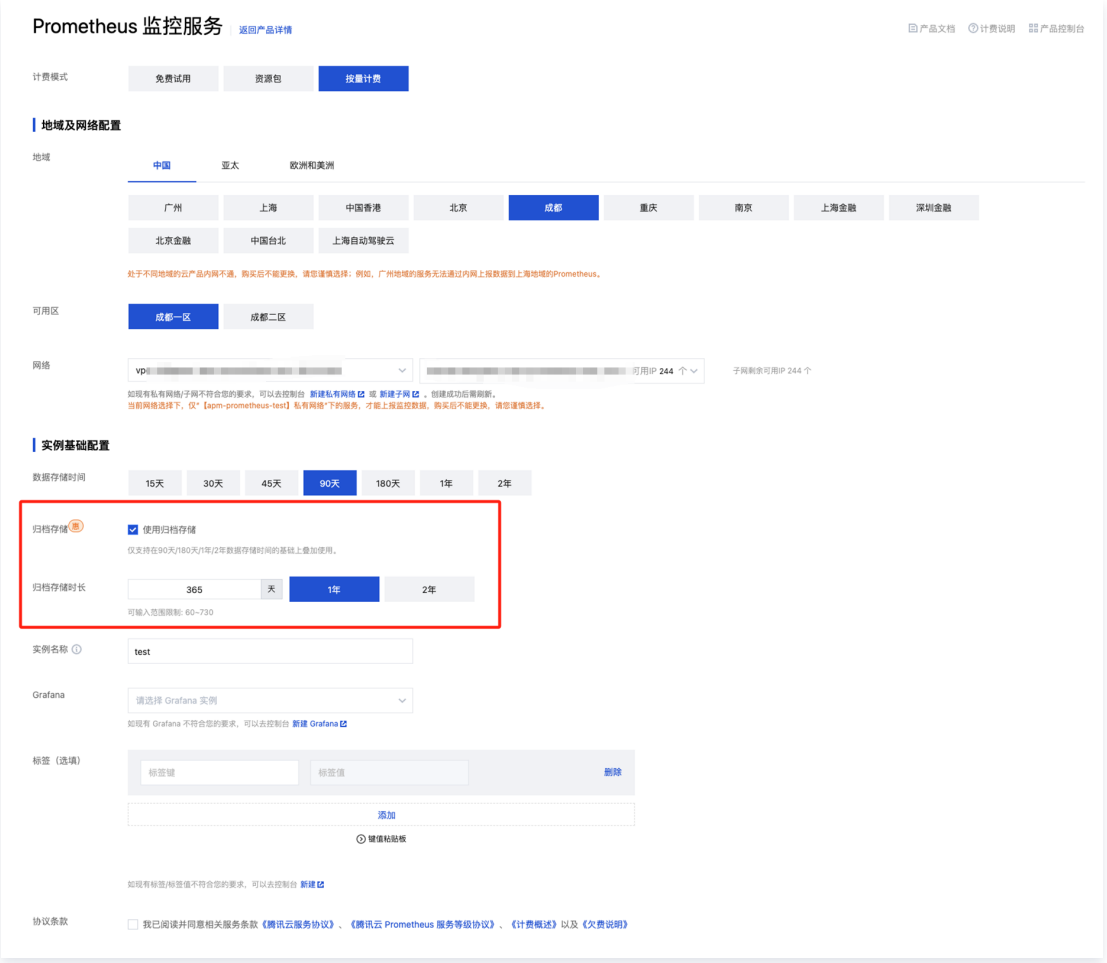
使用限制

- **规格限制**：规格为存储90天、180天、1年、2年的实例，才允许额外配置归档存储。
- **存储时间限制**：归档存储的时间范围，支持自定义输入，可配置的最小值为60天，最大值为2年。
- **查询频率限制**：查询的数据范围若包含归档存储数据，系统会限制查询请求频率为每秒最多3次。在无查询的情况下，系统会“冷却”并积累可用额度，最多可累积15次查询配额（约等于5秒的空闲时间），因此，您可以在短时间内一次性发起最多15个查询。（例子：若前5秒无查询请求，第6秒可以一次性发起15个查询；之后将恢复为每秒最多3次的速率。）

操作步骤

方法一：在实例购买页配置

1. 登录 腾讯云可观测平台 。
2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 在 Prometheus 监控页面单击**新建**，若**数据存储时间**选择了90天、180天、1年、2年，则在该配置项下方有**归档存储**项。

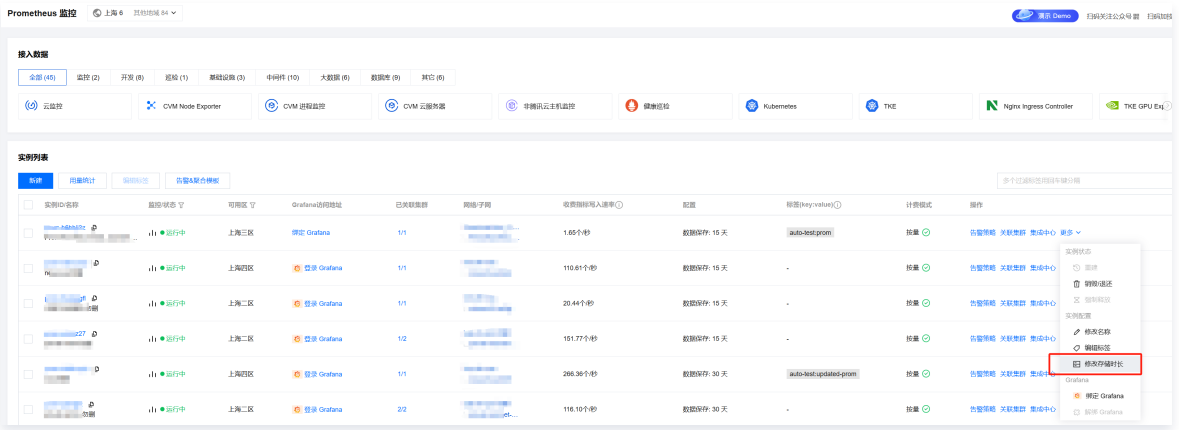


4. 归档存储支持自定义输入存储时长，输入范围限制为60 – 730天。其他配置项按照页面提示填写信息，配置完后，勾选服务条款并单击**立即购买**支付即可。

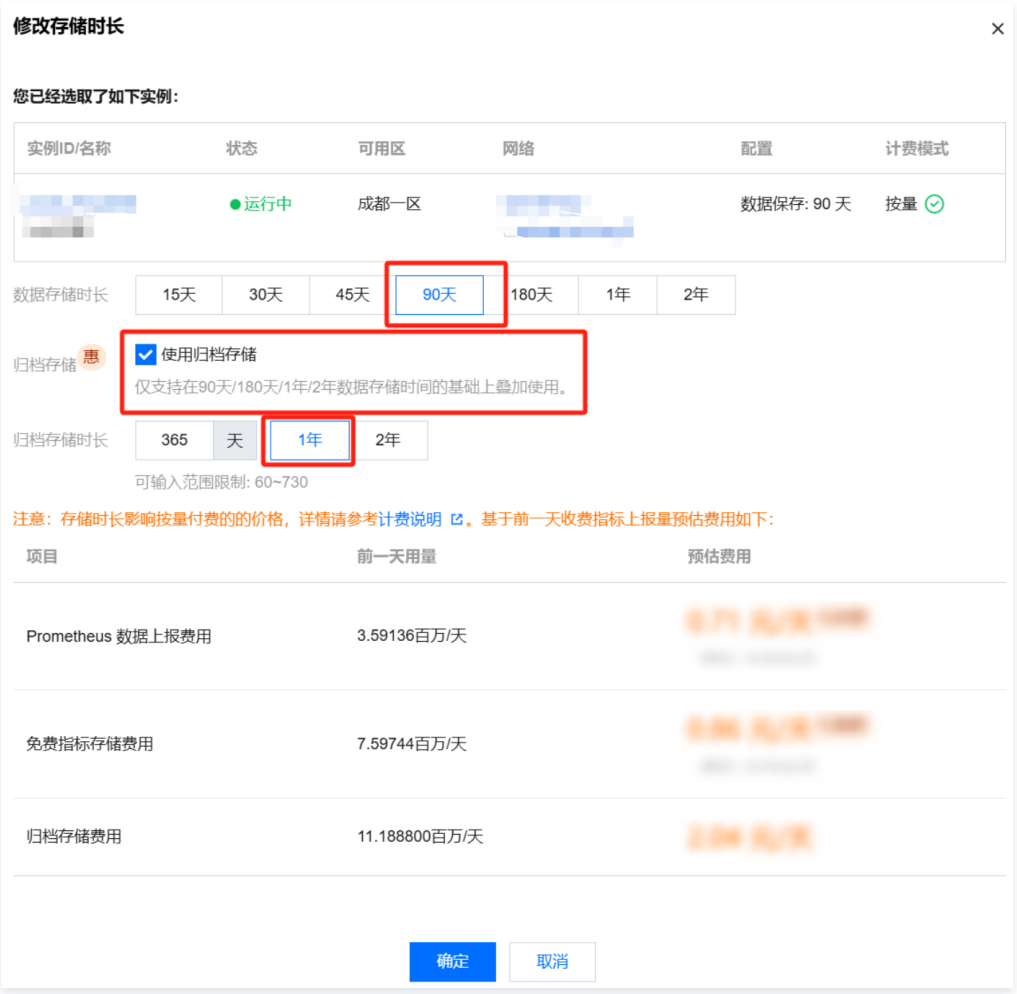
方法二：在实例列表页配置

1. 登录 腾讯云可观测平台 。

2. 在左侧菜单栏中选择 **Prometheus 监控**。
3. 在实例列表中，选择需要修改存储时长的 Prometheus 实例，单击右侧的**更多 > 修改存储时长**。

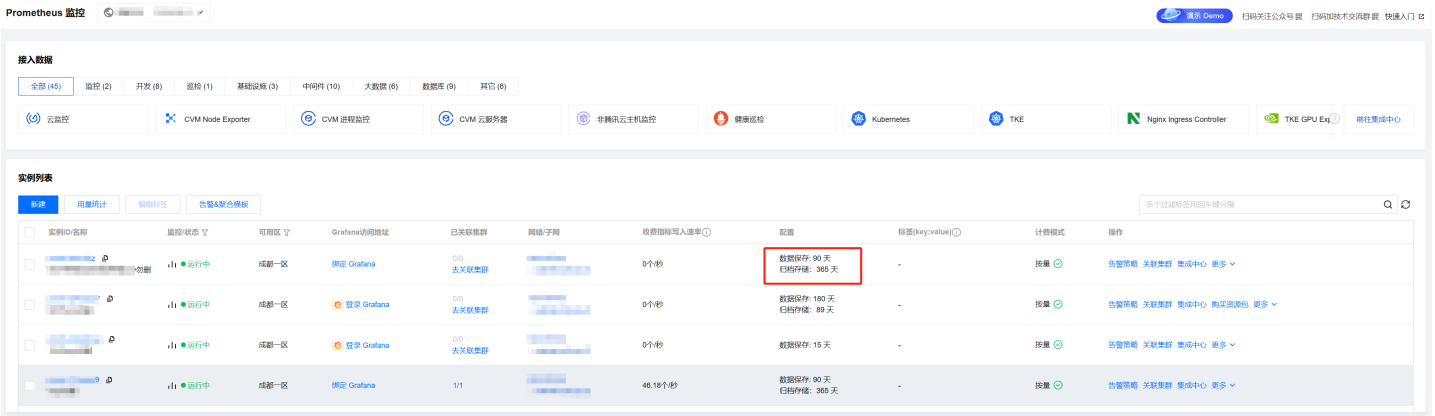


4. 当数据存储时长修改为90天、180天、1年、2年时，可勾选下方的**归档存储**，并自定义输入归档存储时长，修改完单击**确定**即可，如下图所示。



配置结果展示

按照上面任意一个方法配置成功后，返回实例列表页可以看到配置下方增加了一个归档存储天数展示，如下图所示：



附加说明

正常存储与增加归档存储的对比如下：

存储方式	价格	查询频率限制	参考文档
数据存储	高	无	按量付费介绍
数据存储+归档存储	低	有	归档存储付费介绍

告警策略
告警策略概述

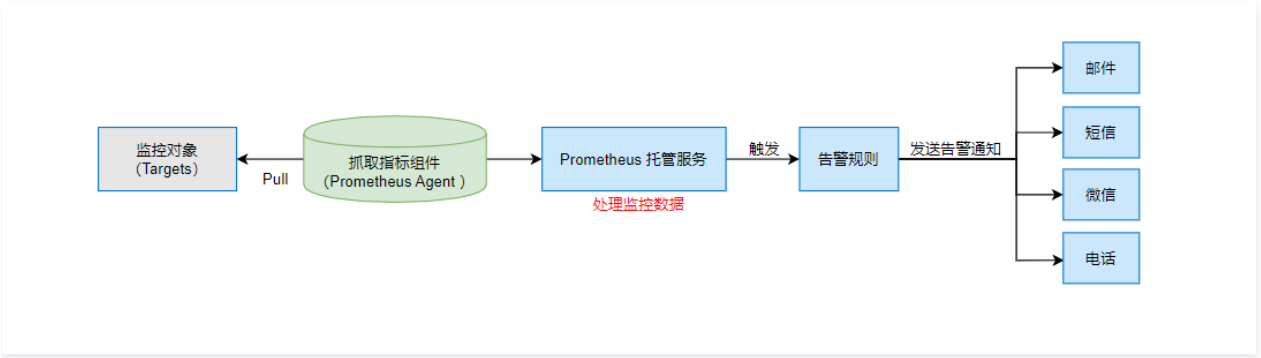
最近更新时间：2024-12-05 16:07:33

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、微信、短信、电话告警渠道通知您采取措施。

> **说明：**
> Prometheus 监控服务告警结合了腾讯云可观测平台告警能力和 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。
- 使用腾讯云可观测平台告警通知模板，支持邮件、短信、电话多种接收方式。



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：2024-12-05 16:07:33

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 for more than 5
minutes."
```

```
  # 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 (current value: {{
$value }}s)"
```

新建告警策略

最近更新时间：2025-05-30 14:14:31

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。

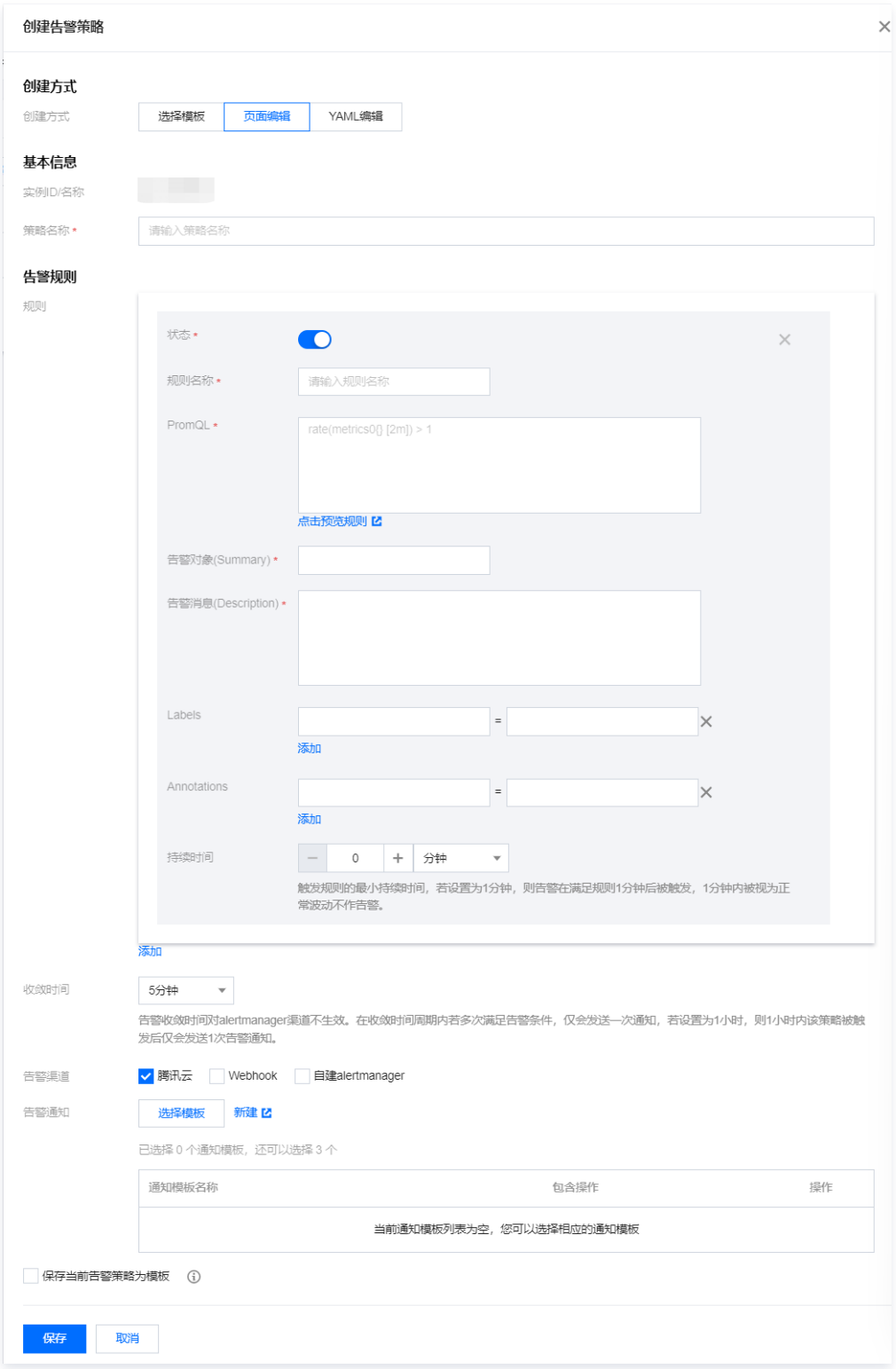
前提条件

- 已创建 Prometheus 实例 。
- 已 关联集群 或已安装 集成中心 的数据集成。

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，单击顶部菜单栏的**告警管理** ，进入**告警策略**页面。
3. 在告警策略页中单击**新建告警策略**，在弹框中配置告警策略信息。
4. 创建方式：有**选择模板**、**页面编辑**、**YAML 编辑**三种创建方式，下面以**页面编辑**示例。
   - **实例 ID/名称**： 默认显示，无需填写。
   - **策略名称**：自定义名称。
   - **规则**：告警规则组，支持添加多个规则。
     - **规则名称**：自定义名称。
     - **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。
     - **告警对象(Summary)**：允许用户自定义告警标题，支持通过 {{ $labels.<labelname> }} 变量访问当前告警实例中指定标签的值，详情请参见 模板说明 。
     - **告警消息(Description)**：可使用默认模板也可自定义，允许用户自定义告警内容。支持通过 $labels.<labelname> 变量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，详情请参见 模板说明 。
     - **Labels**：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。
     - **Annotations**：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。
     - **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。
   - **收敛时间**：默认每5分钟告警一次，您也可以自定义告警通知频率。
   - **告警渠道**：可选择腾讯云告警渠道、webhook 告警渠道或自建 alertmanager 告警渠道。
   - **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 。
   - **保存当前告警策略为模板**：支持将当前页面配置的告警策略内容保存为模板，后续可引用该模板快速创建告警。



模板说明

Prometheus 支持模板化 label 和 annotations 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，其中 labelname 为指定标签的名称。{{ $value }} 则可以获取当前 PromQL 表达式计算的样本值，可通过模板函数对 {{ $value }} 进行多种格式转换，以提升数值的可读性，以下是常见的转换方式及示例。

- 基本数值格式化

```
{{ $value }}               // 原始值（如 123.456）
{{ printf "%.2f" $value }} // 保留两位小数（如 123.46）
{{ printf "%.0f" $value }} // 取整（如 123）
```

```
{{ printf "%.1e" $value }} // 科学计数法（如 1.2e+02）
{{ printf "%.2f%%" $value }} // 转换为百分比（如 123.46%）
{{ $value | printf "%.2f%%" }} // 等价写法
```

- 数据单位转换

```
// 字节单位（B, KB, MB, GB 等）
{{ $value | humanize1024 }} // 自动转换字节单位（如 1048576 → "1.0 MB"）
{{ $value | humanize1024 | printf "%.2f" }} // 保留两位小数（如 "1.00 MB"）

// 十进制单位（k, M, G 等）
{{ $value | humanize }} // 自动转换十进制单位（如 1000000 → "1.0 M"）

// 时间单位（s, ms, µs 等）
{{ $value | humanizeDuration }} // 自动转换时间单位（如 0.001 → "1 ms"）

// 百分比格式化转换
{{ $value | humanizePercentage }} // 自动转换为百分比格式（如 0.12345 → "12.3%"）
```

例如，某比率类指标取值范围为 [0,1]，在告警内容中可使用 {{ $value | humanizePercentage }} 转换为百分比格式。如果某比率类指标取值范围为 [0,100]，在告警内容中可使用 {{ printf "%.2f" $value }} 转换为保留两位小数的百分比格式。与此同时，如果指标包含一个名为 instance 的标签表示实例的 IP，在告警内容和告警对象中可使用 {{ $labels.instance }} 来获取标签的值。对应控制台中告警规则的配置如下：

收到的告警通知中，告警对象将附带触发告警的实例 IP，告警消息中将附带触发告警的实例 IP 和当前样本值，具体信息如下：

告警状态 State	告警对象 Summary	告警消息 Description
Active	实例10.0.0.[redacted] CPU使用率过高	主机 CPU 使用率过高。实例ID: 10.0.0.[redacted]，当前值: 1.228%。
Active	实例10.0.0.[redacted] CPU使用率过高	主机 CPU 使用率过高。实例ID: 10.0.0.[redacted]，当前值: 4.633%。

暂停告警策略

最近更新时间：2024-12-05 16:07:33

本文指导您在 Prometheus 监控服务控制台中，如何暂停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，单击顶部导航栏的**告警管理**，进入**告警策略**页面。
3. 在告警策略页面，找到需要关闭的告警策略，在操作列表中单击**全部暂停**。
4. 在弹框中单击**确定**即可。

策略类型说明（旧）

最近更新时间：2024-12-05 16:07:33

该策略说明仅适用于旧的包年包月类型的 Prometheus 实例。按量付费类型和资源包类型的 Prometheus 实例，请参考控制台的模板管理中的默认模板内容。
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	(sum(rate(rest_client_requests_total{code=~"5.."}[5m])) by (cluster,instance, job) /sum(rate(rest_client_requests_total[5m])) by (cluster,instance, job))> 0.01	15m	客户端访问 APIServer 出错率大于1%
客户端访问 APIServer 证书快过期	apiserver_client_certificate_expiration_seconds_count{job="apiserver"} > 0 and on(job) histogram_quantile(0.01, sum by (cluster, job, le) (rate(apiserver_client_certificate_expiration_seconds_bucket{job="apiserver"}[5m]))) < 86400	无	访问 APIServer 的客户端证书将在24小时后过期
聚合 API 出错	sum by(cluster, name, namespace) (increase(aggregator_unavailable_apiservice_count[5m])) > 2	无	聚合 API 最近5分钟报错
聚合 API 可用性低	(1 − max by(name, namespace, cluster) (avg_over_time(aggregator_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	absent(sum(up{job="apiserver"}) by (cluster) > 0)	5m	APIServer 从采集目标中消失
Scheduler 故障	absent(sum(up{job="kube-scheduler"}) by (cluster) > 0)	15m	Scheduler 从采集目标中消失
Controller Manager 故障	absent(sum(up{job="kube-controller-manager"}) by (cluster) > 0)	15m	Controller Manager 从采集目标中消失
Kubernetes 核心组件版本不一致	count by (cluster) (count by (cluster, gitVersion) (label_replace(kubernetes_build_info{job!~"kube-dns|coredns"},"gitVersion","$1","gitVersion","(v[0-9]*.[0-9]*).*"))) > 1	15m	Kubernetes 核心组件运行了不同的版本

Kubelet

策略名称	策略表达式	持续时间	策略描述
Node 状态异常	kube_node_status_condition{job=~".*kube-state-metrics",condition="Ready",status="true"} == 0	15m	Node 状态异常持续15m
Node 不可达	kube_node_spec_taint{job=~".*kube-state-metrics",key="node.kubernetes.io/unreachable",effect="NoSchedule"} == 1	15m	Node 不可达，上面的工作负载会重新调度
Node 上运行太多 pod	count by(cluster, node) ((kube_pod_status_phase{job=~".*kube-state-metrics",phase="Running"} == 1) * on(instance,pod,namespace,cluster) group_left(node) topk by(instance,pod,namespace,cluster) (1, kube_pod_info{job="prometheus-kube-state-metrics"})) /max by(cluster, node) (kube_node_status_capacity_pods{job=~".*kube-state-metrics"} != 1) > 0.95/td>	15m	Node 上运行 pod 量快达到上限

Node 状态抖动	sum(changes(kube_node_status_condition{status="true",condition="Ready"}[15m])) by (cluster, node) > 2	15m	Node 状态在正常和异常之间抖动
Kubelet 的客户端证书快过期	kubelet_certificate_manager_client_ttl_seconds < 86400	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	kubelet_certificate_manager_server_ttl_seconds < 86400	无	Kubelet 服务端证书将在24小时后过期
Kubelet 客户端证书续签出错	increase(kubelet_certificate_manager_client_expiration_renew_errors[5m]) > 0	15m	Kubelet 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_pleg_relist_duration_seconds_bucket[5m])) by (cluster, instance, le) * on(instance, cluster) 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, instance, le)) * on(cluster, 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) > 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) (max by (cluster, namespace, pod, container) (kube_pod_container_resource_requests_cpu_cores{job=~".*kube-state-metrics"}) * on(cluster, namespace, pod) group_left() max by (cluster, namespace, pod) (kube_pod_status_phase{phase=~"Pending|Running"} == 1)) /sum by (cluster) (kube_node_status_allocatable_cpu_cores) >(count by (cluster) (kube_node_status_allocatable_cpu_cores)-1) / count by (cluster) (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) (max by (cluster, namespace, pod, container) (kube_pod_container_resource_requests_memory_bytes{job=~".*kube-state-metrics"}) * on(cluster, namespace, pod) group_left() max by (cluster, namespace, pod) (kube_pod_status_phase{phase=~"Pending|Running"} == 1)) /sum by (cluster) (kube_node_status_allocatable_memory_bytes) >(count by (cluster) (kube_node_status_allocatable_memory_bytes)-1) /count by (cluster) (kube_node_status_allocatable_memory_bytes)	5m	集群内 Pod 申请的内存总量过多，已无法容忍 Node 挂掉
集群 CPU 配额过载	sum by (cluster) (kube_resourcequota{job=~".*kube-state-metrics", type="hard", resource="cpu"}) /sum by (cluster) (kube_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) (kube_resourcequota{job=~".*kube-state-metrics", type="hard", resource="memory"}) /sum by (cluster) (kube_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, namespace, resource) (kube_resourcequota{job=~".*kube-state-metrics", type="used"}) /sum by (cluster, namespace, resource) (kube_resourcequota{job=~".*kube-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, container, pod, namespace) /sum(increase(container_cpu_cfs_periods_total{}[5m])) by (cluster, 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"}[5m])) by (cluster, namespace, pod, container) / sum(kube_pod_container_resource_limits_cpu_cores) by (cluster, namespace, pod, container) > 0.75	15m	Pod 的 CPU 使用率超过75%
Pod 的内存使用率高	sum(container_memory_working_set_bytes{job="kubelet", metrics_path="/metrics/cadvisor", image!="", container!="POD"}) by (cluster, namespace, pod, container) /sum(kube_pod_container_resource_limits_memory_bytes) by (cluster, 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	策略描述
Pod 频繁重启	increase(kube_pod_container_status_restarts_total{job=~".*kube-state-metrics"}[5m]) > 0	15m	Pod 最近5m频繁重启
Pod 状态异常	sum by (namespace, pod, cluster) (max by(namespace, pod, cluster) (kube_pod_status_phase{job=~".*kube-state-metrics", phase=~"Pending|Unknown"}) * on(namespace, pod, cluster) group_left(owner_kind) topk by(namespace, pod) (1, max by(namespace, pod, owner_kind, cluster) (kube_pod_owner{owner_kind!="Job"}))) > 0	15m	Pod处于NotReady 状态超过15分钟
容器状态异常	sum by (namespace, pod, container, cluster) (kube_pod_container_status_waiting_reason{job=~".*kube-state-metrics"}) > 0	1h	容器长时间处于 Waiting 状态
Deployment 部署版本不匹配	kube_deployment_status_observed_generation{job=~".*kube-state-metrics"} !=kube_deployment_metadata_generation{job=~".*kube-state-metrics"}	15m	部署版本和设置版本不一致，表示 deployment 变更没有生效
Deployment 副本数不匹配	(kube_deployment_spec_replicas{job=~".*kube-state-metrics"} != kube_deployment_status_replicas_available{job=~".*kube-state-metrics"}) and (changes(kube_deployment_status_replicas_updated{job=~".*kube-state-metrics"}[5m]) == 0)	15m	实际副本数和设置副本数不一致
Statefulset 部署版本不匹配	(kube_statefulset_status_replicas_ready{job=~".*kube-state-metrics"} != kube_statefulset_status_replicas{job=~".*kube-state-metrics"}) and (changes(kube_statefulset_status_replicas_updated{job=~".*kube-state-metrics"}[5m]) == 0)	15m	部署版本和设置版本不一致，表示 statefulset 变更没有生效

Statefulset 副本数不匹配	kube_statefulset_status_observed_generation{job=~".*kube-state-metrics"} !=kube_statefulset_metadata_generation{job=~".*kube-state-metrics"}	15m	实际副本数和设置副本数不一致
Statefulset 更新未生效	(max without (revision) (kube_statefulset_status_current_revision{job=~".*kube-state-metrics"}unless kube_statefulset_status_update_revision{job=~".*kube-state-metrics"}) * (kube_statefulset_replicas{job=~".*kube-state-metrics"} != kube_statefulset_status_replicas_updated{job=~".*kube-state-metrics"})) and (changes(kube_statefulset_status_replicas_updated{job=~".*kube-state-metrics"}[5m]) == 0)	15m	Statefulset 部分 pod 没有更新
Daemonset 变更卡住	((kube_daemonset_status_current_number_scheduled{job=~".*kube-state-metrics"} != kube_daemonset_status_desired_number_scheduled{job=~".*kube-state-metrics"}) or (kube_daemonset_status_number_misscheduled{job=~".*kube-state-metrics"} != 0) or (kube_daemonset_updated_number_scheduled{job=~".*kube-state-metrics"} != kube_daemonset_status_desired_number_scheduled{job=~".*kube-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)	15m	Daemonset 变更超过15分钟
Daemonset 部分 node 未调度	kube_daemonset_status_desired_number_scheduled{job=~".*kube-state-metrics"} - kube_daemonset_status_current_number_scheduled{job=~".*kube-state-metrics"} > 0	10m	Daemonset 在部分 node 未被调度
Daemonset 部分 node 被错误调度	kube_daemonset_status_number_misscheduled{job=~".*kube-state-metrics"} > 0	15m	Daemonset 被错误调度到一些 node
Job 运行太久	kube_job_spec_completions{job=~".*kube-state-metrics"} - kube_job_status_succeeded{job=~".*kube-state-metrics"} > 0	12h	Job 执行时间超过12小时
Job 执行失败	kube_job_failed{job=~".*kube-state-metrics"} > 0	15m	Job 执行失败
副本数和 HPA 不匹配	(kube_hpa_status_desired_replicas{job=~".*kube-state-metrics"} !=kube_hpa_status_current_replicas{job=~".*kube-state-metrics"}) andchanges(kube_hpa_status_current_replicas[15m]) == 0	15m	实际副本数和 HPA 设置的不一致
副本数达到 HPA 最大值	kube_hpa_status_current_replicas{job=~".*kube-state-metrics"} ==kube_hpa_spec_max_replicas{job=~".*kube-state-metrics"}	15m	实际副本数达到 HPA 配置的最大值
PersistentVolume 状态异常	kube_persistentvolume_status_phase{phase=~"Failed|Pending",job=~".*kube-state-metrics"} > 0	15m	PersistentVolume 处于 Failed 或 Pending 状态

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	(node_filesystem_avail_bytes{job="node-exporter",fstype!=""} / node_filesystem_size_bytes{job="node-exporter",fstype!=""} * 100 < 3and node_filesystem_readonly{job="node-exporter",fstype!=""} == 0)	1h	文件系统空间预计在4小时后使用完
文件系统空间使用率高	(node_filesystem_avail_bytes{job="node-exporter",fstype!=""} / node_filesystem_size_bytes{job="node-exporter",fstype!=""} * 100 < 15and predict_linear(node_filesystem_avail_bytes{job="node-exporter",fstype!=""}[6h], 4*60*60) < 0and node_filesystem_readonly{job="node-exporter",fstype!=""} == 0)	1h	文件系统可用空间低于5%
文件系统 inode 快耗尽	(node_filesystem_files_free{job="node-exporter",fstype!=""} / node_filesystem_files{job="node-exporter",fstype!=""} * 100 < 3and node_filesystem_readonly{job="node-exporter",fstype!=""} == 0)	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	(node_filesystem_files_free{job="node-exporter",fstype!=""} / node_filesystem_files{job="node-exporter",fstype!=""} * 100 < 20and predict_linear(node_filesystem_files_free{job="node-exporter",fstype!=""}[6h], 4*60*60) < 0and node_filesystem_readonly{job="node-exporter",fstype!=""} == 0)	1h	文件系统可用 inode 低于3%
网卡状态不稳定	changes(node_network_up{job="node-exporter",device!~"veth.+"}[2m]) > 2	2m	网卡状态不稳定，在 up 和 down 间频繁变化
网卡接收出错	increase(node_network_receive_errs_total[2m]) > 10	1h	网卡接收数据出错
网卡发送出错	increase(node_network_transmit_errs_total[2m]) > 10	1h	网卡发送数据出错
机器时钟未同步	min_over_time(node_timex_sync_status[5m]) == 0	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	(node_timex_offset_seconds > 0.05and deriv(node_timex_offset_seconds[5m]) >= 0)or(node_timex_offset_seconds < -0.05and deriv(node_timex_offset_seconds[5m]) <= 0)	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：2024-11-07 10:34:32

本文指导您在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、微信。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 MONITOR 读权限。
- 创建、编辑通知模板：子账号需拥有 MONITOR 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，选择**告警管理 > 告警配置 > 通知模板**。
2. 单击**新建通知模板**，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：可选用户/用户组/值班表，如需创建告警接收组请参见 创建接收组 。
  - 通知周期：周一 – 周日，可自行勾选。
  - 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、企业微信五种告警渠道。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：
    - 拨打类型：
      - 同时拨打：支持接收对象下的所有用户同时接收电话告警。
      - 轮询拨打：对接收对象下的用户轮询进行电话告警。
        - 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。
    - 电话确认：开启后用户需要在接通电话后按1确认已接通，否则当前电话告警电话会再次触达。
- 接口回调：填写公网可访问到的url作为回调接口地址，腾讯云可观测平台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见 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。
- 告警回调 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。

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。

查看告警指标图表

最近更新时间：2024-12-05 16:07:33

本文将为您介绍在触发 Prometheus 告警后，如何快速查看告警指标图表。

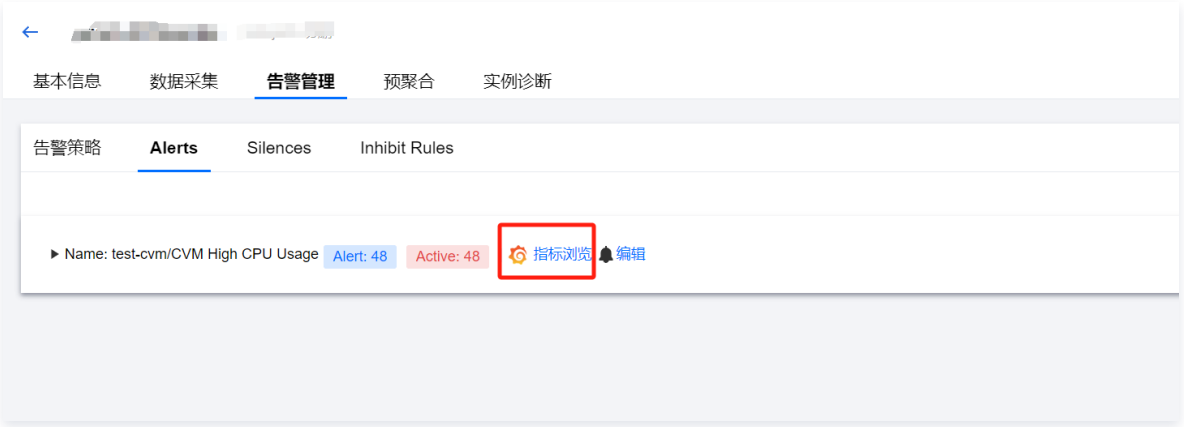
前提条件：

当前 Prometheus 实例已绑定 Grafana 服务，绑定指引可参考 Grafana 服务 。

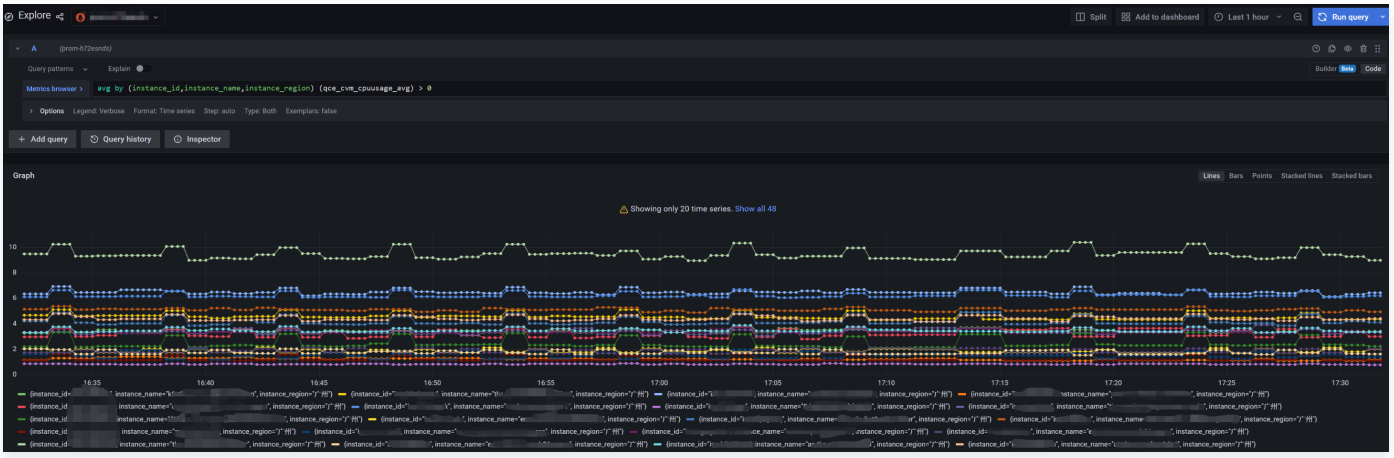
操作步骤

方法一：通过 Prometheus 控制台查看

1. 登录 Prometheus 监控服务控制台 。
2. 在 Prometheus 实例列表中，单击**实例 ID/名称** 。
3. 进入 Prometheus 管理中心，在顶部导航栏中单击**告警管理 > Alerts** ，再单击 Alerts 页面中的**指标浏览**。



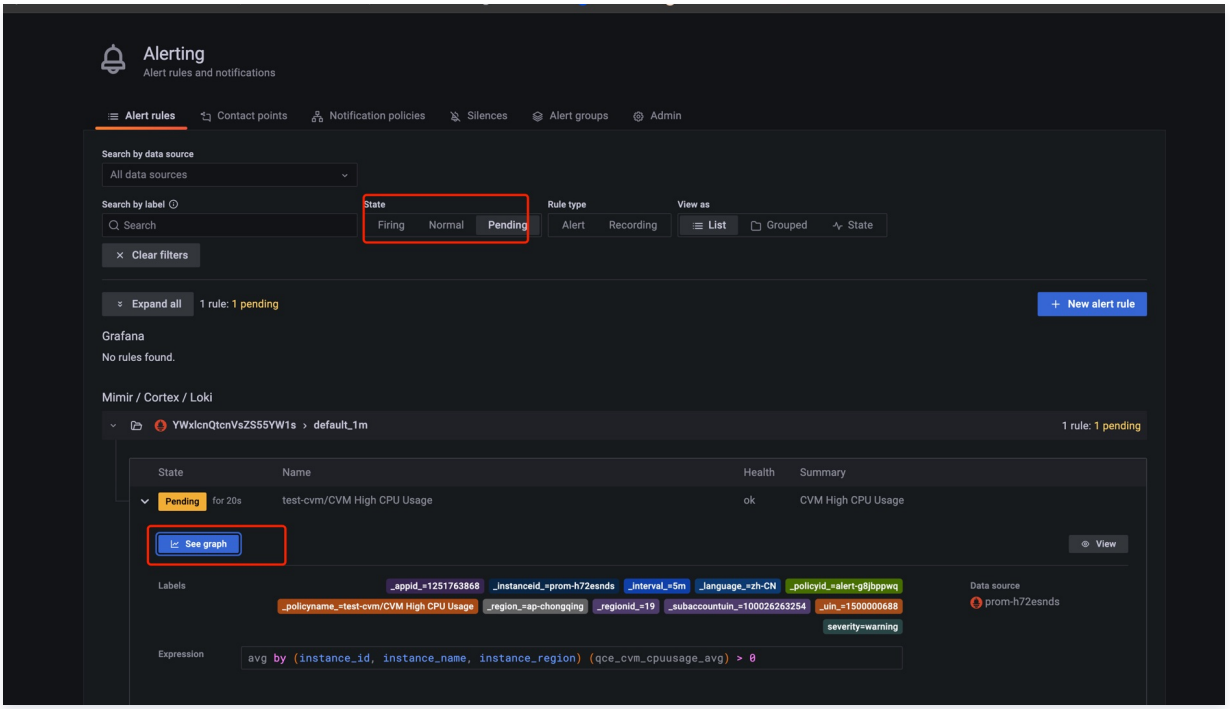
4. 单击**指标浏览**后会跳转到 Grafana 平台页面，展示与 Prometheus 告警相关的指标数据图表，如下图所示：

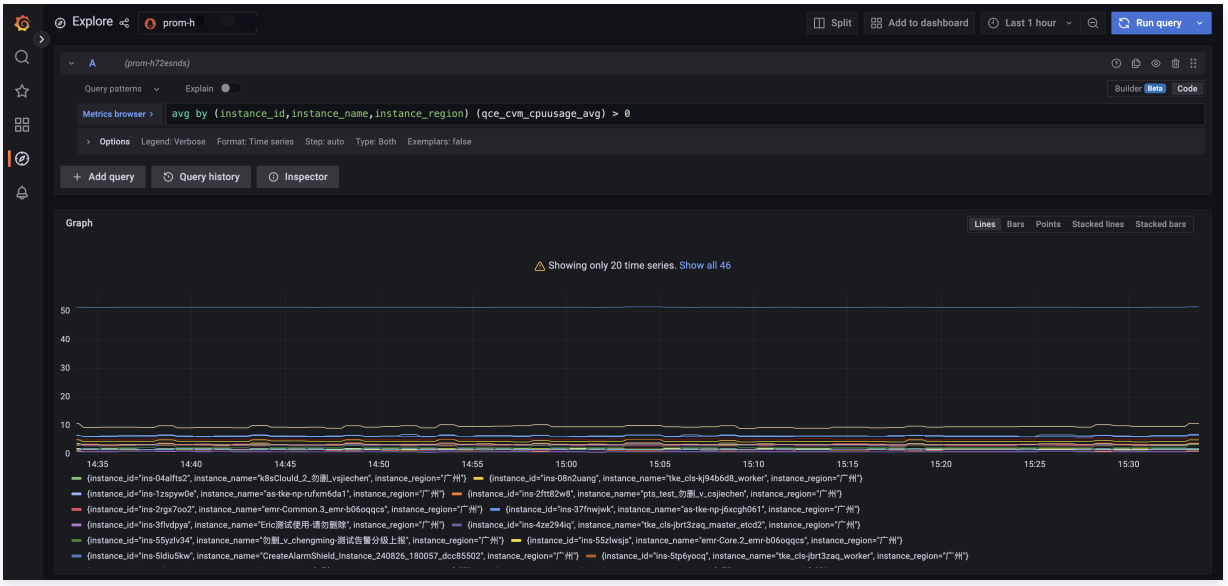


方法二：通过 Grafana 平台查看

1. 登录 Grafana 服务控制台 。
2. 单击需要查看实例右侧的 

3. 输入账号密码登录之后点击左侧导航栏的  进入Alerting 页面，再点击 See graph 即可查看与 Prometheus 告警相关的指标数据图表。


Alerting
Alert rules and notifications
Alert rules
Contact points
Notification policies
Silences
Alert groups
Admin
Search by data source
All data sources
Search by label
Search
State
Firing
Normal
Pending
Rule type
Alert
Recording
View as
List
Grouped
State
Clear filters
Expand all
1 rule: 1 pending
New alert rule
Grafana
No rules found.
Mimir / Cortex / Loki
YWxlcnQtcnVsZS55YW1s › default_1m
1 rule: 1 pending
State
Name
Health
Summary
Pending
for 20s
test-cvm/CVM High CPU Usage
ok
CVM High CPU Usage
See graph
View
Labels
appid=1251763868
instanceid=prom-h72esnds
interval=5m
language=zh-CN
policyid=alert-g8jbppwq
policyname=test-cvm/CVM High CPU Usage
region=ap-chongqing
regionid=19
subaccountuin=100026263254
uin=1500000688
severity=warning
Data source
prom-h72esnds
Expression
avg by (instance_id, instance_name, instance_region) (qce_cvm_cpuusage_avg) > 0


Explore
prom-h
Split
Add to dashboard
Last 1 hour
Run query
A
(prom-h72esnds)
Query patterns
Explain
Builder
Beta
Code
Metrics browser
avg by (instance_id,instance_name,instance_region) (qce_cvm_cpuusage_avg) > 0
Options
Legend: Verbose
Format: Time series
Step: auto
Type: Both
Exemplars: false
Add query
Query history
Inspector
Graph
Lines
Bars
Points
Stacked lines
Stacked bars
Showing only 20 time series. Show all 46
50
40
30
20
10
0
14:35
14:40
14:45
14:50
14:55
15:00
15:05
15:10
15:15
15:20
15:25
15:30

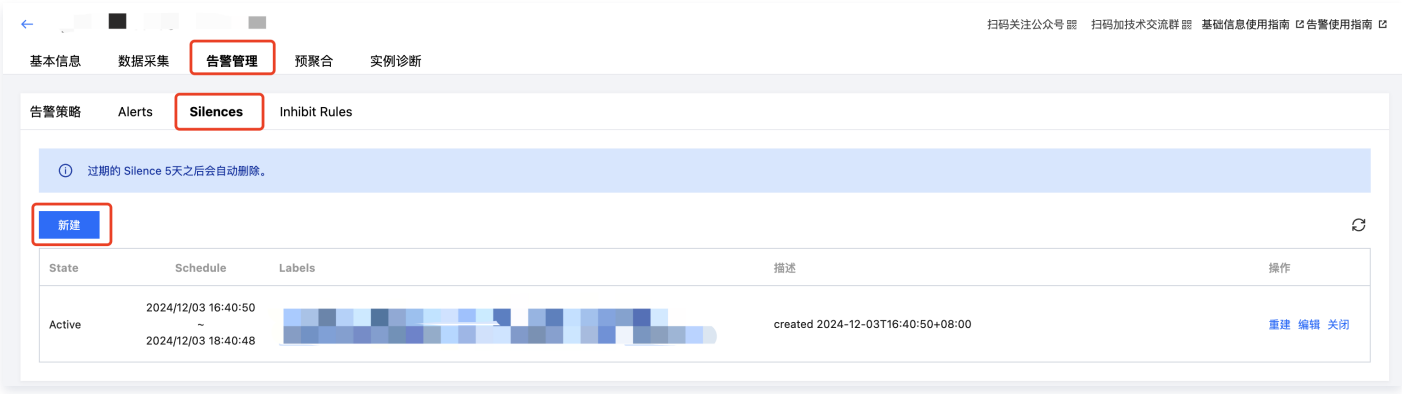
告警静默

最近更新时间：2024-12-18 10:15:23

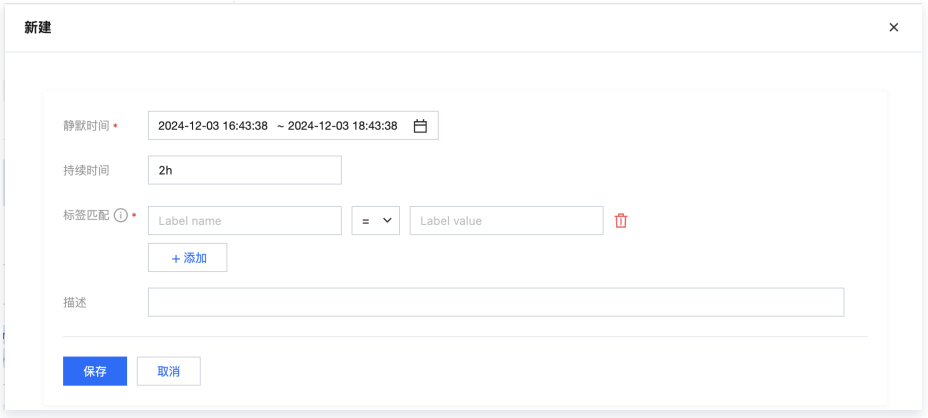
告警静默一般用于处理已知问题或在系统维护期间，暂时关闭特定警报的通知，或者对于频繁触发但不需要关注的警报，使用静默来减少干扰。告警静默需要设置静默策略，对满足匹配规则的告警进行静音处理，不发送告警通知。

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**实例 ID/名称** 。
3. 进入 Prometheus 管理中心，在顶部导航栏中选择**告警管理 > Silences**，单击**新建**。



4. 跳转到新建页面后，根据页面提示配置抑制规则，配置完单击**保存**即可。



参数说明

参数	说明
静默时间	必填项，静默规则生效的起止时间。
持续时间	静默持续的时长，即静默时间的截止时间减去开始时间。
标签匹配	必填项，满足标签匹配规则的告警将被静默，输入标签名称、条件、标签值。可添加多个规则，满足所有匹配规则的告警信息才会被静默。
描述	对于静默规则的说明。

注意：
在静默期间，符合匹配规则的告警消息和告警恢复消息都会被静默。

示例

使用场景：系统维护期间静默相关告警

场景描述

某个业务系统，设置了高负载时的告警，例如当 CPU 使用率超过某个阈值。为了进行定期的系统维护，需要在某个时间段内进行服务器升级，期间虽然会出现一些高负载，但已知这些不会影响用户体验，因此希望静音相关的告警。
假设需要静音的告警：

- 策略名称：HighCPUUsage。
- 相关实例：“server-”开头的实例。

则可以按照如下方式配置静默规则：

- 静默时间：2024-12-03 20:00:00~2024-12-03 22:00:00
- 持续时间：2h
- 标签匹配：
  - alertname=HighCPUUsage
  - instance=~server-.*
- 描述：系统维护，静默 CPU 高负载的告警。

整体效果

在2024-12-03 20:00:00 到 2024-12-03 22:00:00 的时间段内，所有匹配标签规则的告警（即“server-”开头实例的 CPU 高负载告警）将不会触发通知，从而避免了在维护期间的告警干扰。

告警抑制

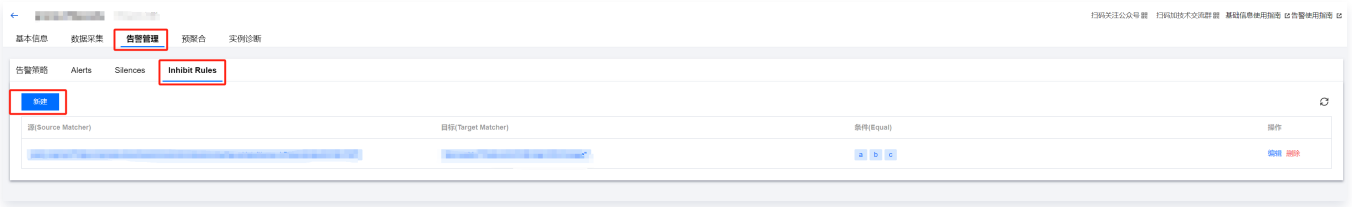
最近更新时间：2024-12-05 16:07:33

前言

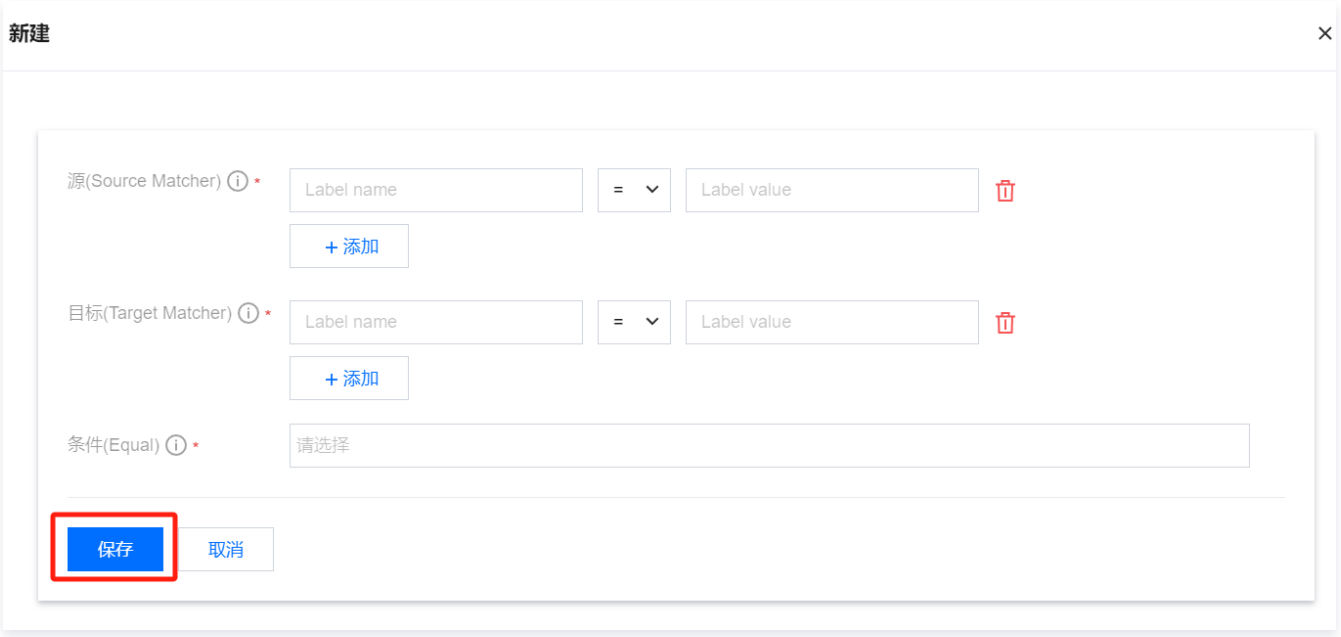
为了避免由于相同问题导致的成百上千的相似告警通知带来额外的运维工作量，我们增加了告警抑制功能。告警抑制指的是若某种类型的告警被触发，则抑制与之相关的其他相似告警。例如：如果告警内容是某个集群无法访问，则可以配置 Inhibition 规则，静默与该集群相关的所有其他告警。

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**实例 ID/名称** 。
3. 进入 Prometheus 管理中心，在顶部导航栏中单击**告警管理 > Inhibit Rules > 新建**。



4. 跳转到新建页面后，根据页面提示配置抑制规则，配置完点击**保存**即可。



参数说明

参数	说明
源(Source Matcher)	触发的告警，选择标签名称、条件、标签值。
目标(Target Matcher)	需要被静默的告警，选择标签名称、条件、标签值。
条件(Equal)	目标和源告警对于匹配条件中的标签名称必须具有相同的标签值，选择标签名称。

说明：

- **Inhibition 规则设置**：当存在满足某种规则的告警（源）时，抑制规则会静默满足另一种规则的告警（目标）。目标和源告警对于匹配条件中的标签名称必须具有相同的标签值。

- 为了防止警报自我抑制，与规则的目标端和源端都匹配的告警不能被与目标端和源端都匹配的其他告警（包括其自身）抑制。因此，建议告警的源和目标规则设计上要确保不会有任何告警同时匹配源规则和目标规则。

示例

使用场景：服务器 CPU 高负载告警

场景描述：

在一个监控系统中，配置了两个告警：

- 告警 A：CPU 负载超过90%。
- 告警 B：系统响应时间超过500ms。

这两个告警都是由于同一原因引起的，即服务器 CPU 高负载，导致系统性能下降。

告警 A 的策略规则如下：

```
alert: HighCPUUsage
expr: avg(rate(cpu_usage_seconds_total[5m])) by (instance) > 0.9
```

告警 B 的策略规则如下：

```
alert: HighResponseTime
expr: avg(response_time_seconds) by (instance) > 0.5
```

则 Inhibition 规则配置方式如下：

- 源：alert=HighCPUUsage
- 目标：alert=HighResponseTime
- 匹配条件：instance

整体效果：

- cpu_usage_seconds_total 指标在5分钟内的平均速率为95%，该指标的标签 instance=instanceX，则会触发告警 A，发送告警通知；
- response_time_seconds 指标的平均值为0.8s，该指标的标签 instance=instanceX，则会触发告警 B，但由于匹配上了 Inhibition 规则，所以不会发送告警通知。

标签管理
标签示例

最近更新时间：2024-12-05 16:07:34

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、李四、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、李四、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	李四
prom-1jqwv16	游戏	游戏 B	李四
prom-1jqwv17	游戏	游戏 B	李四
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 。
- 筛选出游戏部门中李四负责的 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“李四”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：2024-12-05 16:07:34

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**，对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：

编辑标签

最近更新时间：2024-12-05 16:07:34

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. 在弹出的窗口中，根据实际需求进行添加、修改或者删除标签：

使用限制

最近更新时间：2024-12-05 16:07:34

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127个字符。

标签值限制

- 只支持字母、数字、空格、汉字或特殊符号。
- 标签值最大长度为255个字符。

访问控制
访问控制概述

最近更新时间：2024-12-05 16:07:34

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。
- 您无法限制其他人的访问权限，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：2024-12-05 16:07:34

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见 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

腾讯云可观测平台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*"
]
```

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 `uin/${OwnerUin}`，如 uin/100000000001	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-73jingds"
            ]
        }
    ]
}
```

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-65jidfafk) 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-73jingds",
          "qcs::monitor:ap-guangzhou:uin/1250000000:prom-instance/prom-65jidfafk"
      ],
      "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：2024-12-05 16:07:34

Prometheus 自定义策略

如果预设策略无法满足需求，可进入 访问管理控制台 > 策略， 单击**新建自定义策略**创建自定义策略。

创建自定义策略的方法可参见 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。可以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	`qcs::monitor:$region:$account:prom-instance/*` `qcs::monitor:$region:$account:prom-instance/$instanceId`

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
CreateAlarmNotice	创建通知模板
DeleteAlarmNotices	删除告警通知模板（批量）
ModifyAlarmNotice	修改通知模板
DescribeAlarmNotices	查询通知模板列表
DescribeAlarmNotice	查询单个通知模板的详情
DescribeAlarmNoticeCallbacks	查询账号下所有回调URL列表

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：2025-04-24 19:00:43

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 TKE_QCSLinkedRoleInPrometheusService、TKE_QCSRole 和 CM_QCSLinkedRoleInTMP 三个服务角色。

- TKE_QCSLinkedRoleInPrometheusService 角色是用于授权 Prometheus 访问容器服务，默认关联的预设策略如下：

  QcloudAccessForTKELinkedRoleInPrometheusService：该策略仅用于腾讯云容器服务（TKE）访问其他云服务资源。包含对象存储（COS）相关操作权限。
- TKE_QCSRole 角色是用于授权容器服务，默认关联的预设策略如下：
  - QcloudAccessForTKERole：该策略用于腾讯云容器服务（TKE）访问云资源。
  - QcloudAccessForTKERoleInOpsManagement：该策略用于腾讯云容器服务（TKE）访问其他云服务资源。包含日志服务（CLS）相关操作权限。
- CM_QCSLinkedRoleInTMP 角色是用于授权 Prometheus 获取云资源信息，默认关联的预设策略如下：

  QcloudAccessForCMLinkedRoleInTMP：该策略用于 monitor 访问其他云服务资源。

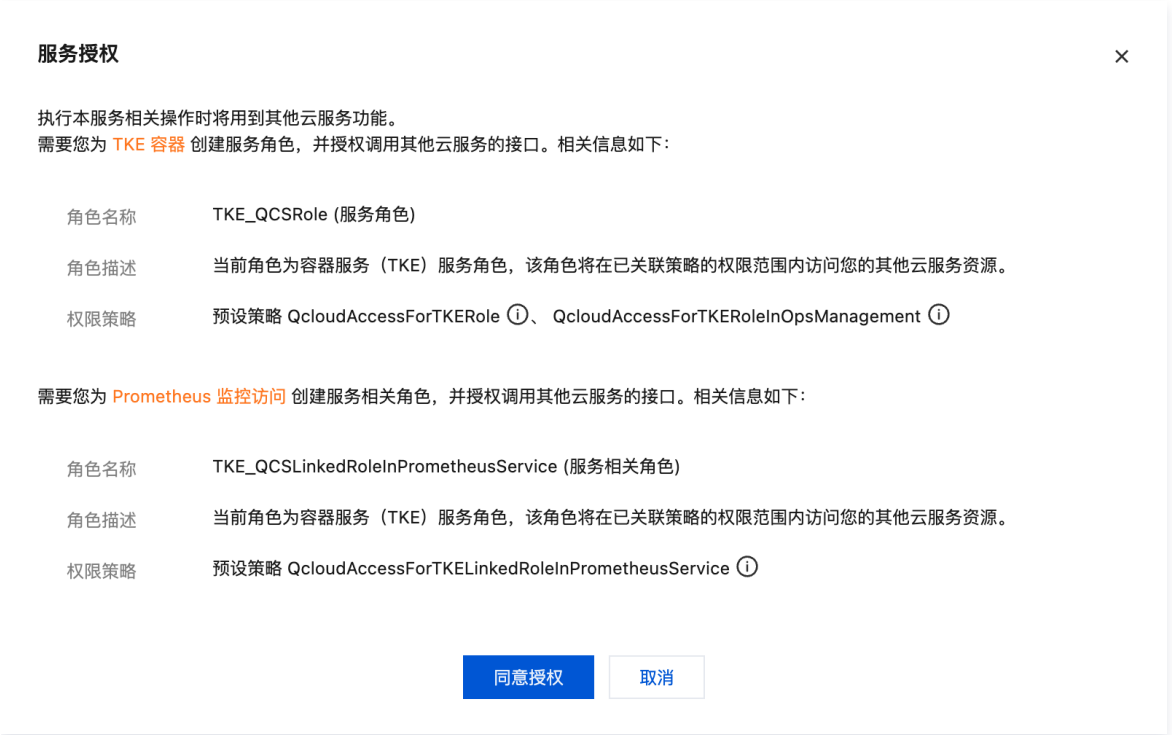
操作场景

为了采集腾讯云容器服务（TKE）或集成中心其他组件监控的数据，Prometheus 服务实例需要访问相关 API 服务，需要您的授权委托，才能正常访问。此角色无需主动寻找配置，在未授权的情况下，使用相关功能时会自动弹出授权界面。

授权步骤

主账号授权

进入 Prometheus 实例购买页，填写完基本信息单击**立即购买**后将会出现授权提示框，进行 TKE_QCSLinkedRoleInPrometheusService 和 TKE_QCSRole 两个角色的授权，如下所示。单击**同意授权**即可授权成功。



若您需要使用集成中心的**云监控**、**CVM Node Exporter**、**CVM 进程监控**、**CVM 云服务器**、**EMR**、**Cdwch** 的一键安装功能，则需进行 CM_QCSLinkedRoleInTMP 角色的授权，如下图所示：

单击提示框的点击授权，即会弹出如下窗口，单击同意授权后即可授权成功。

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

主账号完成上述授权操作，成功创建了角色后，子账号无权限访问角色，需主账号对子账号授予 PassRole 权限，子账号才能正常使用，否则会提示失败。
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 QcloudCamFullAccess
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	详情请参见 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在 访问管理 > 策略 > 新建自定义策略里边按策略语法创建，语法示例如下：

```
{
 "version": "2.0",
 "statement": [
     {
         "effect": "allow",
         "action": "cam:PassRole",
         "resource":
"qcs::cam::uin/${OwnerUin}:role/tencentcloudServiceRoleName/TKE_QCSLinkedRoleInPrometheusService"
     },
     {
         "effect": "allow",
         "action": "cam:PassRole",
```

```
            "resource": "qcs::cam::uin/${OwnerUin}:roleName/TKE_QCSRole"
        }
    ]
}
```

为控制权限，示例语法限制了 cam:PassRole 仅对 TKE_QCSLinkedRoleInPrometheusService 和 TKE_QCSRole 服务角色生效，您可根据需要增减生效的角色。

2. 新建完后，参见 授权管理 在自定义策略下关联子账号即可。

Grafana 服务

最近更新时间：2024-12-05 16:07:34

Prometheus 监控服务与 Grafana 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

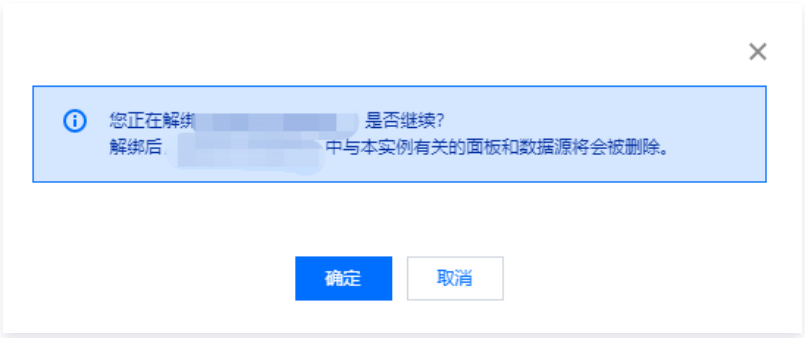
1. 登录 Prometheus 监控服务控制台。
2. 在 Prometheus 列表中找对应的 Prometheus 实例，在操作列单击**更多 > Grafana > 绑定 Grafana**，或在 Grafana 访问地址列点击**绑定 Grafana** 。
3. 在弹框中选择 Grafana 实例，并单击**确定**即可。

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 ，请参见 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 列表中找对应的 Prometheus 实例，在操作列单击**更多 > Grafana > 解绑 Grafana**。

3. 在弹框中单击**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 列表中找对应的 Prometheus 实例，单击 Grafana 访问地址列的**登录 Grafana** 功能。

3. 进入 Grafana 界面输入账号和密码即可登录，也可以选择 SSO 登录。

> **说明：**
> 更多 Grafana 操作请参见 Grafana 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：2024-12-05 16:07:34

支持的 API

Prometheus 监控服务所有支持的 API 与开源 Prometheus 提供的 API 有相同的参数输入和响应数据格式，未在此列出的 API 默认都不支持，或者部分 API 不是所有的版本都支持，以当前文档为准，所有 API 均为 HTTP 协议：

API	说明	使用方式
/api/v1/query	查询某一时刻的数据	推荐使用 Grafana，可使用 HTTP 相关工具，部分 SDK 提供实现
/api/v1/query_range	查询时间范围类的数据	推荐使用 Grafana，可使用 HTTP 相关工具，部分 SDK 提供实现
/api/v1/series	查询 series	推荐使用 Grafana，可使用 HTTP 相关工具，部分 SDK 提供实现
/api/v1/labels	查询标签名	推荐使用 Grafana，可使用 HTTP 相关工具，部分 SDK 提供实现
/api/v1/label/{label_name}/values	查询标签名多对应的值	推荐使用 Grafana，可使用 HTTP 相关工具，部分 SDK 提供实现
/api/v1/prom/write	remote write 上报数据	常用方式为使用相关 Agent，例如 Prometheus
/metrics/{job}/{label-pairs*}	Pushgateway 上报数据	开源 SDK

说明：
Prometheus 监控服务不直接提供 SDK 实现，需使用开源 Prometheus 实现，各语言的实现请参见：开源指引 。
Pushgateway 协议在这里仅作为 remote write 协议的补充，不支持删除操作，可以理解为我们提供的删除接口是空的不执行任何逻辑，特殊需求可自行部署 PushGateway 。

认证方法

以上提供的所有的 API 都需要认证，且所有的认证方式都支持下面两种。

Basic Auth（推荐）

Basic Auth 兼容原生 Prometheus Query 的认证方式，用户名为用户的 APPID，进入控制台基本信息里面的 Token 即为密码。
了解更多：https://swagger.io/docs/specification/authentication/basic-authentication/

Bearer Token

Bearer Token 随着实例创建而生成，进入控制台查看基本信息里面的 Token。
了解更多：https://swagger.io/docs/specification/authentication/bearer-authentication/

API 响应及相关状态码

数据上报请求无固定的响应格式，程序只需关注状态码即可，错误响应最好记录详细的响应信息。
查询请求的响应格式为 JSON，基本结构如下：

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

相关状态码注解，错误相关的详细信息会包含在 HTTP 响应体内：

状态码	请求类型	概述
400	查询 / 数据上报	请求参数错误 / 数据上报时如果 series 达到上限可能会出现此错误
401	查询 / 数据上报	认证失败
404	查询 / 数据上报	API 不存在
422	查询	查询请求的表达式无法执行(RFC4918)
429	数据上报	数据上报时 samples 速率达到上限（对于基础版，如果自监控上体现出余量还较多，Agent 会自动重新上报数据，这种情况理论上不会丢失数据，平均下来不超过限制）
500	查询 / 数据上报	内部错误，频繁出现请联系我们
503	查询 / 数据上报	服务在启动、重建或升级中 / 查询请求被中止或者超时

数据写入

最近更新时间：2024-12-10 14:55:32

操作场景

在一些场景下（例如：Flink job 的生命周期可能很短，来不及等待 Prometheus 拉取其指标），我们需要将指标数据直接写入 Prometheus。此时可以使用 Remote Write 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 访问这里。使用 remote write 我们可以把 VPC 内其他 Prometheus 的数据写入到腾讯云托管的 Prometheus 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 采集指标建议以 Pull 模式拉取为主，但是有些场景我们仍然需要使用 Push 推送的方式，详情请参见 官方文档。详细使用参见 Pushgateway 接入。

监控数据查询

最近更新时间：2024-12-05 16:07:34

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

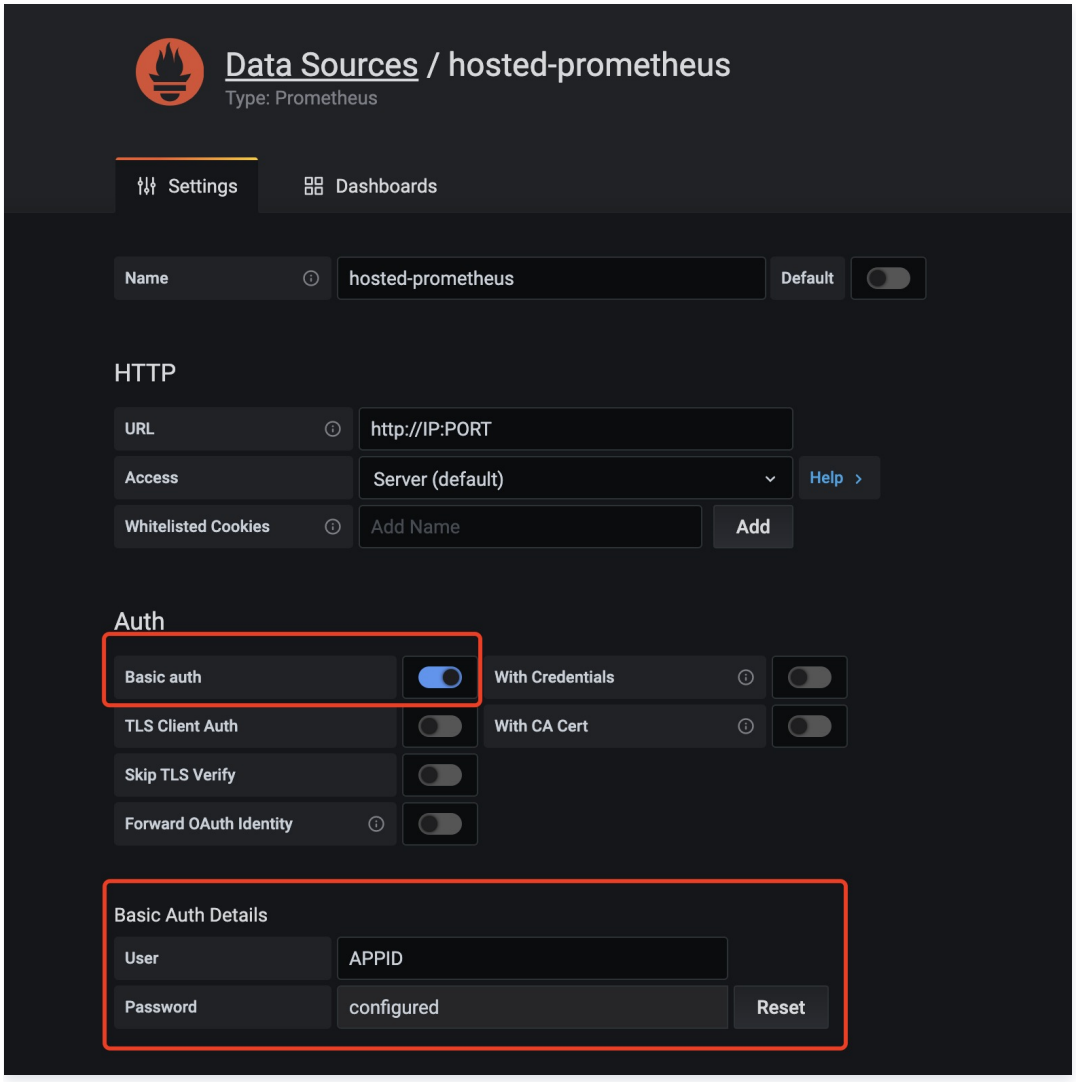
- query=<string>: Prometheus：查询表达式。
- time=<rfc3339 | unix_timestamp>：时间戳， 可选。
- timeout=<duration>：检测超时时间，可选。默认由 `-query.timeout` 参数指定。

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-
01T20:10:30.781Z&end=2015-07-01T20:11:00.781Z&step=15s'
{
   "status" : "success",
   "data" : {
      "resultType" : "matrix",
      "result" : [
         {
            "metric" : {
               "__name__" : "up",
               "job" : "prometheus",
               "instance" : "localhost:9090"
            },
            "values" : [
               [ 1435781430.781, "1" ],
               [ 1435781445.781, "1" ],
               [ 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。


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：2024-12-05 16:07:34

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
node-exporter	process_cpu_seconds_total
node-exporter	process_resident_memory_bytes
node-exporter	up
node-exporter	scrape_duration_seconds
node-exporter	scrape_timeout_seconds
node-exporter	scrape_series_added
node-exporter	scrape_samples_scraped
node-exporter	scrape_samples_post_metric_relabeling
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
cadvisor	up
cadvisor	scrape_duration_seconds

cadvisor	scrape_timeout_seconds
cadvisor	scrape_series_added
cadvisor	scrape_samples_scraped
cadvisor	scrape_samples_post_metric_relabeling
eks-network	container_cpu_usage_seconds_total
eks-network	container_memory_working_set_bytes
eks-network	container_fs_limit_bytes
eks-network	container_fs_reads_bytes_total
eks-network	container_fs_usage_bytes
eks-network	container_fs_writes_bytes_total
eks-network	container_network_receive_bytes_total
eks-network	container_network_receive_packets_dropped_total
eks-network	container_network_receive_packets_total
eks-network	container_network_transmit_bytes_total
eks-network	container_network_transmit_packets_dropped_total
eks-network	container_network_transmit_packets_total
eks-network	up
eks-network	scrape_duration_seconds
eks-network	scrape_timeout_seconds
eks-network	scrape_series_added
eks-network	scrape_samples_scraped
eks-network	scrape_samples_post_metric_relabeling
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
kubelet	workqueue_adds_total
kubelet	workqueue_depth
kubelet	workqueue_queue_duration_seconds_bucket
kubelet	up
kubelet	scrape_duration_seconds
kubelet	scrape_timeout_seconds
kubelet	scrape_series_added
kubelet	scrape_samples_scraped
kubelet	scrape_samples_post_metric_relabeling
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-state-metrics	kube_pod_container_resource_requests
kube-state-metrics	kube_pod_container_resource_limits
kube-state-metrics	kube_node_status_capacity
kube-state-metrics	kube_node_status_allocatable
kube-state-metrics	up
kube-state-metrics	scrape_duration_seconds
kube-state-metrics	scrape_timeout_seconds
kube-state-metrics	scrape_series_added
kube-state-metrics	scrape_samples_scraped
kube-state-metrics	scrape_samples_post_metric_relabeling
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-controller-manager	process_cpu_seconds_total
kube-controller-manager	process_resident_memory_bytes
kube-controller-manager	storage_operation_duration_seconds_bucket
kube-controller-manager	storage_operation_duration_seconds_count
kube-controller-manager	storage_operation_errors_total
kube-controller-manager	up
kube-controller-manager	scrape_duration_seconds
kube-controller-manager	scrape_timeout_seconds
kube-controller-manager	scrape_series_added
kube-controller-manager	scrape_samples_scraped
kube-controller-manager	scrape_samples_post_metric_relabeling
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
kube-scheduler	process_cpu_seconds_total
kube-scheduler	process_resident_memory_bytes
kube-scheduler	rest_client_request_duration_seconds_bucket
kube-scheduler	rest_client_requests_total
kube-scheduler	workqueue_adds_total
kube-scheduler	workqueue_depth
kube-scheduler	workqueue_queue_duration_seconds_bucket
kube-scheduler	up
kube-scheduler	scrape_duration_seconds
kube-scheduler	scrape_timeout_seconds
kube-scheduler	scrape_series_added
kube-scheduler	scrape_samples_scraped
kube-scheduler	scrape_samples_post_metric_relabeling
kube-proxy	rest_client_requests_total
kube-proxy	rest_client_request_duration_seconds_bucket

kube-proxy	process_resident_memory_bytes
kube-proxy	process_cpu_seconds_total
kube-proxy	up
kube-proxy	scrape_duration_seconds
kube-proxy	scrape_timeout_seconds
kube-proxy	scrape_series_added
kube-proxy	scrape_samples_scraped
kube-proxy	scrape_samples_post_metric_relabeling

容器常用指标推荐

最近更新时间：2024-12-05 16:07:34

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 a 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 a 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-state-metrics	kube_deployment_status_replicas_updated	The number of updated replicas 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 containers 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 moved 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-state-metrics	kube_pod_created	Unix creation timestamp.
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 a 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 daemon 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 a specific generation of the desired state for the StatefulSet.
kube-state-metrics	kube_secret_created	Unix creation timestamp.
kube-state-metrics	kube_endpoint_address_not_ready	Number of addresses not ready in endpoint.
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 in 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 a deployment.
kube-state-metrics	kube_endpoint_created	Unix creation timestamp.
kube-state-metrics	kube_replicaset_labels	Kubernetes labels converted to Prometheus labels.
kube-state-metrics	kube_replicaset_metadata_generation	Sequence number representing a 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-state-metrics	kube_secret_metadata_resource_version	Resource version representing a 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-state-metrics	kube_node_status_capacity	The capacity for different 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-	kube_node_spec_unschedulable	Whether a node can schedule new pods.

metrics		
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 a 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-metrics	kube_mutatingwebhookconfiguration_metadata_resource_version	Resource version representing a specific version of the MutatingWebhookConfiguration.
kube-state-metrics	kube_daemonset_updated_number_scheduled	The total number of nodes that are running updated daemon pod.
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_conntrack_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_conntrack_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 state 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 guests (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 or 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 state memory.
node-exporter	node_memory_HugePages_Surp	Memory information field HugePages_Surp.
node-exporter	node_timex_loop_time_constant	Phase-locked loop time constant.
node-exporter	node_timex_pps_stability_hertz	Pulse per second stability.

容器监控图表指标

最近更新时间：2025-05-30 14:14:31

集群监控概览

图表名称	查询语句	使用的指标	配置文件
CPU Requests Commitment	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster"}) / sum(kube_node_status_allocatable_cpu_cores{cluster="$cluster"})	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		kube_node_status_allocatable_cpu_cores	kube-state-metrics
CPU Limits Commitment	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster"}) / sum(kube_node_status_allocatable_cpu_cores{cluster="$cluster"})	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		kube_node_status_allocatable_cpu_cores	kube-state-metrics
Memory Utilisation	1 - sum(:node_memory_MemAvailable_bytes:sum{cluster="$cluster"}) / sum(node_memory_MemTotal_bytes{cluster="$cluster"})	node_memory_MemAvailable_bytes	node-exporter
		node_memory_MemTotal_bytes	node-exporter
Memory Requests Commitment	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster"}) / sum(kube_node_status_allocatable_memory_bytes{cluster="$cluster"})	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		kube_node_status_allocatable_memory_bytes	kube-state-metrics
Memory Limits Commitment	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster"}) / sum(kube_node_status_allocatable_memory_bytes{cluster="$cluster"})	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
		kube_node_status_allocatable_memory_bytes	kube-state-metrics
Node Count	count(kube_node_info{cluster="$cluster"})	kube_node_info	kube-state-metrics
Pod Count	count(kube_pod_info{cluster="$cluster"})	kube_pod_info	kube-state-metrics
Node Request CPU Average Percent	avg(sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster"})by(node)/sum(kube_node_status_capacity_cpu_cores{cluster="$cluster"})by(node))	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		kube_node_status_capacity_cpu_cores	kube-state-metrics
Node Request Memory Average Percent	avg(sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster"})by(node)/sum(kube_node_status_capacity_memory_bytes{cluster="$cluster"})by(node))	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		kube_node_status_capacity_memory_bytes	kube-state-metrics
API Server Success Request Percent	sum(irate(apiserver_request_total{cluster="$cluster",code=~"20.*",verb=~"GET\|LIST"}[5m]))/sum(irate(apiserver_request_total{cluster="$cluster",verb=~"GET|LIST"}[5m]))	apiserver_request_total	kube-apiserver
		apiserver_request_total	kube-apiserver

Namespace Overview	count(kube_pod_info{cluster="$cluster"}) by (namespace)	kube_pod_info	kube-state-metrics
	count(kube_service_info{cluster="$cluster"}) by(namespace)	kube_service_info	kube-state-metrics
	count(kube_pod_container_info{cluster="$cluster"}) by(namespace)	kube_pod_container_info	kube-state-metrics
	count(kube_configmap_info{cluster="$cluster"}) by(namespace)	kube_configmap_info	kube-state-metrics
	count(kube_secret_info{cluster="$cluster"}) by(namespace)	kube_secret_info	kube-state-metrics
	count(kube_deployment_created{cluster="$cluster"}) by (namespace)	kube_deployment_created	kube-state-metrics
	count(kube_statefulset_created{cluster="$cluster"}) by (namespace)	kube_statefulset_created	kube-state-metrics
	count(kube_job_created{cluster="$cluster"}) by (namespace)	kube_job_created	kube-state-metrics
	count(kube_cronjob_created{cluster="$cluster"}) by (namespace)	kube_cronjob_created	kube-state-metrics
	count(kube_pod_status_ready{cluster="$cluster",condition="false"}==1) by(namespace) - (count(kube_pod_status_phase{cluster="$cluster",phase="Succeeded"}==1) by(namespace) or vector(0)) or count(kube_pod_status_ready{cluster="$cluster",condition="false"}==1) by(namespace)	kube_pod_status_ready	kube-state-metrics
		kube_pod_status_phase	kube-state-metrics
		kube_pod_status_ready	kube-state-metrics
	count(kube_deployment_status_replicas_ready{cluster="$cluster"} <kube_deployment_spec_replicas{cluster="$cluster"}) by (namespace)	kube_deployment_status_replicas_ready	kube-state-metrics
		kube_deployment_spec_replicas	kube-state-metrics
	count(kube_statefulset_status_replicas_ready{cluster="$cluster"} <kube_statefulset_replicas{cluster="$cluster"}) by (namespace)	kube_statefulset_status_replicas_ready	kube-state-metrics
		kube_statefulset_replicas	kube-state-metrics
	count(kube_daemonset_status_number_unavailable{cluster="$cluster"}>0)by(namespace)	kube_daemonset_status_number_unavailable	kube-state-metrics
	count(kube_job_status_failed{cluster="$cluster"} == 1) by (namespace)	kube_job_status_failed	kube-state-metrics
	count(kube_daemonset_created{cluster="$cluster"}) by (namespace)	kube_daemonset_created	kube-state-metrics
	count(kube_persistentvolumeclaim_info{cluster="$cluster"}) by (namespace)	kube_persistentvolumeclaim_info	kube-state-metrics
CPU Usage	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_r	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标

	ate{cluster="$cluster", container!="POD", container!=""}) by (namespace)		
CPU Quota	sum(kube_pod_owner{cluster="$cluster"}) by (namespace)	kube_pod_owner	kube-state-metrics
	count(avg(namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster"}) by (workload, namespace)) by (namespace)	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", container!="POD", container!=""}) by (namespace)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster"}) by (namespace)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", container!="POD", container!=""}) by (namespace) / sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster"}) by (namespace)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster"}) by (namespace)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", container!="POD", container!=""}) by (namespace) / sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster"}) by (namespace)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
Memory Usage (working_set)	sum(container_memory_working_set_bytes{cluster="$cluster", container!="", container!="POD"}) by (namespace)	container_memory_working_set_bytes	cadvisor
Memory Requests	sum(kube_pod_owner{cluster="$cluster"}) by (namespace)	kube_pod_owner	kube-state-metrics
	count(avg(namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster"}) by (workload, namespace)) by (namespace)	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(container_memory_rss{cluster="$cluster", container!="", container!="POD"}) by (namespace)	container_memory_rss	cadvisor
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster"}) by (namespace)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(container_memory_rss{cluster="$cluster", container!="", container!="POD"}) by (namespace) / sum(kube_pod_container_resource_requ	container_memory_rss	cadvisor

	ests_memory_bytes{cluster="$cluster"}) by (namespace)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster"}) by (namespace)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
	sum(container_memory_rss{cluster="$cluster", container!="", container!="POD"}) by (namespace) / sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster"}) by (namespace)	container_memory_rss	cadvisor
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Node Memory Usage (Top 10)	sum(label_replace(topk(10, 1-(node_memory_MemAvailable_bytes{cluster="$cluster"} / node_memory_MemTotal_bytes{cluster="$cluster"})), "node_ip", "$1", "instance", "(.*)"))by(node_ip)	node_memory_MemAvailable_bytes	node-exporter
		node_memory_MemTotal_bytes	node-exporter
Node CPU Usage (Top 10)	topk(10, sum(label_replace(1 - sum(rate(node_cpu_seconds_total{cluster="$cluster",mode="idle"}[1m])) by (instance) / sum(rate(node_cpu_seconds_total{cluster="$cluster"}[1m])) by (instance),"host_ip","$1","instance","(.*)"))by(host_ip))	node_cpu_seconds_total	node-exporter
		node_cpu_seconds_total	node-exporter
Node Disk Usage (Top 10)	topk(10, sum(label_replace(1-node_filesystem_free_bytes{cluster="$cluster",mountpoint="/"}/node_filesystem_size_bytes{cluster="$cluster",mountpoint="/",fstype!="rootfs"},"host_ip","$1","instance","(.*)"))by(host_ip))	node_filesystem_free_bytes	node-exporter
Node Network In (Top 10)	topk(10, sum(label_replace(max(irate(node_network_receive_bytes_total{cluster="$cluster"}[1m])) by (instance),"host_ip","$1","instance","(.*)"))by(host_ip))	node_network_receive_bytes_total	node-exporter
Node Network Out (Top 10)	topk(10, sum(label_replace(max(irate(node_network_transmit_bytes_total{cluster="$cluster"}[1m])) by (instance),"host_ip","$1","instance","(.*)"))by(host_ip))	node_network_transmit_bytes_total	node-exporter
Node Sockets Count(Top 10)	topk(10, sum(label_replace(max(node_sockstat_TCP_alloc{cluster="$cluster"}) by (instance),"host_ip","$1","instance","(.*)"))by(host_ip))	node_sockstat_TCP_alloc	node-exporter
Container Memory Usage(Top10)	topk(10, sum (container_memory_working_set_bytes{cluster="$cluster",container!="",container!="POD"}) by (container))	container_memory_working_set_bytes	cadvisor

Container Memory Usage/Limit(Top10)	topk(10, avg(container_memory_working_set_bytes{cluster="$cluster",container!=""}/(container_spec_memory_limit_bytes{cluster="$cluster"}!=0)) by (container, pod, namespace))	container_memory_working_set_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor
Container CPU Usage(Top10)	topk(10, sum(rate(container_cpu_usage_seconds_total{cluster="$cluster",container!="",container!="POD"}[2m])) by (container))	container_cpu_usage_seconds_total	cadvisor
Container Network	topk(10, sum(irate(container_network_receive_bytes_total{cluster="$cluster",image!="",container!="",container!="POD"}[2m])) by (pod))	container_network_receive_bytes_total	cadvisor
	-topk(10, sum(irate(container_network_transmit_bytes_total{cluster="$cluster",image!="",container!="",container!="POD"}[2m])) by (pod))	container_network_transmit_bytes_total	cadvisor
Container Memory Usage/Limit (Top 10)	topk(10, avg(container_memory_working_set_bytes{cluster="$cluster",container!=""}/(container_spec_memory_limit_bytes{cluster="$cluster"}!=0)) by (container, pod, namespace))	container_memory_working_set_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor
Container CPU Usage (Top 10)	topk(10, sum(irate(container_cpu_usage_seconds_total{cluster="$cluster",container!="",container!="POD"}[1m])) by (container,pod,namespace)or on() vector(0))	container_cpu_usage_seconds_total	cadvisor
Container Socket Count(Top 10)	topk(10, sum(container_sockets{cluster="$cluster",container!=""}) by (container,pod,namespace)or on() vector(0))	container_sockets	cadvisor

集群 Namespace 大盘

图表名称	查询语句	使用的指标	配置文件
CPU Usage	sum(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"}[2m]))	container_cpu_usage_seconds_total	cadvisor
CPU Usage/Request(%)	sum(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"}[2m]))/sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster",namespace=~"$namespace", unit="core", resource="cpu"})	container_cpu_usage_seconds_total	cadvisor
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU Usage/Limit(%)	sum(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"}[2m]))/sum(kube_pod_container_resource_li	container_cpu_usage_seconds_total	cadvisor

	mits_cpu_cores{cluster="$cluster",namespace=~"$namespace", unit="core", resource="cpu"}) or on() vector(0)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
CPU Request	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster",namespace=~"$namespace", unit="core", resource="cpu"})	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU Limit	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster",namespace=~"$namespace", unit="core", resource="cpu"})	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
Cluster Available	sum(sum(kube_node_status_capacity{resource="cpu",cluster="$cluster",namespace=~"$namespace"}) by (node) + sum(kube_node_spec_unschedulable{cluster="$cluster",namespace=~"$namespace"}==0) by(node))	kube_node_status_capacity	kube-state-metrics
		kube_node_spec_unschedulable	kube-state-metrics
StatefulSet Created	count(kube_statefulset_created{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_statefulset_created	kube-state-metrics
Pod Created	count(kube_pod_info{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_pod_info	kube-state-metrics
Containers	count(kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_pod_container_info	kube-state-metrics
DaemonSet Created	count(kube_daemonset_created{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_daemonset_created	kube-state-metrics
Job Created	count(kube_job_info{cluster="$cluster",namespace="$namespace"})or on() vector(0)	kube_job_info	kube-state-metrics
Job Active	count(kube_job_status_active{cluster="$cluster",namespace="$namespace"}==1)or on() vector(0)	kube_job_status_active	kube-state-metrics
Cron Job Created	count(kube_cronjob_created{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_cronjob_created	kube-state-metrics
Cron Job Active	count(kube_cronjob_status_active{cluster="$cluster",namespace="$namespace"}==1) or on() vector(0)	kube_cronjob_status_active	kube-state-metrics
Unbound PVC	count(kube_persistentvolumeclaim_status_phase{phase!="Bound", cluster="$cluster",namespace="$namespace"}==1) or on() vector(0)	kube_persistentvolumeclaim_status_phase	kube-state-metrics
PersistentVolumeClaim Created	count(kube_persistentvolumeclaim_info{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_persistentvolumeclaim_info	kube-state-metrics
Service Created	count(kube_service_info{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_service_info	kube-state-metrics
LoadBalancer Created	count(kube_service_spec_type{type="LoadBalancer",	kube_service_spec_type	kube-state-metrics

	cluster="$cluster",namespace="$namespace"}) or on() vector(0)		
Ingress Created	count(kube_ingress_info{cluster="$cluster", namespace="$namespace"})or on() vector(0)	kube_ingress_info	kube-state-metrics
ConfigMap Created	count(kube_configmap_info{cluster="$cluster",namespace="$namespace"})	kube_configmap_info	kube-state-metrics
Secret Created	count(kube_secret_info{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_secret_info	kube-state-metrics
PVC Storage Requests Total	sum(kube_persistentvolumeclaim_resource_requests_storage_bytes{cluster="$cluster", namespace="$namespace"}) or on() vector(0)	kube_persistentvolumeclaim_resource_requests_storage_bytes	kube-state-metrics
Pod NotReady	count(kube_pod_status_ready{condition="false", cluster="$cluster",namespace="$namespace"}==1) by(namespace) - (count(kube_pod_status_phase{phase="Succeeded", cluster="$cluster",namespace="$namespace"}==1) by(namespace) or vector(0)) or count(kube_pod_status_ready{condition="false", cluster="$cluster",namespace="$namespace"}==1) by(namespace)	kube_pod_status_ready	kube-state-metrics
		kube_pod_status_phase	kube-state-metrics
		kube_pod_status_ready	kube-state-metrics
Pod UnSchedulable	count(kube_pod_status_unschedulable{cluster="$cluster",namespace="$namespace"}) or on() vector(0)	kube_pod_status_unschedulable	kube-state-metrics
Deployment NotReady	count(sum(kube_deployment_status_replicas_ready{cluster="$cluster",namespace="$namespace"}) by (deployment) <sum(kube_deployment_spec_replicas{cluster="$cluster",namespace="$namespace"}) by (deployment)) or on() vector(0)	kube_deployment_status_replicas_ready	kube-state-metrics
		kube_deployment_spec_replicas	kube-state-metrics
Daemonset NotReady	count(kube_daemonset_status_number_unavailable{cluster="$cluster",namespace="$namespace"}>0) or on() vector(0)	kube_daemonset_status_number_unavailable	kube-state-metrics
Job Failed	count(kube_job_status_failed{cluster="$cluster",namespace="$namespace"} == 1)	kube_job_status_failed	kube-state-metrics
CPU Usage	sum(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"}[2m])) or on() vector(0)	container_cpu_usage_seconds_total	cadvisor
CPU Quota	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"}) by (pod)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics

	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""}) by (pod) / sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"}) by (pod)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""}) by (pod) / sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
Memory Usage	sum(container_memory_working_set_bytes{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"}) or on() vector(0)	container_memory_working_set_bytes	cadvisor
Memory Usage/Request(%)	sum(container_memory_working_set_bytes{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"})/sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace=~"$namespace", unit="byte", resource="memory"}) or on() vector(0)	container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Memory Usage/Limit(%)	sum(container_memory_working_set_bytes{cluster="$cluster",namespace=~"$namespace",container!="",container!="POD"})/sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster",namespace=~"$namespace", unit="byte", resource="memory"}) or on() vector(0)	container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Memory Request	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace=~"$namespace", unit="byte", resource="memory"})	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Memory Limit	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster",namespace=~"$namespace", unit="byte", resource="memory"})	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Cluster Available	sum(sum(kube_node_status_capacity{resource="memory"}) by (node) + sum(kube_node_spec_unschedulable==0) by(node)) or on() vector(0)	kube_node_status_capacity	kube-state-metrics
		kube_node_spec_unschedulable	kube-state-metrics
Memory Usage (w/o cache)	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"}) by (pod)	container_memory_working_set_bytes	cadvisor

	scalar(kube_resourcequota{cluster="$cluster", namespace="$namespace", type="hard",resource="requests.memory"})	kube_resourcequota	kube-state-metrics
	scalar(kube_resourcequota{cluster="$cluster", namespace="$namespace", type="hard",resource="limits.memory"})	kube_resourcequota	kube-state-metrics
Memory Quota	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace",container!="", container!="POD"}) by (pod)	container_memory_working_set_bytes	cadvisor
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"}) by (pod)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace",container!="", container!="POD"}) by (pod) / sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace="$namespace"}) by (pod)	container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"}) by (pod)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace",container!="", container!="POD"}) by (pod) / sum(kube_pod_container_resource_limits_memory_bytes{namespace="$namespace"}) by (pod)	container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
	sum(container_memory_rss{cluster="$cluster", namespace="$namespace",container!="", container!="POD"}) by (pod)	container_memory_rss	cadvisor
	sum(container_memory_cache{cluster="$cluster", namespace="$namespace",container!="", container!="POD"}) by (pod)	container_memory_cache	cadvisor
	sum(container_memory_swap{cluster="$cluster", namespace="$namespace",container!="", container!="POD"}) by (pod)	container_memory_swap	cadvisor
Containers	group by (image, container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"})	kube_pod_container_info	kube-state-metrics
	group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() max by (container,pod) (kube_pod_container_status_running{cluster="$cluster",namespace="$namespace"})	kube_pod_container_info	kube-state-metrics
		kube_pod_container_status_running	kube-state-metrics

group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() max by (container,pod) (kube_pod_container_status_restarts_total{cluster="$cluster",namespace="$namespace"})	kube_pod_container_info	kube-state-metrics
	kube_pod_container_status_restarts_total	kube-state-metrics
group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() max(irate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[1m])) by (pod,container)	kube_pod_container_info	kube-state-metrics
	container_cpu_usage_seconds_total	cadvisor
group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() (max(irate(container_cpu_usage_seconds_total{container!="",container!="POD",cluster="$cluster",namespace="$namespace"}[1m])) by (container,pod) / (max(container_spec_cpu_quota{container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}/100000 > 0) by (container,pod)))	kube_pod_container_info	kube-state-metrics
	container_cpu_usage_seconds_total	cadvisor
	container_spec_cpu_quota	cadvisor
group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() sum by (container,pod) (kube_pod_container_resource_requests_cpu_cores{cluster="$cluster",namespace="$namespace"})	kube_pod_container_info	kube-state-metrics
	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() (max(irate(container_cpu_usage_seconds_total{container!="",container!="POD",cluster="$cluster",namespace="$namespace"}[1m])) by (container,pod) / (max by (container,pod) (kube_pod_container_resource_requests_cpu_cores{cluster="$cluster",namespace="$namespace"})))	kube_pod_container_info	kube-state-metrics
	container_cpu_usage_seconds_total	cadvisor
	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() sum by (container,pod) (kube_pod_container_resource_limits{resource="cpu",cluster="$cluster",namespace="$namespace"})	kube_pod_container_info	kube-state-metrics
	kube_pod_container_resource_limits	kube-state-metrics

	group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() max(container_memory_working_set_bytes{container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,container)	kube_pod_container_info	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
	group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() (max(container_memory_working_set_bytes{container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,container) / max(container_spec_memory_limit_bytes{container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod, container) < 1)	kube_pod_container_info	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor
	group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() sum by (container,pod) (kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace="$namespace"})	kube_pod_container_info	kube-state-metrics
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	group by (container, container_id, pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace"}) * on(container,pod) group_left() (max(container_memory_working_set_bytes{container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,container) / max by (container,pod) (kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace="$namespace"}))	kube_pod_container_info	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics

API Server（独立集群）

图表名称	查询语句	使用的指标	配置文件
Availability > 99.000%	1 - ((sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_count{cluster="$cluster"}[5m])) - sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_bucket{le="1",cluster="$cluster"}[5m]))) + sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-	apiserver_request_duration_seconds_count	kube-apiserver
		apiserver_request_duration_seconds_bucket	kube-apiserver

	apiserver",code=~"5..", cluster="$cluster"}[5m]) or vector(0))) / sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",cluster="$cluster"}[5m]))	apiserver_request_total	kube-apiserver
ErrorBudget > 99.000%	100 * (1 - ((sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_count{cluster="$cluster"}[5m])) - sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_bucket{le="1",cluster="$cluster"}[5m]))) + sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",code=~"5..", cluster="$cluster"}[5m]) or vector(0))) / sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",cluster="$cluster"}[5m])) -0.990000)	apiserver_request_duration_seconds_count	kube-apiserver
		apiserver_request_duration_seconds_bucket	kube-apiserver
		apiserver_request_total	kube-apiserver
Read Availability	1 - ((sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_count{verb=~"LIST|GET", cluster="$cluster"}[5m])) - sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_bucket{verb=~"LIST\|GET",le="1", cluster="$cluster"}[5m]))) + sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",verb=~"LIST|GET",code=~"5..", cluster="$cluster"}[5m]) or vector(0))) / sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",verb=~"LIST\|GET", cluster="$cluster"}[5m]))	apiserver_request_duration_seconds_count	kube-apiserver
		apiserver_request_duration_seconds_bucket	kube-apiserver
		apiserver_request_total	kube-apiserver
Read SLI - Requests	sum by (code) (rate(apiserver_request_total{job="kube-apiserver",verb=~"LIST|GET",cluster="$cluster"}[5m]))	apiserver_request_total	kube-apiserver
Read SLI - Errors	sum by (resource) (rate(apiserver_request_total{job="kube-apiserver",verb=~"LIST|GET",code=~"5..",cluster="$cluster"}[5m]))/ sum by (resource) (rate(apiserver_request_total{job="kube-apiserver",verb=~"LIST\|GET",cluster="$cluster"}[5m]))	apiserver_request_total	kube-apiserver
Read SLI - Duration	histogram_quantile(0.99, sum by (le, resource,cluster,cluster_type) (rate(apiserver_request_duration_seconds_bucket{job="kube-apiserver",verb=~"LIST|GET",cluster="$cluster"}[5m]))) > 0	apiserver_request_duration_seconds_bucket	kube-apiserver

Write Availability	1 - ((sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_count{verb=~"POST|PUT|PATCH|DELETE", cluster="$cluster"}[5m])) - sum by (cluster,cluster_type) (increase(apiserver_request_duration_seconds_bucket{verb=~"POST\|PUT\|PATCH\|DELETE",le="1", cluster="$cluster"}[5m]))) + sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",verb=~"POST|PUT|PATCH|DELETE",code=~"5..", cluster="$cluster"}[5m]) or vector(0))) / sum by (cluster,cluster_type) (increase(apiserver_request_total{job="kube-apiserver",verb=~"POST\|PUT\|PATCH\|DELETE", cluster="$cluster"}[5m]))	apiserver_request_duration_seconds_count	kube-apiserver
		apiserver_request_duration_seconds_bucket	kube-apiserver
		apiserver_request_total	kube-apiserver
Write SLI - Requests	sum by (code) (rate(apiserver_request_total{job="kube-apiserver",verb=~"POST|PUT|PATCH|DELETE",cluster="$cluster"}[5m]))	apiserver_request_total	kube-apiserver
Write SLI - Errors	sum by (resource) (rate(apiserver_request_total{job="kube-apiserver",verb=~"POST|PUT|PATCH|DELETE",code=~"5..",cluster="$cluster"}[5m]))/ sum by (resource) (rate(apiserver_request_total{job="kube-apiserver",verb=~"POST\|PUT\|PATCH\|DELETE",cluster="$cluster"}[5m]))	apiserver_request_total	kube-apiserver
Write SLI - Duration	histogram_quantile(0.99, sum by (le, resource,cluster,cluster_type) (rate(apiserver_request_duration_seconds_bucket{job="kube-apiserver",verb=~"POST|PUT|PATCH|DELETE",cluster="$cluster"}[5m]))) > 0	apiserver_request_duration_seconds_bucket	kube-apiserver
Work Queue Add Rate	sum(rate(workqueue_adds_total{job="kube-apiserver", instance=~"$instance", cluster=~"$cluster"}[5m])) by (instance, name)	workqueue_adds_total	kube-apiserver
Work Queue Depth	sum(rate(workqueue_depth{job="kube-apiserver", instance=~"$instance", cluster=~"$cluster"}[5m])) by (instance, name)	workqueue_depth	kube-apiserver
Work Queue Latency	histogram_quantile(0.99, sum(rate(workqueue_queue_duration_seconds_bucket{job="kube-apiserver", instance=~"$instance", cluster=~"$cluster"}[5m])) by (instance, name, le))	workqueue_queue_duration_seconds_bucket	kube-apiserver
Memory	process_resident_memory_bytes{job="kube-apiserver",instance=~"$instance", cluster=~"$cluster"}	process_resident_memory_bytes	kube-apiserver
CPU usage	rate(process_cpu_seconds_total{job="kube-apiserver",instance=~"$instance", cluster=~"$cluster"}[5m])	process_cpu_seconds_total	kube-apiserver

Controller Manager（独立集群）

图表名称	查询语句	使用的指标	配置文件

Up	sum(up{cluster=~"$cluster",job="kube-controller-manager"})	up	kube-controller-manager
Work Queue Add Rate	sum(rate(workqueue_adds_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance"}[5m])) by (instance, name)	workqueue_adds_total	kube-controller-manager
Work Queue Depth	sum(rate(workqueue_depth{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance"}[5m])) by (instance, name)	workqueue_depth	kube-controller-manager
Work Queue Latency	histogram_quantile(0.99, sum(rate(workqueue_queue_duration_seconds_bucket{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance"}[5m])) by (instance, name, le))	workqueue_queue_duration_seconds_bucket	kube-controller-manager
Kube API Request Rate	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-controller-manager
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kube-controller-manager
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-controller-manager
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"5.."}[5m]))	rest_client_requests_total	kube-controller-manager
Post Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance", verb="POST"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-controller-manager
Get Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance", verb="GET"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-controller-manager
Memory	process_resident_memory_bytes{cluster=~"$cluster",job="kube-controller-manager",instance=~"$instance"}	process_resident_memory_bytes	kube-controller-manager
CPU usage	rate(process_cpu_seconds_total{cluster=~"$cluster",job="kube-controller-manager",instance=~"$instance"}[5m])	process_cpu_seconds_total	kube-controller-manager

Scheduler（独立集群）

图表名称	查询语句	使用的指标	配置文件
Up	sum(up{cluster=~"$cluster", job="kube-scheduler"})	up	kube-scheduler
Kube API Request Rate	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-scheduler

	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kube-scheduler
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-scheduler
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"5.."}[5m]))	rest_client_requests_total	kube-scheduler
Post Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance", verb="POST"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-scheduler
Get Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance", verb="GET"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-scheduler
Memory	process_resident_memory_bytes{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance"}	process_resident_memory_bytes	kube-scheduler
CPU usage	rate(process_cpu_seconds_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance"}[5m])	process_cpu_seconds_total	kube-scheduler

Kubelet

图表名称	查询语句	使用的指标	配置文件
Up	sum(up{cluster="$cluster", job="kubelet"})	up	kubelet
Running Pods	sum(kubelet_running_pods{cluster="$cluster", job="kubelet", instance=~"$instance"})	kubelet_running_pods	kubelet
Running Container	sum(kubelet_running_containers{cluster="$cluster", job="kubelet", instance=~"$instance"})	kubelet_running_containers	kubelet
Actual Volume Count	sum(volume_manager_total_volumes{cluster="$cluster", job="kubelet", instance=~"$instance", state="actual_state_of_world"})	volume_manager_total_volumes	kubelet
Desired Volume Count	sum(volume_manager_total_volumes{cluster="$cluster", job="kubelet", instance=~"$instance",state="desired_state_of_world"})	volume_manager_total_volumes	kubelet
Config Error Count	sum(rate(kubelet_node_config_error{cluster="$cluster", job="kubelet", instance=~"$instance"}[5m]))	kubelet_node_config_error	kubelet
Operation Rate	sum(rate(kubelet_runtime_operations_total{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (operation_type, instance)	kubelet_runtime_operations_total	kubelet
Operation Error Rate	sum(rate(kubelet_runtime_operations_errors_total{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, operation_type)	kubelet_runtime_operations_errors_total	kubelet
Operation duration 99th quantile	histogram_quantile(0.99, sum(rate(kubelet_runtime_operations_duration_seconds_bucket{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, operation_type, le))	kubelet_runtime_operations_duration_seconds_bucket	kubelet

Pod Start Rate	sum(rate(kubelet_pod_start_duration_seconds_count{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance)	kubelet_pod_start_duration_seconds_count	kubelet
	sum(rate(kubelet_pod_worker_duration_seconds_count{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance)	kubelet_pod_worker_duration_seconds_count	kubelet
Pod Start Duration	histogram_quantile(0.99, sum(rate(kubelet_pod_start_duration_seconds_count{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, le))	kubelet_pod_start_duration_seconds_count	kubelet
	histogram_quantile(0.99, sum(rate(kubelet_pod_worker_duration_seconds_bucket{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, le))	kubelet_pod_worker_duration_seconds_bucket	kubelet
Storage Operation Rate	sum(rate(storage_operation_duration_seconds_count{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, operation_name, volume_plugin)	storage_operation_duration_seconds_count	kubelet
Storage Operation Error Rate	sum(rate(storage_operation_errors_total{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, operation_name, volume_plugin)	storage_operation_errors_total	kubelet
Storage Operation Duration 99th quantile	histogram_quantile(0.99, sum(rate(storage_operation_duration_seconds_bucket{cluster="$cluster", job="kubelet", instance=~"$instance"}[5m])) by (instance, operation_name, volume_plugin, le))	storage_operation_duration_seconds_bucket	kubelet
Cgroup manager operation rate	sum(rate(kubelet_cgroup_manager_duration_seconds_count{cluster="$cluster", job="kubelet", instance=~"$instance"}[5m])) by (instance, operation_type)	kubelet_cgroup_manager_duration_seconds_count	kubelet
Cgroup manager 99th quantile	histogram_quantile(0.99, sum(rate(kubelet_cgroup_manager_duration_seconds_bucket{cluster="$cluster", job="kubelet", instance=~"$instance"}[5m])) by (instance, operation_type, le))	kubelet_cgroup_manager_duration_seconds_bucket	kubelet
PLEG relist rate	sum(rate(kubelet_pleg_relist_duration_seconds_count{cluster="$cluster", job="kubelet", instance=~"$instance"}[5m])) by (instance)	kubelet_pleg_relist_duration_seconds_count	kubelet
PLEG relist interval	histogram_quantile(0.99, sum(rate(kubelet_pleg_relist_interval_seconds_bucket{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, le))	kubelet_pleg_relist_interval_seconds_bucket	kubelet
PLEG relist duration	histogram_quantile(0.99, sum(rate(kubelet_pleg_relist_duration_seconds_bucket{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])) by (instance, le))	kubelet_pleg_relist_duration_seconds_bucket	kubelet
RPC Rate	sum(rate(rest_client_requests_total{cluster="$cluster",job="kubelet", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kubelet
	sum(rate(rest_client_requests_total{cluster="$cluster",job="kubelet", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kubelet
Request duration	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{clus	rest_client_request_duration_seconds_bucke	kubelet

99th quantile	ter="$cluster",job="kubelet", instance=~"$instance"}[5m])) by (instance, verb, url, le))	t	
Memory	process_resident_memory_bytes{cluster="$cluster",job="kubelet",instance=~"$instance"}	process_resident_memory_bytes	kubelet
CPU usage	rate(process_cpu_seconds_total{cluster="$cluster",job="kubelet",instance=~"$instance"}[5m])	process_cpu_seconds_total	kubelet
Goroutines	go_goroutines{cluster="$cluster",job="kubelet",instance=~"$instance"}	go_goroutines	kubelet

Proxy（非默认安装组件）

图表名称	查询语句	使用的指标	配置文件
Up	sum(up{job="kube-proxy"})	up	kube-proxy
Rules Sync Rate	sum(rate(kubeproxy_sync_proxy_rules_duration_seconds_count{job="kube-proxy", instance=~"$instance"}[5m]))	kubeproxy_sync_proxy_rules_duration_seconds_count	kube-proxy
Rule Sync Latency 99th Quantile	histogram_quantile(0.99,rate(kubeproxy_sync_proxy_rules_duration_seconds_bucket{job="kube-proxy", instance=~"$instance"}[5m]))	kubeproxy_sync_proxy_rules_duration_seconds_bucket	kube-proxy
Network Programming Rate	sum(rate(kubeproxy_network_programming_duration_seconds_count{job="kube-proxy", instance=~"$instance"}[5m]))	kubeproxy_network_programming_duration_seconds_count	kube-proxy
Network Programming Latency 99th Quantile	histogram_quantile(0.99, sum(rate(kubeproxy_network_programming_duration_seconds_bucket{job="kube-proxy", instance=~"$instance"}[5m])) by (instance, le))	kubeproxy_network_programming_duration_seconds_bucket	kube-proxy
Kube API Request Rate	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-proxy
	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kube-proxy
	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-proxy
	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"5.."}[5m]))	rest_client_requests_total	kube-proxy
Post Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{job="kube-proxy",instance=~"$instance",verb="POST"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-proxy
Kube API Request Rate	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-proxy
	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kube-proxy
	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-proxy
	sum(rate(rest_client_requests_total{job="kube-proxy", instance=~"$instance",code=~"5.."}[5m]))	rest_client_requests_total	kube-proxy

Post Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{job="kube-proxy",instance=~"$instance",verb="POST"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-proxy
Get Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{job="kube-proxy", instance=~"$instance", verb="GET"}[5m])) by (verb, url, le))	rest_client_request_duration_seconds_bucket	kube-proxy
Memory	process_resident_memory_bytes{job="kube-proxy",instance=~"$instance"}	process_resident_memory_bytes	kube-proxy
CPU usage	rate(process_cpu_seconds_total{job="kube-proxy",instance=~"$instance"}[5m])	process_cpu_seconds_total	kube-proxy

集群节点监控详情

图表名称	查询语句	使用的指标	配置文件
服务器资源总览表	node_uname_info{job=~"$job", cluster=~"$cluster"} - 0	node_uname_info	node-exporter
	node_memory_MemTotal_bytes{job=~"$job",cluster=~"$cluster"} - 0	node_memory_MemTotal_bytes	node-exporter
	count(node_cpu_seconds_total{job=~"$job",mode='system',cluster=~"$cluster"}) by (instance)	node_cpu_seconds_total	node-exporter
	sum(time() - node_boot_time_seconds{job=~"$job",cluster=~"$cluster"})by(instance)	node_boot_time_seconds	node-exporter
	max((node_filesystem_size_bytes{job=~"$job",cluster=~"$cluster",fstype=~"ext.?|xfs"}- node_filesystem_free_bytes{job=~"$job",cluster=~"$cluster",fstype=~"ext.?|xfs"}) *100/(node_filesystem_avail_bytes{job=~"$job",cluster=~"$cluster",fstype=~"ext.?|xfs"}+ (node_filesystem_size_bytes{job=~"$job",cluster=~"$cluster",fstype=~"ext.?|xfs"}- node_filesystem_free_bytes{job=~"$job",cluster=~"$cluster",fstype=~"ext.?|xfs"}))))by(instance)	node_filesystem_size_bytes	node-exporter
		node_filesystem_avail_bytes	node-exporter
		node_filesystem_free_bytes	node-exporter
	(1 - avg(irate(node_cpu_seconds_total{job=~"$job",mode="idle",cluster=~"$cluster"}[5m])) by (instance)) * 100	node_cpu_seconds_total	node-exporter
	(1 - (node_memory_MemAvailable_bytes{job=~"$job",cluster=~"$cluster"} / (node_memory_MemTotal_bytes{job=~"$job",cluster=~"$cluster"}))) * 100	node_memory_MemAvailable_bytes	node-exporter
		node_memory_MemTotal_bytes	node-exporter
	node_load5{job=~"$job",cluster=~"$cluster"}	node_load5	node-exporter
	max(irate(node_disk_written_bytes_total{job=~"$job",cluster=~"$cluster"}[5m])) by (instance)	node_disk_written_bytes_total	node-exporter
	max(irate(node_network_receive_bytes_total{job=~"$job",cluster=~"$cluster"}[5m])*8) by (instance)	node_network_receive_bytes_total	node-exporter
	max(irate(node_network_transmit_bytes_total{job=~"$job",cluster=~"$cluster"}[5m])*8) by (instance)	node_network_transmit_bytes_total	node-exporter

	node_load5{job=~"$job",cluster=~"$cluster"}	node_load5	node-exporter
整体总负载与整体平均 CPU 使用率	count(node_cpu_seconds_total{job=~"$job",cluster=~"$cluster", mode='system'})	node_cpu_seconds_total	node-exporter
	sum(node_load5{job=~"$job",cluster=~"$cluster"})	node_load5	node-exporter
	avg(1 - avg(irate(node_cpu_seconds_total{job=~"$job",mode="idle",cluster=~"$cluster"}[5m])) by (instance)) * 100	node_cpu_seconds_total	node-exporter
整体总内存与整体平均内存使用率	sum(node_memory_MemTotal_bytes{job=~"$job",cluster=~"$cluster"})	node_memory_MemTotal_bytes	node-exporter
	sum(node_memory_MemTotal_bytes{job=~"$job",cluster=~"$cluster"} - node_memory_MemAvailable_bytes{job=~"$job",cluster=~"$cluster"})	node_memory_MemTotal_bytes	node-exporter
		node_memory_MemAvailable_bytes	node-exporter
	(sum(node_memory_MemTotal_bytes{job=~"$job",cluster=~"$cluster"} - node_memory_MemAvailable_bytes{job=~"$job",cluster=~"$cluster"}) / sum(node_memory_MemTotal_bytes{job=~"$job",cluster=~"$cluster"}))*100	node_memory_MemTotal_bytes	node-exporter
		node_memory_MemAvailable_bytes	node-exporter
整体总磁盘与整体平均磁盘使用率	sum(avg(node_filesystem_size_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance))	node_filesystem_size_bytes	node-exporter
	sum(avg(node_filesystem_size_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance)) - sum(avg(node_filesystem_free_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance))	node_filesystem_size_bytes	node-exporter
		node_filesystem_free_bytes	node-exporter
	(sum(avg(node_filesystem_size_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance)) - sum(avg(node_filesystem_free_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance))) *100/(sum(avg(node_filesystem_avail_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance)) + (sum(avg(node_filesystem_size_bytes{job=~"$job",fstype=~"xfs\|ext.*"})by(device,instance)) - sum(avg(node_filesystem_free_bytes{job=~"$job",cluster=~"$cluster",fstype=~"xfs|ext.*"})by(device,instance))))	node_filesystem_size_bytes	node-exporter
		node_filesystem_free_bytes	node-exporter
		node_filesystem_avail_bytes	node-exporter
运行时间	avg(time() - node_boot_time_seconds{instance=~"$node",cluster=~"$cluster"}) 75	node_boot_time_seconds	node-exporter
CPU 核数	count(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node", mode='system'})	node_cpu_seconds_total	node-exporter
总内存	sum(node_memory_MemTotal_bytes{cluster=~"$cluster",instance=~"$node"})	node_memory_MemTotal_bytes	node-exporter
总 CPU 使用率	100 - (avg(irate(node_cpu_seconds_total{instance=~"$node",mode="idle",cluster=~"$cluster"}[5m])) * 100)	node_cpu_seconds_total	node-exporter

内存使用率	(1 - (node_memory_MemAvailable_bytes{instance=~"$node",cluster=~"$cluster"} / (node_memory_MemTotal_bytes{instance=~"$node",cluster=~"$cluster"}))* 100	node_memory_MemAvailable_bytes	node-exporter
最大分区使用率	(node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint="$maxmount"}-node_filesystem_free_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*\|xfs",mountpoint="$maxmount"})*100 /(node_filesystem_avail_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint="$maxmount"}+(node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*\|xfs",mountpoint="$maxmount"}-node_filesystem_free_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint="$maxmount"}))	node_filesystem_size_bytes	node-exporter
		node_filesystem_free_bytes	node-exporter
		node_filesystem_avail_bytes	node-exporter
CPU iowait	avg(irate(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node",mode="iowait"}[5m])) * 100	node_cpu_seconds_total	node-exporter
各分区可用空间	node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}-0	node_filesystem_size_bytes	node-exporter
	node_filesystem_avail_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}-0	node_filesystem_avail_bytes	node-exporter
	(node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}-node_filesystem_free_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}) *100/(node_filesystem_avail_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}+(node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}-node_filesystem_free_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}))	node_filesystem_size_bytes	node-exporter
		node_filesystem_free_bytes	node-exporter
		node_filesystem_avail_bytes	node-exporter
CPU 使用率	avg(irate(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node",mode="system"}[5m])) by (instance) *100	node_cpu_seconds_total	node-exporter
	avg(irate(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node",mode="user"}[5m])) by (instance) *100	node_cpu_seconds_total	node-exporter
	avg(irate(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node",mode="iowait"}[5m])) by (instance) *100	node_cpu_seconds_total	node-exporter
	(1 - avg(irate(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node",mode="idle"}[5m])) by (instance))*100	node_cpu_seconds_total	node-exporter
内存信息	node_memory_MemTotal_bytes{cluster=~"$cluster",instance=~"$node"}	node_memory_MemTotal_bytes	node-exporter
	node_memory_MemTotal_bytes{cluster=~"$cluster",instance=~"$node"} - node_memory_MemAvailable_bytes{cluster=~"$cluster",instance=~"$node"}	node_memory_MemTotal_bytes	node-exporter
		node_memory_MemAvailable_bytes	node-exporter
	node_memory_MemAvailable_bytes{cluster=~"$cluster",instance=~"$node"}	node_memory_MemAvailable_bytes	node-exporter

	(1 - (node_memory_MemAvailable_bytes{cluster=~"$cluster",instance=~"$node"} / (node_memory_MemTotal_bytes{cluster=~"$cluster",instance=~"$node"}))* 100	node_memory_MemAvailable_bytes	node-exporter
		node_memory_MemTotal_bytes	node-exporter
每秒网络带宽使用	irate(node_network_receive_bytes_total{cluster=~"$cluster",instance=~'$node',device=~"$device"}[5m])*8	node_network_receive_bytes_total	node-exporter
	irate(node_network_transmit_bytes_total{cluster=~"$cluster",instance=~'$node',device=~"$device"}[5m])*8	node_network_transmit_bytes_total	node-exporter
系统平均负载	node_load1{cluster=~"$cluster",instance=~"$node"}	node_load1	node-exporter
	node_load5{cluster=~"$cluster",instance=~"$node"}	node_load5	node-exporter
	node_load15{cluster=~"$cluster",instance=~"$node"}	node_load15	node-exporter
	sum(count(node_cpu_seconds_total{cluster=~"$cluster",instance=~"$node", mode='system'}) by (cpu,instance)) by(instance)	node_cpu_seconds_total	node-exporter
每秒磁盘读写容量	irate(node_disk_read_bytes_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_read_bytes_total	node-exporter
	irate(node_disk_written_bytes_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_written_bytes_total	node-exporter
磁盘使用率	(node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}- node_filesystem_free_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}) *100/(node_filesystem_avail_bytes {cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs", mountpoint !~".*pod.*"}+ (node_filesystem_size_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}- node_filesystem_free_bytes{cluster=~"$cluster",instance=~'$node',fstype=~"ext.*|xfs",mountpoint !~".*pod.*"}))	node_filesystem_size_bytes	node-exporter
		node_filesystem_free_bytes	node-exporter
		node_filesystem_avail_bytes	node-exporter
	node_filesystem_files_free{cluster=~"$cluster",instance=~'$node',fstype=~"ext.?|xfs"} / node_filesystem_files{cluster=~"$cluster",instance=~'$node',fstype=~"ext.?|xfs"}	node_filesystem_files_free	node-exporter
磁盘读写速率（IOPS）	irate(node_disk_reads_completed_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_reads_completed_total	node-exporter
	irate(node_disk_writes_completed_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_writes_completed_total	node-exporter
	node_disk_io_now{cluster=~"$cluster",instance=~"$node"}	node_disk_io_now	node-exporter
每1秒内 I/O 操作耗时占比	irate(node_disk_io_time_seconds_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_io_time_seconds_total	node-exporter

每次 IO 读写的耗时	irate(node_disk_read_time_seconds_total{cluster=~"$cluster",instance=~"$node"}[5m]) / irate(node_disk_reads_completed_total{instance=~"$node"}[5m])	node_disk_read_time_seconds_total	node-exporter
		node_disk_reads_completed_total	node-exporter
	irate(node_disk_write_time_seconds_total{cluster=~"$cluster",instance=~"$node"}[5m]) / irate(node_disk_writes_completed_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_write_time_seconds_total	node-exporter
		node_disk_writes_completed_total	node-exporter
	irate(node_disk_io_time_seconds_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_io_time_seconds_total	node-exporter
	irate(node_disk_io_time_weighted_seconds_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_disk_io_time_weighted_seconds_total	node-exporter
网络 Socket 连接信息	node_netstat_Tcp_CurrEstab{cluster=~"$cluster",instance=~'$node'}	node_netstat_Tcp_CurrEstab	node-exporter
	node_sockstat_TCP_tw{cluster=~"$cluster",instance=~'$node'}	node_sockstat_TCP_tw	node-exporter
	node_sockstat_sockets_used{cluster=~"$cluster",instance=~'$node'}	node_sockstat_sockets_used	node-exporter
	node_sockstat_UDP_inuse{cluster=~"$cluster",instance=~'$node'}	node_sockstat_UDP_inuse	node-exporter
	node_sockstat_TCP_alloc{cluster=~"$cluster",instance=~'$node'}	node_sockstat_TCP_alloc	node-exporter
	irate(node_netstat_Tcp_PassiveOpens{cluster=~"$cluster",instance=~'$node'}[5m])	node_netstat_Tcp_PassiveOpens	node-exporter
	irate(node_netstat_Tcp_ActiveOpens{cluster=~"$cluster",instance=~'$node'}[5m])	node_netstat_Tcp_ActiveOpens	node-exporter
	irate(node_netstat_Tcp_InSegs{cluster=~"$cluster",instance=~'$node'}[5m])	node_netstat_Tcp_InSegs	node-exporter
	irate(node_netstat_Tcp_OutSegs{cluster=~"$cluster",instance=~'$node'}[5m])	node_netstat_Tcp_OutSegs	node-exporter
	irate(node_netstat_Tcp_RetransSegs{cluster=~"$cluster",instance=~'$node'}[5m])	node_netstat_Tcp_RetransSegs	node-exporter
打开的文件描述符(左)/每秒上下文切换次数(右)	node_filefd_allocated{cluster=~"$cluster",instance=~"$node"}	node_filefd_allocated	node-exporter
	irate(node_context_switches_total{cluster=~"$cluster",instance=~"$node"}[5m])	node_context_switches_total	node-exporter
	(node_filefd_allocated{cluster=~"$cluster",instance=~"$node"}/node_filefd_maximum{cluster=~"$cluster",instance=~"$node"}) *100	node_filefd_allocated	node-exporter
		node_filefd_maximum	node-exporter

节点 Pod 监控

图表名称	查询语句	使用的指标	配置文件

Pods	count(kube_pod_info{node=~"$node"})	kube_pod_info	kube-state-metrics
Pod Request Memory	sum(kube_pod_container_resource_requests_memory_bytes{node=~"$node"})by(node)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Pod Request CPU Cores	sum(kube_pod_container_resource_requests_cpu_cores{node=~"$node"})by(node)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU Usage	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", node=~"$node", container!="POD", container!=""}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
CPU Quota	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", node=~"$node", container!="POD", container!=""}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", node=~"$node"}) by (pod)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", node=~"$node", container!="POD", container!=""}) by (pod) / sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", node=~"$node"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", node=~"$node"}) by (pod)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", node=~"$node", container!="POD", container!=""}) by (pod) / sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", node=~"$node"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
Memory Usage	sum(node_namespace_pod_container:container_memory_working_set_bytes{cluster="$cluster", node=~"$node", container!="", container!="POD"}) by (pod)	node_namespace_pod_container:container_memory_working_set_bytes	预聚合指标
Memory Quota	sum(node_namespace_pod_container:container_memory_working_set_bytes{cluster="$cluster", node=~"$node",container!="", container!="POD"}) by (pod)	node_namespace_pod_container:container_memory_working_set_bytes	预聚合指标
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", node=~"$node"}) by (pod)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(node_namespace_pod_container:container_memory_working_set_bytes{cluster="$cluster", node=~"$node",container!="", container!="POD"}) by (pod) / sum(kube_pod_container_resource_requests_memory_bytes{node=~"$node"}) by (pod)	node_namespace_pod_container:container_memory_working_set_bytes	预聚合指标
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics

	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", node=~"$node"}) by (pod)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
	sum(node_namespace_pod_container:container_memory_working_set_bytes{cluster="$cluster", node=~"$node",container!="", container!="POD"}) by (pod) / sum(kube_pod_container_resource_limits_memory_bytes{node=~"$node"}) by (pod)	node_namespace_pod_container:container_memory_working_set_bytes	预聚合指标
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Pod List	group (kube_pod_info{host_ip="$node"})by(created_by_kind, created_by_name,host_network,pod_ip,pod, priority_class,namespace)	kube_pod_info	kube-state-metrics
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() max(kube_pod_status_phase{}==1) by (pod, phase)	kube_pod_info	kube-state-metrics
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() sum(container_memory_working_set_bytes) by (pod)	kube_pod_info	kube-state-metrics
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() sum(rate(container_cpu_usage_seconds_total{image!=""}[5m])) by (pod)	kube_pod_info	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() max(time()-kube_pod_start_time) by (pod)	kube_pod_info	kube-state-metrics
		kube_pod_start_time	kube-state-metrics
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) max(kube_pod_status_ready{condition="true"}) by (pod) or on() vector(0)	kube_pod_info	kube-state-metrics
		kube_pod_status_ready	kube-state-metrics
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() max(rate(container_network_receive_bytes_total{image!=""}[5m])) by (pod) or on() vector(0)	kube_pod_info	kube-state-metrics
		container_network_receive_bytes_total	cadvisor
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() max(rate(container_network_transmit_bytes_total{image!=""}[5m])) by (pod) or on() vector(0)	kube_pod_info	kube-state-metrics
		container_network_transmit_bytes_total	cadvisor
	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() max(rate(container_fs_reads_bytes_total{container!="POD", container!=""}[5m])) by (pod) or on() vector(0)	kube_pod_info	kube-state-metrics
		container_fs_reads_bytes_total	cadvisor

	min(kube_pod_info{host_ip="$node"})by(pod) * on(pod) group_right() max(rate(container_fs_writes_bytes_total{c ontainer!="POD", container!=""}[5m])) by (pod) or on() vector(0)	kube_pod_info	kube-state-metrics
		container_fs_writes_bytes_total	cadvisor

工作负载监控概览

图表名称	查询语句	使用的指标	配置文件
CPU Usage	sum(node_namespace_pod_container:container_cpu_ usage_seconds_total:sum_rate{cluster="$cluster" , namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:rela bel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	node_namespace_pod_conta iner:container_cpu_usage_se conds_total:sum_rate	预聚合指标
		namespace_workload_pod:k ube_pod_owner:relabel	预聚合指标
	scalar(kube_resourcequota{cluster="$cluster", namespace="$namespace", type="hard",resource="requests.cpu"})	kube_resourcequota	kube-state-metrics
	scalar(kube_resourcequota{cluster="$cluster", namespace="$namespace", type="hard",resource="limits.cpu"})	kube_resourcequota	kube-state-metrics
CPU Quota	count(namespace_workload_pod:kube_pod_own er:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	namespace_workload_pod:k ube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_ usage_seconds_total:sum_rate{cluster="$cluster" , namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:rela bel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	node_namespace_pod_conta iner:container_cpu_usage_se conds_total:sum_rate	预聚合指标
		namespace_workload_pod:k ube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_co res{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:rela bel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	kube_pod_container_resourc e_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:k ube_pod_owner:relabel	预聚合指标

	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type) /sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type) /sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Memory Usage	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace",	container_memory_working_set_bytes	cadvisor

	container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	scalar(kube_resourcequota{cluster="$cluster", namespace="$namespace", type="hard",resource="requests.memory"})	kube_resourcequota	kube-state-metrics
	scalar(kube_resourcequota{cluster="$cluster", namespace="$namespace", type="hard",resource="limits.memory"})	kube_resourcequota	kube-state-metrics
Memory Quota	count(namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type) /sum(	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标

	kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type) /sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload_type="$type"}) by (workload, workload_type)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标 Deployment

Deployment

图表名称	查询语句	使用的指标	配置文件
Age	time() - max(kube_deployment_created{cluster="$cluster",namespace="$namespace",deployment="$workload"})	kube_deployment_created	kube-state-metrics
Replicas(Pods)-Request	max(kube_deployment_spec_replicas{deployment="$workload",cluster="$cluster",namespace="$namespace"})	kube_deployment_spec_replicas	kube-state-metrics

Replicas(Pods)-Ready	max(kube_deployment_status_replicas_ready{deployment="$workload",cluster="$cluster",namespace="$namespace"})	kube_deployment_status_replicas_ready	kube-state-metrics
Replica Trend	max(kube_deployment_spec_replicas{deployment="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_deployment_spec_replicas	kube-state-metrics
	max(kube_deployment_status_replicas{deployment="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_deployment_status_replicas	kube-state-metrics
	min(kube_deployment_status_replicas_ready{deployment="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_deployment_status_replicas_ready	kube-state-metrics
	min(kube_deployment_status_replicas_available{deployment="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_deployment_status_replicas_available	kube-state-metrics
	min(kube_deployment_status_replicas_updated{deployment="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_deployment_status_replicas_updated	kube-state-metrics
	min(kube_deployment_status_replicas_unavailable{deployment="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_deployment_status_replicas_unavailable	kube-state-metrics
CPU Usage	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
CPU Quota	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标

	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod) /sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod) /sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
CPU Limit-Total	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1",	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics

	"created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(kube_pod_container_resource_limits_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"}) by (pod))		
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
CPU Request-Total	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(kube_pod_container_resource_requests_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"}) by (pod))	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU Info	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
	max(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(kube_pod_container_resource_requests_cpu_cores{clu	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics

	ster="$cluster",namespace="$namespace",container!="",container!="POD"}) by(pod, container))by(container) max(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}) by (pod, container))by(container)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
CPU Usage/Limit (%)	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container) / max by(container, pod) (kube_pod_container_resource_limits_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"})	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
CPU Usage/Request(%)	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container) / max by(container, pod) (kube_pod_container_resource_requests_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"})	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU User Time(%)	avg(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by	kube_pod_info	kube-state-metrics

	(created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() (max(rate(container_cpu_user_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod,container) / max(rate(container_cpu_user_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"} [5m])+rate(container_cpu_system_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod,container))) by (pod,container)	kube_replicaset_owner	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
		container_cpu_usage_seconds_total	cadvisor
		container_cpu_usage_seconds_total	cadvisor
Memory Usage	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Memory Quota	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod) /sum(kube_pod_container_resource_requests_memory_bytes{clu	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_reque	kube-state-metrics

	ster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	sts_memory_bytes	
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod) /sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="deployment"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Memory Limit-Total	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(container_spec_memory_limit_bytes{cluster="$cluster", namespace="$namespace",container!=""}) by (pod))	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_spec_memory_limit_bytes	cadvisor
Memory Request-Total	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name",	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics

	"(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(kube_pod_container_resource_requests_memory_bytes{resource="memory", cluster="$cluster",namespace="$namespace"}) by (pod))		
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Memory Info	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max by(container, pod) (container_memory_working_set_bytes{cluster="$cluster",namespace="$namespace", container!="", image!="", container!="POD"})	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
	max(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max by(container, pod) (kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace="$namespace"}))by(container)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	max(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max by(container, pod) (kube_pod_container_resource_limits_memory_bytes{cluster="$cluster",namespace="$namespace"}))by(container)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics

Memory Usage/Limit (%)	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max by(container, pod) (container_memory_working_set_bytes{cluster="$cluster",namespace="$namespace", container!="", image!="", container!="POD"})/max by(container, pod) (kube_pod_container_resource_limits_memory_bytes{resource="memory", cluster="$cluster",namespace="$namespace"})	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Memory Usage/Request(%)	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max by(container, pod) (container_memory_working_set_bytes{cluster="$cluster",namespace="$namespace", container!="", image!="", container!="POD"})/max by(container, pod) (kube_pod_container_resource_requests_memory_bytes{resource="memory", cluster="$cluster",namespace="$namespace"})	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Sockets	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(container_sockets{cluster="$cluster",namespace="$namespace",container!=""}) by (pod))	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_sockets	cadvisor
Network In	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by	kube_pod_info	kube-state-metrics

	(created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(rate(container_network_receive_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod))	kube_replicaset_owner	kube-state-metrics
		container_network_receive_bytes_total	cadvisor
Network Out	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() sum(rate(container_network_transmit_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod))	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_network_transmit_bytes_total	cadvisor
Network Errors	sum(label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() (sum(container_network_receive_errors_total{cluster="$cluster",namespace="$namespace"}) by (pod) + sum(container_network_transmit_errors_total{cluster="$cluster",namespace="$namespace"}) by (pod)))	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_network_receive_errors_total	cadvisor
		container_network_transmit_errors_total	cadvisor
Network IO	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right()	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_network_receive_bytes_total	cadvisor

	max(rate(container_network_receive_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod) label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(rate(container_network_transmit_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_network_transmit_bytes_total	cadvisor
File System Read	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(rate(container_fs_reads_bytes_total{cluster="$cluster", namespace="$namespace", container!="POD", container!=""}[5m])) by (pod,container)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_fs_reads_bytes_total	cadvisor
File System Write	label_replace(max(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="ReplicaSet", pod_ip!=""}) by (created_by_name, uid, pod, pod_ip, node), "replicaset", "$1", "created_by_name", "(.+)") * on(replicaset) group_left() max(kube_replicaset_owner{cluster="$cluster",namespace="$namespace",owner_kind="Deployment",owner_name="$workload"}) by (replicaset) * on(pod) group_right() max(rate(container_fs_writes_bytes_total{cluster="$cluster", namespace="$namespace", container!="POD", container!=""}[5m])) by (pod,container)	kube_pod_info	kube-state-metrics
		kube_replicaset_owner	kube-state-metrics
		container_fs_writes_bytes_total	cadvisor

StatefulSet

图表名称	查询语句	使用的指标	配置文件
Generation	max(kube_statefulset_metadata_generation{cluster="$cluster",namespace="$namespace", statefulset="$workload"})	kube_statefulset_metadata_generation	kube-state-metrics

Replicas(Pods)-Request	max(kube_statefulset_replicas{statefulset="$workload",cluster="$cluster",namespace="$namespace"})	kube_statefulset_replicas	kube-state-metrics
Replicas(Pods)-Ready	max(kube_statefulset_status_replicas_ready{statefulset="$workload",cluster="$cluster",namespace="$namespace"})	kube_statefulset_status_replicas_ready	kube-state-metrics
Age	time() - max(kube_statefulset_created{cluster="$cluster",namespace="$namespace",statefulset="$workload"})	kube_statefulset_created	kube-state-metrics
Replica Trend	max(kube_statefulset_replicas{statefulset="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_statefulset_replicas	kube-state-metrics
	max(kube_statefulset_status_replicas{statefulset="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_statefulset_status_replicas	kube-state-metrics
	min(kube_statefulset_status_replicas_ready{statefulset="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_statefulset_status_replicas_ready	kube-state-metrics
	min(kube_statefulset_status_replicas_available{statefulset="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_statefulset_status_replicas_available	kube-state-metrics
	min(kube_statefulset_status_replicas_updated{statefulset="$workload",cluster="$cluster",namespace="$namespace"}) without (instance, pod)	kube_statefulset_status_replicas_updated	kube-state-metrics
CPU Usage	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
CPU Quota	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cor	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics

	es{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)		
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod) /sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod) /sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"}	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics

	* on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)		
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
CPU Limit-Total	sum(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(kube_pod_container_resource_limits_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"}) by (pod))	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
CPU Request-Total	sum(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(kube_pod_container_resource_requests_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"}) by (pod))	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU Info	group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container,image)	kube_pod_info	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
	group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster",namespace="$namespace"}) by (pod, container,image)	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
	group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster",namespace="$namespace"}) by (pod, container,image)	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU Usage/Limit (%)	group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container) / max by(container, pod) (kube_pod_container_resource_limits_cpu_cores{	kube_pod_info	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics

CPU Usage/Request(%)	resource="cpu", group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(rate(container_cpu_usage_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container) / max by(container, pod) (kube_pod_container_resource_requests_cpu_cores{resource="cpu", cluster="$cluster",namespace="$namespace"})	kube_pod_info	kube-state-metrics
		container_cpu_usage_seconds_total	cadvisor
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
CPU User Time(%)	avg(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() (max(rate(container_cpu_user_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod, container,image) / max(rate(container_cpu_user_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])+rate(container_cpu_system_seconds_total{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}[5m])) by (pod,container,image))) by (pod,container,image)	kube_pod_info	kube-state-metrics
		container_cpu_user_seconds_total	cadvisor
		container_cpu_user_seconds_total	cadvisor
		container_cpu_system_seconds_total	cadvisor
Memory Usage	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Memory Quota	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics

namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod) /sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	container_memory_working_set_bytes	cadvisor
	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="statefulset"}) by (pod) /sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace",	container_memory_working_set_bytes	cadvisor
	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics

	workload="$workload", workload_type="statefulset"}) by (pod)	namespace_workload_pod :kube_pod_owner:relabel	预聚合指标
Memory Limit-Total	sum(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(container_spec_memory_limit_bytes{cluster="$cluster",namespace="$namespace",container!="",container!="POD"}) by (pod))	kube_pod_info	kube-state-metrics
		container_spec_memory_limit_bytes	cadvisor
Memory Request-Total	sum(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(kube_pod_container_resource_requests_memory_bytes{resource="memory", cluster="$cluster",namespace="$namespace"}) by (pod))	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Memory Info	avg(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max by(container, pod, image) (container_memory_working_set_bytes{cluster="$cluster",namespace="$namespace", container!="", image!="", container!="POD"}))by (container, pod, image)	kube_pod_info	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
	max(avg(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max by(container, pod, image) (kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace="$namespace"}))by (container, pod))by(container)	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	max(avg(group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max by(container, pod) (kube_pod_container_resource_limits_memory_bytes{cluster="$cluster",namespace="$namespace"}))by (container, pod))by(container)	kube_pod_info	kube-state-metrics
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Memory Usage/Limit(%)	group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max by(container, pod) (container_memory_working_set_bytes{cluster="$cluster",namespace="$namespace", container!="", image!="", container!="POD"})/max by(container, pod)	kube_pod_info	kube-state-metrics
		container_memory_working_set_bytes	cadvisor

	(kube_pod_container_resource_limits_memory_bytes{resource="memory", cluster="$cluster",namespace="$namespace"})	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Memory Usage/Request(%)	group(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max by(container, pod) (container_memory_working_set_bytes{cluster="$cluster",namespace="$namespace", container!="", image!="", container!="POD"})/max by(container, pod) (kube_pod_container_resource_requests_memory_bytes{resource="memory", cluster="$cluster",namespace="$namespace"})	kube_pod_info	kube-state-metrics
		container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Sockets	sum(sum(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(container_sockets{cluster="$cluster",namespace="$namespace",container!=""}) by (pod))	kube_pod_info	kube-state-metrics
		container_sockets	cadvisor
Network In	sum(sum(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(rate(container_network_receive_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod))	kube_pod_info	kube-state-metrics
		container_network_receive_bytes_total	cadvisor
Network Out	sum(sum(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() sum(rate(container_network_transmit_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod))	kube_pod_info	kube-state-metrics
		container_network_transmit_bytes_total	cadvisor
Network Errors	sum(sum(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() (sum(container_network_receive_errors_total{cluster="$cluster",namespace="$namespace"}) by (pod) + sum(container_network_transmit_errors_total{cluster="$cluster",namespace="$namespace"}) by (pod)))	kube_pod_info	kube-state-metrics
		container_network_receive_errors_total	cadvisor
		container_network_transmit_errors_total	cadvisor
Network IO	sum(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(rate(container_network_receive_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod)	kube_pod_info	kube-state-metrics
		container_network_receive_bytes_total	cadvisor

	– sum(kube_pod_info{cluster="$cluster",namespace="$namespace",created_by_kind="StatefulSet",pod_ip!="", created_by_name="$workload"}) by (pod) * on(pod) group_right() max(rate(container_network_transmit_bytes_total{cluster="$cluster",namespace="$namespace"}[5m])) by (pod)	kube_pod_info	kube-state-metrics
		container_network_transmit_bytes_total	cadvisor

DaemonSet

图表名称	查询语句	使用的指标	配置文件
CPU Usage	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
CPU Quota	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标

	{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod) /sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", container!="POD", container!=""} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod) /sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Memory Usage	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel	container_memory_working_set_bytes	cadvisor

	{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
Memory Quota	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod) /sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	container_memory_working_set_bytes	cadvisor
		namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics

	{cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", container!="", container!="POD"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod) /sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace"} * on(namespace,pod) group_left(workload, workload_type) namespace_workload_pod:kube_pod_owner:relabel {cluster="$cluster", namespace="$namespace", workload="$workload", workload_type="daemonset"}) by (pod)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
		namespace_workload_pod: kube_pod_owner:relabel	预聚合指标
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
		namespace_workload_pod: kube_pod_owner:relabel	预聚合指标

集群 Pod 监控

图表名称	查询语句	使用的指标	配置文件
Age	time() - max(kube_pod_created{pod=~"$pod",cluster="$cluster",namespace="$namespace"})	kube_pod_created	kube-state-metrics
Restart Count-Last 1 Hour	ceil(sum (increase(kube_pod_container_status_restarts_total{pod=~"$pod",cluster="$cluster",namespace="$namespace"}[1h])))	kube_pod_container_status_restarts_total	kube-state-metrics
Requests-CPU	sum(kube_pod_container_resource_requests_cpu_cores{pod=~"$pod"}) or vector(0)	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
Requests-Memory	sum(kube_pod_container_resource_requests_memory_bytes{pod=~"$pod"}) or vector(0)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
Limits-CPU	sum(kube_pod_container_resource_limits_cpu_cores{pod=~"$pod"}) or vector(0)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
Limits-Memory	sum(kube_pod_container_resource_limits_memory_bytes{pod=~"$pod"}) or vector(0)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Containers	group by (image, container,pod) (kube_pod_container_info{cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_info	kube-state-metrics

	sum by (container,pod) (kube_pod_container_resource_requests_cpu_cores{cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum by (container,pod) (kube_pod_container_resource_requests_memory_bytes{cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	max by (container,pod) (kube_pod_container_status_running{cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_status_running	kube-state-metrics
	sum by (container,pod) (kube_pod_container_resource_limits{resource="cpu", cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_resource_limits	kube-state-metrics
	sum by (container,pod) (kube_pod_container_resource_limits{resource="memory",cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_resource_limits	kube-state-metrics
	max by (container,pod) (kube_pod_container_status_restarts_total{cluster="$cluster",namespace="$namespace", pod=~"$pod"})	kube_pod_container_status_restarts_total	kube-state-metrics
CPU Usage (%)	max(irate(container_cpu_usage_seconds_total{pod=~"$pod",container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}[1m])) by (container,namespace,pod) / max(container_spec_cpu_quota{pod=~"$pod",container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}/100000) by (container,namespace,pod) or on() vector(0)	container_cpu_usage_seconds_total	cadvisor
		container_spec_cpu_quota	cadvisor
CPU Usage By Cores	max(irate(container_cpu_usage_seconds_total{pod=~"$pod",container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}[1m])) by (pod,container,namespace)or on() vector(0)	container_cpu_usage_seconds_total	cadvisor
CPU Load (10s)	max(container_cpu_load_average_10s{namespace=~"$namespace", pod=~"$pod", container!="", container!="POD"} / 1000)by(pod,container)	container_cpu_load_average_10s	cadvisor
CPU Throttled Percent	max (rate (container_cpu_cfs_throttled_seconds_total{image!="", container!="", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (container,pod) / max (rate (container_cpu_cfs_periods_total{image!="", container!="", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (container,pod) or on() vector(0)	container_cpu_cfs_throttled_seconds_total	cadvisor
		container_cpu_cfs_periods_total	cadvisor
CPU Quota	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", pod="$pod", container!="POD", container!=""}) by (container)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
	sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace",	kube_pod_container_resource_requests_cpu	kube-state-metrics

	pod="$pod"}) by (container)	_cores	
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", pod="$pod", container!="POD", container!=""}) by (container) / sum(kube_pod_container_resource_requests_cpu_cores{cluster="$cluster", namespace="$namespace", pod="$pod"}) by (container)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_requests_cpu_cores	kube-state-metrics
	sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace", pod="$pod"}) by (container)	kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
	sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{cluster="$cluster", namespace="$namespace", pod="$pod", container!="POD", container!=""}) by (container) / sum(kube_pod_container_resource_limits_cpu_cores{cluster="$cluster", namespace="$namespace", pod="$pod"}) by (container)	node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate	预聚合指标
		kube_pod_container_resource_limits_cpu_cores	kube-state-metrics
Memory Usage (WSS)	max(container_memory_working_set_bytes{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container)	container_memory_working_set_bytes	cadvisor
Memory Usage	max(container_memory_usage_bytes{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container)	container_memory_usage_bytes	cadvisor
Memory Usage (RSS)	max(container_memory_rss{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container) or on() vector(0)	container_memory_rss	cadvisor
Memory Cache	max(container_memory_cache{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container)	container_memory_cache	cadvisor
Usage WSS/Limit (%)	(max(container_memory_working_set_bytes{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container)/ max(container_spec_memory_limit_bytes{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace, container) * 100) <= 100 or on() vector(0)	container_memory_working_set_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor
Usage/Limit (%)	(max(container_memory_usage_bytes{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container)/ max(container_spec_memory_limit_bytes{pod=~"$pod",container !="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace, container) * 100) <= 100 or on() vector(0)	container_memory_usage_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor

Usage RSS/Limit (%)	(max(container_memory_rss{pod=~"$pod",container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace,container)/ sum(container_spec_memory_limit_bytes{pod=~"$pod",container!="",container!="POD",cluster="$cluster",namespace=~"$namespace"}) by (pod,namespace, container) * 100) <= 100 or on() vector(0)	container_memory_rss	cadvisor
		container_spec_memory_limit_bytes	cadvisor
Memory Failcnt	max (increase(container_memory_failcnt{cluster="$cluster",namespace=~"$namespace", pod=~"$pod", container!=""}[1m])) by (pod,container)	container_memory_failcnt	cadvisor
Memory Quota	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", pod="$pod", container!="POD", container!=""}) by (container)	container_memory_working_set_bytes	cadvisor
	sum(kube_pod_container_resource_requests_memory_bytes{cluster="$cluster", namespace="$namespace", pod="$pod"}) by (container)	kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", pod="$pod", container!="", container!="POD"}) by (container) / sum(kube_pod_container_resource_requests_memory_bytes{namespace="$namespace", pod="$pod"}) by (container)	container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_requests_memory_bytes	kube-state-metrics
	sum(kube_pod_container_resource_limits_memory_bytes{cluster="$cluster", namespace="$namespace", pod="$pod", container!=""}) by (container)	kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
	sum(container_memory_working_set_bytes{cluster="$cluster", namespace="$namespace", pod="$pod", container!="", container!="POD"}) by (container) / sum(kube_pod_container_resource_limits_memory_bytes{namespace="$namespace", pod="$pod"}) by (container)	container_memory_working_set_bytes	cadvisor
		kube_pod_container_resource_limits_memory_bytes	kube-state-metrics
Network Input	max (rate (container_network_receive_bytes_total{image!="",cluster="$cluster",namespace=~"$namespace", pod_name=~"$pod"}[1m])) by(pod)	container_network_receive_bytes_total	cadvisor
Network Output	max (rate (container_network_transmit_bytes_total{image!="",cluster="$cluster",namespace=~"$namespace", pod_name=~"$pod"}[1m]))by(pod)	container_network_transmit_bytes_total	cadvisor
Network Input Error (%)	max (increase (container_network_receive_packets_dropped_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface) / max (increase (container_network_receive_packets_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface)	container_network_receive_packets_dropped_total	cadvisor
		container_network_receive_packets_total	cadvisor
	max (increase (container_network_receive_errors_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface) / max (increase (container_network_receive_packets_total{id!="/",	container_network_receive_errors_total	cadvisor

	cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface)	container_network_receive_packets_total	cadvisor
Network Output Error (%)	max (increase (container_network_transmit_packets_dropped_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface) / max (increase (container_network_transmit_packets_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface)	container_network_transmit_packets_dropped_total	cadvisor
		container_network_transmit_packets_total	cadvisor
	max (increase (container_network_transmit_errors_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface) / max (increase (container_network_receive_packets_total{id!="/", cluster="$cluster",namespace=~"$namespace", pod=~"$pod"}[1m])) by (pod,interface)	container_network_transmit_errors_total	cadvisor
		container_network_receive_packets_total	cadvisor
File System Read	max (rate(container_fs_reads_bytes_total{cluster="$cluster",namespace=~"$namespace", pod=~"$pod", container!=""}[1m]))by (container,pod)	container_fs_reads_bytes_total	cadvisor
File System Write	max (rate(container_fs_writes_bytes_total{cluster="$cluster",namespace=~"$namespace", pod=~"$pod", container!=""}[1m])) by (container,pod)	container_fs_writes_bytes_total	cadvisor
Network Socket	max(container_sockets{cluster="$cluster",namespace=~"$namespace", pod=~"$pod", container!=""}) by (container,pod)	container_sockets	cadvisor
Process Number	count(container_processes{cluster="$cluster",namespace=~"$namespace", pod=~"$pod", container!=""}) by (container,pod)	container_processes	cadvisor

集群网络监控

图表名称	查询语句	使用的指标	配置文件
Current Rate of Bytes Received	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_bytes_total	cadvisor
Current Rate of Bytes Transmitted	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_bytes_total	cadvisor
Current Status	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_bytes_total	cadvisor
	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_bytes_total	cadvisor
	sort_desc(avg(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_bytes_total	cadvisor

	sort_desc(avg(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_bytes_total	cadvisor
	sort_desc(sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_packets_total	cadvisor
	sort_desc(sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_packets_total	cadvisor
	sort_desc(sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_packets_total	cadvisor
	sort_desc(sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_packets_dropped_total	cadvisor
Average Rate of Bytes Received	sort_desc(avg(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_bytes_total	cadvisor
Average Rate of Bytes Transmitted	sort_desc(avg(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_bytes_total	cadvisor
Receive Bandwidth	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_bytes_total	cadvisor
Transmit Bandwidth	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_bytes_total	cadvisor
Rate of Received Packets	sort_desc(sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_packets_total	cadvisor
Rate of Transmitted Packets	sort_desc(sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_packets_total	cadvisor
Rate of Received Packets Dropped	sort_desc(sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_receive_packets_dropped_total	cadvisor
Rate of Transmitted Packets Dropped	sort_desc(sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~".+"}[5m])) by (namespace))	container_network_transmit_packets_dropped_total	cadvisor
Rate of TCP Retransmits out of all sent segments	sort_desc(sum(rate(node_netstat_Tcp_RetransSegs{cluster=~"$cluster"}[5m]) / rate(node_netstat_Tcp_OutSegs{cluster=~"$cluster"}[$interval:$resolution])) by (instance))	node_netstat_Tcp_RetransSegs	node-exporter
Rate of TCP SYN Retransmits out of all retransmits	sort_desc(sum(rate(node_netstat_TcpExt_TCPSynRetrans{cluster=~"$cluster"}[$interval:$resolution]) / rate(node_netstat_Tcp_RetransSegs{cluster=~"$cluster"}[$interval:$resolution])) by (instance))	node_netstat_TcpExt_TCPSynRetrans	node-exporter
		node_netstat_Tcp_RetransSegs	node-exporter

命名空间 Pods 网络监控

图表名称	查询语句	使用的指标	配置文件
Current Rate of Bytes Received	sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]))	container_network_receive_bytes_total	cadvisor
Current Rate of Bytes Transmitted	sum(irate(container_network_transmit_bytes_total{namespace=~"$namespace"}[5m]))	container_network_transmit_bytes_total	cadvisor
Current Status	sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_receive_bytes_total	cadvisor
	sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_transmit_bytes_total	cadvisor
	sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_receive_packets_total	cadvisor
	sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_transmit_packets_total	cadvisor
	sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_receive_packets_dropped_total	cadvisor
	sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_transmit_packets_dropped_total	cadvisor
Receive Bandwidth	sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_receive_bytes_total	cadvisor
Transmit Bandwidth	sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_transmit_bytes_total	cadvisor
Rate of Received Packets	sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_receive_packets_total	cadvisor
Rate of Transmitted Packets	sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_transmit_packets_total	cadvisor
Rate of Received Packets Dropped	sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_receive_packets_dropped_total	cadvisor
Rate of Transmitted Packets Dropped	sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) by (pod)	container_network_transmit_packets_dropped_total	cadvisor

命名空间工作负载网络监控

图表名称	查询语句	使用的指标	配置文件
Current Rate of Bytes Received	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Current Rate of Bytes Transmitted	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Current Status	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sort_desc(avg(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sort_desc(avg(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标

	sort_desc(sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_packets_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sort_desc(sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_packets_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sort_desc(sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_packets_dropped_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
	sort_desc(sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_packets_dropped_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Average Rate of Bytes Received	sort_desc(avg(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Average Rate of Bytes Transmitted	sort_desc(avg(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Receive Bandwidth	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])	container_network_receive_bytes_total	cadvisor

	* on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))		
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Transmit Bandwidth	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Received Packets	sort_desc(sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_packets_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Transmitted Packets	sort_desc(sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_packets_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Received Packets Dropped	sort_desc(sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_receive_packets_dropped_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Transmitted Packets Dropped	sort_desc(sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~".+", workload_type="$type"}) by (workload))	container_network_transmit_packets_dropped_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标

Pod 网络监控

图表名称	查询语句	使用的指标	配置文件
Current Rate of Bytes Received	sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace",	container_network_receive_bytes_total	cadvisor

	pod=~"$pod"}[5m]))		
Current Rate of Bytes Transmitted	sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m]))	container_network_transmit_bytes_total	cadvisor
Receive Bandwidth	sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m])) by (pod)	container_network_receive_bytes_total	cadvisor
Transmit Bandwidth	sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m])) by (pod)	container_network_transmit_bytes_total	cadvisor
Rate of Received Packets	sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m])) by (pod)	container_network_receive_packets_total	cadvisor
Rate of Transmitted Packets	sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m])) by (pod)	container_network_transmit_packets_total	cadvisor
Rate of Received Packets Dropped	sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m])) by (pod)	container_network_receive_packets_dropped_total	cadvisor
Rate of Transmitted Packets Dropped	sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace", pod=~"$pod"}[5m])) by (pod)	container_network_transmit_packets_dropped_total	cadvisor

工作负载网络监控

图表名称	查询语句	使用的指标	配置文件
Current Rate of Bytes Received	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_transmit_bytes_total	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	cadvisor
Current Rate of Bytes Transmitted	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Average Rate of Bytes Received	sort_desc(avg(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_receive_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标

Average Rate of Bytes Transmitted	sort_desc(avg(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Receive Bandwidth	sort_desc(sum(irate(container_network_receive_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_receive_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Transmit Bandwidth	sort_desc(sum(irate(container_network_transmit_bytes_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_transmit_bytes_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Received Packets	sort_desc(sum(irate(container_network_receive_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_receive_packets_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Transmitted Packets	sort_desc(sum(irate(container_network_transmit_packets_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_transmit_packets_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标
Rate of Received Packets Dropped	sort_desc(sum(irate(container_network_receive_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m]) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_receive_packets_dropped_total	预聚合指标
		namespace_workload_pod:kube_pod_owner:relabel	cadvisor

Rate of Transmitted Packets Dropped	sort_desc(sum(irate(container_network_transmit_packets_dropped_total{cluster=~"$cluster",namespace=~"$namespace"}[5m])) * on (namespace,pod) group_left(workload,workload_type) namespace_workload_pod:kube_pod_owner:relabel{cluster=~"$cluster",namespace=~"$namespace", workload=~"$workload", workload_type="$type"}) by (pod))	container_network_transmit_packets_dropped_total	cadvisor
		namespace_workload_pod:kube_pod_owner:relabel	预聚合指标

PVC 存储监控

图表名称	查询语句	使用的指标	配置文件
Volume Space Usage	(sum without(instance, node) (kubelet_volume_stats_capacity_bytes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"}) – sum without(instance, node) (kubelet_volume_stats_available_bytes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"}))	kubelet_volume_stats_capacity_bytes	kubelet
		kubelet_volume_stats_available_bytes	kubelet
	sum without(instance, node) (kubelet_volume_stats_available_bytes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"})	kubelet_volume_stats_available_bytes	kubelet
Volume Space Usage	(kubelet_volume_stats_capacity_bytes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"} – kubelet_volume_stats_available_bytes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"}) / kubelet_volume_stats_capacity_bytes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"} * 100	kubelet_volume_stats_capacity_bytes	kubelet
		kubelet_volume_stats_available_bytes	kubelet
		kubelet_volume_stats_capacity_bytes	kubelet
Volume inodes Usage	sum without(instance, node) (kubelet_volume_stats_inodes_used{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"})	kubelet_volume_stats_inodes_used	kubelet
	(sum without(instance, node) (kubelet_volume_stats_inodes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"}) – sum without(instance, node) (kubelet_volume_stats_inodes_used{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"}))	kubelet_volume_stats_inodes	kubelet
		kubelet_volume_stats_inodes_used	kubelet

Volume inodes Usage	kubelet_volume_stats_inodes_used{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"} / kubelet_volume_stats_inodes{cluster="$cluster", job="kubelet", namespace="$namespace", persistentvolumeclaim="$volume"} * 100	kubelet_volume_stats_inodes_used	kubelet
		kubelet_volume_stats_inodes	kubelet

Controller Manager（托管集群）

图表名称	查询语句	使用的指标	配置文件
Memory Usage	pod_mem_usage{cluster="$cluster", job="kube-controller-manager",instance=~"$instance"}	pod_mem_usage	gpu 监控
CPU Usage	pod_core_usage{cluster="$cluster", job="kube-controller-manager",instance=~"$instance"}	pod_core_usage	gpu 监控
In Traffic	sum(rate(container_network_receive_bytes_total{cluster="$cluster", job="kube-controller-manager",instance=~"$instance"}[5m])) by(instance)	container_network_receive_bytes_total	cadvisor
Out Traffic	sum(rate(container_network_transmit_bytes_total{cluster="$cluster", job="kube-controller-manager",instance=~"$instance"}[5m])) by(instance)	container_network_transmit_bytes_total	cadvisor
Work Queue Depth	sum(rate(workqueue_depth{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance"}[5m])) by (instance, name)	workqueue_depth	kubelet
Work Queue Add Rate	sum(rate(workqueue_adds_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance"}[5m])) by (instance, name)	workqueue_adds_total	kube-controller-manager
Work Queue Latency	histogram_quantile(0.99, sum(rate(workqueue_queue_duration_seconds_bucket{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance"}[5m])) by (instance, name, le))	workqueue_queue_duration_seconds_bucket	kube-controller-manager
Kube API Request Rate	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-controller-manager
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kube-controller-manager
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-controller-manager
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance",code=~"5.."}[5m]))	rest_client_requests_total	kube-controller-manager

Post Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance", verb="POST"}[5m])) by (verb, le))	rest_client_request_duration_seconds_bucket	kube-controller-manager
Get Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-controller-manager", instance=~"$instance", verb="GET"}[5m])) by (verb, le))	rest_client_request_duration_seconds_bucket	kube-controller-manager

Scheduler（托管集群）

图表名称	查询语句	使用的指标	配置文件
Memory Usage	pod_mem_usage{cluster="$cluster", job="kube-scheduler", instance=~"$instance"}	pod_mem_usage	gpu 监控
CPU Usage	pod_core_usage{cluster="$cluster", job="kube-scheduler", instance=~"$instance"}	pod_core_usage	gpu 监控
In Traffic	sum(rate(container_network_receive_bytes_total{job="kube-scheduler", cluster="$cluster",instance=~"$instance"}[5m])) by (instance)	container_network_receive_bytes_total	cadvisor
Out Traffic	sum(rate(container_network_transmit_bytes_total{job="kube-scheduler", cluster="$cluster",instance=~"$instance"}[5m])) by (instance)	container_network_transmit_bytes_total	cadvisor
Pending Pods	sum(scheduler_pending_pods{job="kube-scheduler", cluster="$cluster", instance=~"$instance"}) by (queue)	scheduler_pending_pods	kube-scheduler
Number of attempts to successfully schedule(P90)	histogram_quantile(0.9, sum(rate(scheduler_pod_scheduling_attempts_bucket{job="kube-scheduler", cluster="$cluster", instance=~"$instance"}[5m])) by (le,instance))	scheduler_pod_scheduling_attempts_bucket	kube-scheduler
Get Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance", verb="GET"}[5m])) by (verb, le))	rest_client_request_duration_seconds_bucket	kube-scheduler
Post Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance", verb="POST"}[5m])) by (verb, le))	rest_client_request_duration_seconds_bucket	kube-scheduler
Kube API Request Rate	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-scheduler
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"3.."}[5m]))	rest_client_requests_total	kube-scheduler
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-scheduler
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"5.."}[5m]))	rest_client_requests_total	kube-scheduler

API Server（托管集群）

图表名称	查询语句	使用的指标	配置文件
Memory Usage	pod_mem_usage{cluster="$cluster", job="kube-scheduler", instance=~"$instance"}	pod_mem_usage	gpu 监控
CPU Usage	pod_core_usage{cluster="$cluster", job="kube-scheduler", instance=~"$instance"}	pod_core_usage	gpu 监控
In Traffic	sum(rate(container_network_receive_bytes_total{job="kube-scheduler", cluster="$cluster",instance=~"$instance"}[5m])) by (instance)	container_network_receive_bytes_total	cadvisor
Out Traffic	sum(rate(container_network_transmit_bytes_total{job="kube-scheduler", cluster="$cluster",instance=~"$instance"}[5m])) by (instance)	container_network_transmit_bytes_total	cadvisor
Pending Pods	sum(scheduler_pending_pods{job="kube-scheduler", cluster="$cluster", instance=~"$instance"}) by (queue)	scheduler_pending_pods	kube-scheduler
		scheduler_pod_scheduling_attempts_bucket	kube-scheduler
Get Request Latency 99th Quantile	histogram_quantile(0.99, sum(rate(rest_client_request_duration_seconds_bucket{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance", verb="GET"}[5m])) by (verb, le))	rest_client_request_duration_seconds_bucket	kube-scheduler
		rest_client_request_duration_seconds_bucket	kube-scheduler
Kube API Request Rate	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"2.."}[5m]))	rest_client_requests_total	kube-scheduler
		rest_client_requests_total	kube-scheduler
	sum(rate(rest_client_requests_total{cluster=~"$cluster",job="kube-scheduler", instance=~"$instance",code=~"4.."}[5m]))	rest_client_requests_total	kube-scheduler
		rest_client_requests_total	kube-scheduler
Write Request/s	sum(rate(apiserver_request_count{cluster="$cluster", verb=~"POST\|PUT\|PATCH\|DELETE",instance=~"$instance"}[5m])) by (instance) or sum(rate(apiserver_request_total{cluster="$cluster", verb=~"POST\|PUT\|PATCH\|DELETE",instance=~"$instance"}[5m])) by (instance)	apiserver_request_count	kube-apiserver
		apiserver_request_total	kube-apiserver
Read Request/s	sum(rate(apiserver_request_count{cluster="$cluster", verb=~"LIST\|GET",instance=~"$instance"}[5m])) by (instance) or sum(rate(apiserver_request_total{cluster="$cluster", verb=~"LIST\|GET",instance=~"$instance"}[5m])) by (instance)	apiserver_request_count	kube-apiserver
		apiserver_request_total	kube-apiserver
Latency(Average)	(sum(rate(apiserver_request_duration_seconds_sum{cluster="$cluster", verb=~"GET\|LIST\|POST\|PUT\|DELETE\|PATCH", instance=~"$instance"}[5m])) by(instance))	apiserver_request_duration_seconds_sum	kube-apiserver

	/(sum(rate(apiserver_request_duration_seconds_count{cluster="$cluster", verb=~"GET\|LIST\|POST\|PUT\|DELETE\|PATCH", instance=~"$instance"}[5m])) by(instance))	apiserver_request_duration_seconds_count	kube-apiserver
Latency(P99)	histogram_quantile(0.99, sum(rate(apiserver_request_duration_seconds_bucket{verb="LIST", cluster="$cluster",instance=~"$instance"}[5m])) by (le,instance))	apiserver_request_duration_seconds_bucket	kube-apiserver
Current Inflight Request	sum(apiserver_current_inflight_requests{cluster="$cluster",instance=~"$instance"}) by (instance)	apiserver_current_inflight_requests	kube-apiserver
Self Request/s	sum by (instance) (rate(apiserver_selfrequest_total{cluster="$cluster",instance=~"$instance"}[5m]))	apiserver_selfrequest_total	kube-apiserver
Response Body Size(P99)	histogram_quantile(0.99, sum by (instance,le) (rate(apiserver_response_sizes_bucket{cluster="$cluster",instance=~"$instance"}[5m])))	apiserver_response_sizes_bucket	kube-apiserver
Watch Events/s	sum by (instance) (rate(apiserver_watch_events_total{cluster="$cluster",instance=~"$instance"}[5m]))	apiserver_watch_events_total	kube-apiserver
Too Many Objects Events/s	sum(rate(list_too_many_objects_events_total{cluster="$cluster",instance=~"$instance"}[5m])) by (instance)	list_too_many_objects_events_total	kube-apiserver
Too Old Objects Events/s	sum(rate(watch_too_old_objects_events_total{cluster="$cluster",instance=~"$instance"}[5m])) by (instance)	watch_too_old_objects_events_total	kube-apiserver
Read Request/s	sum(rate(apiserver_request_count{cluster="$cluster", verb=~"GET\|LIST",instance=~"$instance",resource!=""}[5m])) by (resource) > 0 or sum(rate(apiserver_request_total{cluster="$cluster", verb=~"GET\|LIST",instance=~"$instance",resource!=""}[5m])) by (resource) > 0	apiserver_request_count	kube-apiserver
		apiserver_request_total	kube-apiserver
Write Request/s	sum(rate(apiserver_request_count{cluster="$cluster", verb=~"POST\|PUT\|PATCH\|DELETE",instance=~"$instance"}[5m])) by (resource) > 0 or sum(rate(apiserver_request_total{cluster="$cluster", verb=~"POST\|PUT\|PATCH\|DELETE",instance=~"$instance"}[5m])) by (resource) > 0	apiserver_request_count	kube-apiserver
		apiserver_request_total	kube-apiserver
Latency(Average)	(sum(rate(apiserver_request_duration_seconds_sum{cluster="$cluster", verb=~"GET\|LIST\|POST\|PUT\|DELETE\|PATCH", instance=~"$instance"}[5m])) by(resource)) /(sum(rate(apiserver_request_duration_seconds_count{cluster="$cluster", verb=~"GET\|LIST\|POST\|PUT\|DELETE\|PATCH", instance=~"$instance"}[5m])) by(resource)) >0	apiserver_request_duration_seconds_sum	kube-apiserver
		apiserver_request_duration_seconds_count	kube-apiserver
Latency(P99)	histogram_quantile(0.99, sum(rate(apiserver_request_duration_seconds_bucket{verb=~"GET|LIST|POST|PUT|DELETE|PATCH", cluster="$cluster",instance=~"$instance"}[5m])) by (le,resource))>0	apiserver_request_duration_seconds_bucket	kube-apiserver

GPU Cluster（GPU 集成）

图表名称	查询语句	使用的指标	配置文件

Total GPU Nodes	count(count by(Hostname) (DCGM_FI_DEV_COUNT{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}))	DCGM_FI_DEV_COUNT	TKE GPU Exporter 集成
Allocated GPUs	count(DCGM_FI_DEV_COUNT{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"})	DCGM_FI_DEV_COUNT	TKE GPU Exporter 集成
	sum(max_over_time(gpu_core_allocated{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}[1m]))/100	gpu_core_allocated	TKE GPU Exporter 集成
Allocated GPU Memory Ratio	sum(gpu_mem_allocated{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}) / sum(gpu_mem_allocatable{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}) *100	gpu_mem_allocated	TKE GPU Exporter 集成
		gpu_mem_allocatable	TKE GPU Exporter 集成
Used GPU Memory Ratio	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) / sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) * 100	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
		DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成
Average GPU Utilization	avg(max_over_time(DCGM_FI_DEV_GPU_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m]))	DCGM_FI_DEV_GPU_UTIL	TKE GPU Exporter 集成
GPU Memory Copy Utilization	avg(max_over_time(DCGM_FI_DEV_MEM_COPY_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m]))	DCGM_FI_DEV_MEM_COPY_UTIL	TKE GPU Exporter 集成
The Last one XID Error	sum(DCGM_FI_DEV_XID_ERRORS{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"})	DCGM_FI_DEV_XID_ERRORS	TKE GPU Exporter 集成
GPU Node Details	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,gpu)	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_GPU_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,gpu)	DCGM_FI_DEV_GPU_UTIL	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_POWER_USAGE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,gpu)	DCGM_FI_DEV_POWER_USAGE	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_GPU_TEMP{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,gpu)	DCGM_FI_DEV_GPU_TEMP	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_MEM_COPY_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,gpu)	DCGM_FI_DEV_MEM_COPY_UTIL	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,gpu)	DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成

GPU Node（GPU 集成）

图表名称	查询语句	使用的指标	配置文件
GPU Utilization	avg(max_over_time(DCGM_FI_DEV_GPU_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m]))	DCGM_FI_DEV_GPU_UTIL	TKE GPU Exporter 集成
Allocated GPU Memory	sum(max_over_time(gpu_mem_allocated{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}[1m]))/ sum(gpu_mem_allocatable{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}) *100	gpu_mem_allocated	TKE GPU Exporter 集成
		gpu_mem_allocatable	TKE GPU Exporter 集成
Used GPU Memory	avg(gpu_mem_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"})	gpu_mem_utilization_percentage	TKE GPU Exporter 集成
Allocated GPUs	sum(max_over_time(gpu_core_allocated{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}[1m]))/100	gpu_core_allocated	TKE GPU Exporter 集成
	count(DCGM_FI_DEV_COUNT{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"})	DCGM_FI_DEV_COUNT	TKE GPU Exporter 集成
Allocated Computing Power(Valid in GPU Sharing)	sum(gpu_power_usage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}) by (node)	gpu_power_usage	TKE GPU Exporter 集成
The Last one XID Error	DCGM_FI_DEV_XID_ERRORS{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_XID_ERRORS	TKE GPU Exporter 集成
GPU Utilization	DCGM_FI_DEV_GPU_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_GPU_UTIL	TKE GPU Exporter 集成
GPU Memory Copy Utilization	DCGM_FI_DEV_MEM_COPY_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_MEM_COPY_UTIL	TKE GPU Exporter 集成
Encoder Engine Utilization	DCGM_FI_DEV_ENC_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_ENC_UTIL	TKE GPU Exporter 集成
Decoder Engine Utilization	DCGM_FI_DEV_DEC_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_DEC_UTIL	TKE GPU Exporter 集成
GPU Memory Details	sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by (Hostname,UUID,gpu,modelName)	DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m]))by(Hostname,UUID,gpu,modelName)	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m]))by (Hostname,UUID,gpu,modelName) / sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成

	~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])) by(Hostname,UUID,gpu,modelName) * 100	DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成
BAR1 Used	DCGM_FI_DEV_BAR1_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_BAR1_USED	TKE GPU Exporter 集成
BAR1 Total	DCGM_FI_DEV_BAR1_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_BAR1_TOTAL	TKE GPU Exporter 集成
GPU Memory Used	DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
Graphics Engine Active(%)	DCGM_FI_PROF_GR_ENGINE_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 100	DCGM_FI_PROF_GR_ENGINE_ACTIVE	TKE GPU Exporter 集成
DRAM Active(%)	DCGM_FI_PROF_DRAM_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 100	DCGM_FI_PROF_DRAM_ACTIVE	TKE GPU Exporter 集成
SM Active(%)	DCGM_FI_PROF_SM_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}* 100	DCGM_FI_PROF_SM_ACTIVE	TKE GPU Exporter 集成
SM Occupancy(%)	DCGM_FI_PROF_SM_OCCUPANCY{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_PROF_SM_OCCUPANCY	TKE GPU Exporter 集成
Tensor Core Engine Active(%)	DCGM_FI_PROF_PIPE_TENSOR_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 100	DCGM_FI_PROF_PIPE_TENSOR_ACTIVE	TKE GPU Exporter 集成
FP32 Engine Active(%)	DCGM_FI_PROF_PIPE_FP32_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 100	DCGM_FI_PROF_PIPE_FP32_ACTIVE	TKE GPU Exporter 集成
FP16 Engine Active(%)	DCGM_FI_PROF_PIPE_FP16_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 100	DCGM_FI_PROF_PIPE_FP16_ACTIVE	TKE GPU Exporter 集成
FP64 Engine Active(%)	DCGM_FI_PROF_PIPE_FP64_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 100	DCGM_FI_PROF_PIPE_FP64_ACTIVE	TKE GPU Exporter 集成
PCIE TX Bytes(Device to Host)	rate(DCGM_FI_PROF_PCIE_TX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])	DCGM_FI_PROF_PCIE_TX_BYTES	TKE GPU Exporter 集成
PCIE RX Bytes(Host to Device)	rate(DCGM_FI_PROF_PCIE_RX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])	DCGM_FI_PROF_PCIE_RX_BYTES	TKE GPU Exporter 集成
NVLINK TX Bytes	rate(DCGM_FI_PROF_NVLINK_TX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])	DCGM_FI_PROF_NVLINK_TX_BYTES	TKE GPU Exporter 集成
NVLINK RX Bytes	rate(DCGM_FI_PROF_NVLINK_RX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}[1m])	DCGM_FI_PROF_NVLINK_RX_BYTES	TKE GPU Exporter 集成

Power Usage(W)	DCGM_FI_DEV_POWER_USAGE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_POWER_USAGE	TKE GPU Exporter 集成
Toal Energy Consumption(in J)	DCGM_FI_DEV_TOTAL_ENERGY_CONSUMPTION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} / 1000	DCGM_FI_DEV_TOTAL_ENERGY_CONSUMPTION	TKE GPU Exporter 集成
Memory Temperature(℃)	sum(gpu_temprature{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode"}) by (node,card)	gpu_temprature	TKE GPU Exporter 集成
GPU Temperature(℃)	DCGM_FI_DEV_GPU_TEMP{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_GPU_TEMP	TKE GPU Exporter 集成
SM Clock	DCGM_FI_DEV_SM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 1000* 1000	DCGM_FI_DEV_SM_CLOCK	TKE GPU Exporter 集成
Memory Clock	DCGM_FI_DEV_MEM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 1000 * 1000	DCGM_FI_DEV_MEM_CLOCK	TKE GPU Exporter 集成
APP SM Clock	DCGM_FI_DEV_APP_SM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 1000 * 1000	DCGM_FI_DEV_APP_SM_CLOCK	TKE GPU Exporter 集成
APP Memory Clock	DCGM_FI_DEV_APP_MEM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 1000 * 1000	DCGM_FI_DEV_APP_MEM_CLOCK	TKE GPU Exporter 集成
Video Clock	DCGM_FI_DEV_VIDEO_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"} * 1000 * 1000	DCGM_FI_DEV_VIDEO_CLOCK	TKE GPU Exporter 集成
Clock Throttle Reasons	DCGM_FI_DEV_CLOCK_THROTTLE_REASONS{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_CLOCK_THROTTLE_REASONS	TKE GPU Exporter 集成
Retired Pages(Single-bit Errors)	DCGM_FI_DEV_RETIRED_SBE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_RETIRED_SBE	TKE GPU Exporter 集成
Retired Pages(Double-bit Errors)	DCGM_FI_DEV_RETIRED_DBE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_RETIRED_DBE	TKE GPU Exporter 集成
Power Violation	DCGM_FI_DEV_POWER_VIOLATION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_POWER_VIOLATION	TKE GPU Exporter 集成
Thermal Violation	DCGM_FI_DEV_THERMAL_VIOLATION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_THERMAL_VIOLATION	TKE GPU Exporter 集成
Sync Boost Violation	DCGM_FI_DEV_SYNC_BOOST_VIOLATION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_SYNC_BOOST_VIOLATION	TKE GPU Exporter 集成
Board Relability Violation	DCGM_FI_DEV_RELIABILITY_VIOLATION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_RELIABILITY_VIOLATION	TKE GPU Exporter 集成

Board Limit Violation	DCGM_FI_DEV_BOARD_LIMIT_VIOLATION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_BOARD_LIMIT_VIOLATION	TKE GPU Exporter 集成
Low Util Violation	DCGM_FI_DEV_LOW_UTIL_VIOLATION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode"}	DCGM_FI_DEV_LOW_UTIL_VIOLATION	TKE GPU Exporter 集成

GPU Pod（GPU 集成）

图表名称	查询语句	使用的指标	配置文件
Memory Usage	avg(sum(container_memory_rss{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod, namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_memory_rss	cadvisor
		kube_pod_owner	kube-state-metrics
	avg(sum(container_memory_working_set_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod, namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_memory_working_set_bytes	cadvisor
		kube_pod_owner	kube-state-metrics
	sum(container_memory_rss{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod,namespace)	container_memory_rss	cadvisor
	sum(container_memory_working_set_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod,namespace)	container_memory_working_set_bytes	cadvisor
Memory Percent	avg((sum(container_memory_rss{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod,namespace)/ sum(container_spec_memory_limit_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",namespace=~"$PodNamespace"}) by (pod,namespace) * 100) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_memory_rss	cadvisor
		container_spec_memory_limit_bytes	cadvisor
		kube_pod_owner	kube-state-metrics
	avg((sum(container_memory_working_set_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName", container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod,namespace)/ sum(container_spec_memory_limit_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container	container_memory_working_set_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor

	!="",container!="POD",namespace=~"$PodNamespace"}) by (pod,namespace) * 100) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	kube_pod_owner	kube-state-metrics
	(sum(container_memory_rss{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod,namespace)/ sum(container_spec_memory_limit_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",namespace=~"$PodNamespace"}) by (pod,namespace) * 100)	container_memory_rss	cadvisor
		container_spec_memory_limit_bytes	cadvisor
	(sum(container_memory_working_set_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}) by (pod,namespace)/ sum(container_spec_memory_limit_bytes{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",namespace=~"$PodNamespace"}) by (pod,namespace) * 100)	container_memory_working_set_bytes	cadvisor
		container_spec_memory_limit_bytes	cadvisor
CPU Usage By Cores	avg(sum(irate(container_cpu_usage_seconds_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m])) by (pod,namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_cpu_usage_seconds_total	cadvisor
		kube_pod_owner	kube-state-metrics
	sum(irate(container_cpu_usage_seconds_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m])) by (pod,namespace)	container_cpu_usage_seconds_total	cadvisor
CPU Usage Percent	avg(sum(irate(container_cpu_usage_seconds_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m])) by (pod,namespace) / sum(container_spec_cpu_quota{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container!="",container!="POD",instance=~"$GPUNode",namespace=~"$PodNamespace"}/100000) by (pod,namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_cpu_usage_seconds_total	cadvisor
		container_spec_cpu_quota	cadvisor
		kube_pod_owner	kube-state-metrics
	sum(irate(container_cpu_usage_seconds_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m])) by	container_cpu_usage_seconds_total	cadvisor

	(pod,namespace) / sum(container_spec_cpu_quota{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container!="",container!="POD",instance=~"$GPUNode",namespace=~"$PodNamespace"}/100000) by (pod,namespace)	container_spec_cpu_quota	cadvisor
Network Bandwidth Usage	avg(sum(rate(container_network_receive_bytes_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m])) by (pod, namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",namespace=~"$PodNamespace", pod=~"$PodName"})) by (namespace,owner_name)	container_network_receive_bytes_total	cadvisor
		kube_pod_owner	kube-state-metrics
	avg(sum(rate(container_network_transmit_bytes_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m])) by (pod, namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",namespace=~"$PodNamespace", pod=~"$PodName"})) by (namespace,owner_name)	container_network_transmit_bytes_total	cadvisor
		kube_pod_owner	kube-state-metrics
	sum (rate (container_network_receive_bytes_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m]))by (pod,namespace)	container_network_receive_bytes_total	cadvisor
	sum (rate (container_network_transmit_bytes_total{cluster=~"$cluster",nodepool=~"$NodePool",pod=~"$PodName",container !="",container!="POD",instance=~"$GPUNode", namespace=~"$PodNamespace"}[1m]))by (pod,namespace)	container_network_transmit_bytes_total	cadvisor
Network Socket	avg(sum(container_sockets{cluster=~"$cluster",nodepool=~"$NodePool",namespace=~"$PodNamespace", pod=~"$PodName", instance=~"$GPUNode", container!=""}) by (pod,namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_sockets	cadvisor
		kube_pod_owner	kube-state-metrics
	sum(container_sockets{cluster=~"$cluster",nodepool=~"$NodePool",namespace=~"$PodNamespace", pod=~"$PodName", instance=~"$GPUNode", container!=""}) by (pod,namespace)	container_sockets	cadvisor
File System	avg(sum (rate(container_fs_reads_bytes_total{cluster=~"$cluster", nodepool=~"$NodePool",namespace=~"$PodNamespace", pod=~"$PodName", instance=~"$GPUNode", container!=""}[1m]))by (pod,namespace) * on(pod,	container_fs_reads_bytes_total	cadvisor

	namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	kube_pod_owner	kube-state-metrics
	avg(sum (rate(container_fs_writes_bytes_total{cluster=~"$cluster",nodepool=~"$NodePool",namespace=~"$PodNamespace", pod=~"$PodName", instance=~"$GPUNode", container!=""}[1m])) by (pod,namespace) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	container_fs_writes_bytes_total	cadvisor
	sum (rate(container_fs_reads_bytes_total{cluster=~"$cluster", nodepool=~"$NodePool",namespace=~"$PodNamespace", pod=~"$PodName", instance=~"$GPUNode", container!=""}[1m]))by (pod,namespace)	container_fs_reads_bytes_total	cadvisor
	sum (rate(container_fs_writes_bytes_total{cluster=~"$cluster",nodepool=~"$NodePool",namespace=~"$PodNamespace", pod=~"$PodName", instance=~"$GPUNode", container!=""}[1m])) by (pod,namespace)	container_fs_writes_bytes_total	cadvisor
Pods Average SM Utilization	avg(avg(pod_core_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",namespace=~"$PodNamespace", pod=~"$PodName"}) by (namespace, pod)* on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName", namespace=~"$PodNamespace"})) by (namespace,owner_name)	pod_core_utilization_percentage	TKE GPU Exporter 集成
		kube_pod_owner	kube-state-metrics
	avg(pod_core_utilization_percentage{cluster=~"$cluster", nodepool=~"$NodePool",node=~"$GPUNode",namespace=~"$PodNamespace", pod=~"$PodName"}) by (namespace, pod)	pod_core_utilization_percentage	TKE GPU Exporter 集成
Pods GPU Memory Used Percentage	avg(sum(pod_mem_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",pod=~"$PodName", namespace=~"$PodNamespace"}) by (pod, namespace,job) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName", namespace=~"$PodNamespace"})) by (namespace,owner_name)	pod_mem_utilization_percentage	TKE GPU Exporter 集成
		kube_pod_owner	kube-state-metrics
	sum(pod_mem_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",pod=~"$PodName", namespace=~"$PodNamespace"}) by (pod, namespace,job)	pod_mem_utilization_percentage	TKE GPU Exporter 集成
Pods Used GPU Memory	avg(sum(pod_mem_usage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",pod=~"$PodName",namespace=~"$PodNamespace"}) by (pod, namespace) * on(pod, namespace) group_left(owner_name) avg by	pod_mem_usage	TKE GPU Exporter 集成

	(owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	kube_pod_owner	kube-state-metrics
	sum(pod_mem_usage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",pod=~"$PodName",namespace=~"$PodNamespace"}) by (pod, namespace)	pod_mem_usage	TKE GPU Exporter 集成
Pods GPU Encode Utilization	avg(avg(pod_enc_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",namespace=~"$PodNamespace",pod=~"$PodName"}) by (namespace, pod) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	pod_enc_utilization_percentage	TKE GPU Exporter 集成
		kube_pod_owner	kube-state-metrics
	avg(pod_enc_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",namespace=~"$PodNamespace",pod=~"$PodName"}) by (namespace, pod)	pod_enc_utilization_percentage	TKE GPU Exporter 集成
Pods GPU Decode Utilization	avg(avg(pod_dec_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",namespace=~"$PodNamespace",pod=~"$PodName"}) by (namespace, pod) * on(pod, namespace) group_left(owner_name) avg by (owner_name, pod, namespace) (kube_pod_owner{cluster=~"$cluster",pod=~"$PodName",namespace=~"$PodNamespace"})) by (namespace,owner_name)	pod_dec_utilization_percentage	TKE GPU Exporter 集成
		kube_pod_owner	kube-state-metrics
	avg(pod_dec_utilization_percentage{cluster=~"$cluster",nodepool=~"$NodePool",node=~"$GPUNode",namespace=~"$PodNamespace",pod=~"$PodName"}) by (namespace, pod)	pod_dec_utilization_percentage	TKE GPU Exporter 集成
GPU Utilization	DCGM_FI_DEV_GPU_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_GPU_UTIL	TKE GPU Exporter 集成
GPU Memory Copy Utilization	DCGM_FI_DEV_MEM_COPY_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_MEM_COPY_UTIL	TKE GPU Exporter 集成
Encoder Engine Utilization	DCGM_FI_DEV_ENC_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_ENC_UTIL	TKE GPU Exporter 集成
Decoder Engine Utilization	DCGM_FI_DEV_DEC_UTIL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_DEC_UTIL	TKE GPU Exporter 集成
GPU Memory Details	sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])) by (UUID,Hostname,modelName)	DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成
	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m]))by (UUID,Hostname,modelName)	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成

	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m]))by (UUID,Hostname,modelName) / sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])) by (UUID,Hostname,modelName) * 100	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
		DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成
GPU Memory Used	DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
GPU Memory Used Percentage	sum(max_over_time(DCGM_FI_DEV_FB_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m]))by (Hostname,UUID) / sum(max_over_time(DCGM_FI_DEV_FB_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])) by (Hostname,UUID) * 100	DCGM_FI_DEV_FB_USED	TKE GPU Exporter 集成
		DCGM_FI_DEV_FB_TOTAL	TKE GPU Exporter 集成
BAR1 Used	DCGM_FI_DEV_BAR1_USED{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_BAR1_USED	TKE GPU Exporter 集成
BAR1 Total	DCGM_FI_DEV_BAR1_TOTAL{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_BAR1_TOTAL	TKE GPU Exporter 集成
Graphics Engine Active(%)	DCGM_FI_PROF_GR_ENGINE_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 100	DCGM_FI_PROF_GR_ENGINE_ACTIVE	TKE GPU Exporter 集成
DRAM Active(%)	DCGM_FI_PROF_DRAM_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 100	DCGM_FI_PROF_DRAM_ACTIVE	TKE GPU Exporter 集成
SM Active(%)	DCGM_FI_PROF_SM_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}* 100	DCGM_FI_PROF_SM_ACTIVE	TKE GPU Exporter 集成
SM Occupancy(%)	DCGM_FI_PROF_SM_OCCUPANCY{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_PROF_SM_OCCUPANCY	TKE GPU Exporter 集成
Tensor Core Engine Active(%)	DCGM_FI_PROF_PIPE_TENSOR_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 100	DCGM_FI_PROF_PIPE_TENSOR_ACTIVE	TKE GPU Exporter 集成
FP32 Engine Active(%)	DCGM_FI_PROF_PIPE_FP32_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 100	DCGM_FI_PROF_PIPE_FP32_ACTIVE	TKE GPU Exporter 集成
FP16 Engine Active(%)	DCGM_FI_PROF_PIPE_FP16_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 100	DCGM_FI_PROF_PIPE_FP16_ACTIVE	TKE GPU Exporter 集成
FP64 Engine Active(%)	DCGM_FI_PROF_PIPE_FP64_ACTIVE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 100	DCGM_FI_PROF_PIPE_FP64_ACTIVE	TKE GPU Exporter 集成
PCIE TX Bytes(Device to Host)	rate(DCGM_FI_PROF_PCIE_TX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])	DCGM_FI_PROF_PCIE_TX_BYTES	TKE GPU Exporter 集成

PCIE RX Bytes(Host to Device)	rate(DCGM_FI_PROF_PCIE_RX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])	DCGM_FI_PROF_PCIE_RX_BYTES	TKE GPU Exporter 集成
NVLINK TX Bytes	rate(DCGM_FI_PROF_NVLINK_TX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])	DCGM_FI_PROF_NVLINK_TX_BYTES	TKE GPU Exporter 集成
NVLINK RX Bytes	rate(DCGM_FI_PROF_NVLINK_RX_BYTES{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}[1m])	DCGM_FI_PROF_NVLINK_RX_BYTES	TKE GPU Exporter 集成
Power Usage(W)	DCGM_FI_DEV_POWER_USAGE{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_POWER_USAGE	TKE GPU Exporter 集成
Toal Energy Consumption(in J)	DCGM_FI_DEV_TOTAL_ENERGY_CONSUMPTION{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} / 1000	DCGM_FI_DEV_TOTAL_ENERGY_CONSUMPTION	TKE GPU Exporter 集成
GPU Temperature(℃)	DCGM_FI_DEV_GPU_TEMP{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_GPU_TEMP	TKE GPU Exporter 集成
SM Clock	DCGM_FI_DEV_SM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 1000* 1000	DCGM_FI_DEV_SM_CLOCK	TKE GPU Exporter 集成
Memory Clock	DCGM_FI_DEV_MEM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 1000 * 1000	DCGM_FI_DEV_MEM_CLOCK	TKE GPU Exporter 集成
APP SM Clock	DCGM_FI_DEV_APP_SM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 1000 * 1000	DCGM_FI_DEV_APP_SM_CLOCK	TKE GPU Exporter 集成
APP Memory Clock	DCGM_FI_DEV_APP_MEM_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 1000 * 1000	DCGM_FI_DEV_APP_MEM_CLOCK	TKE GPU Exporter 集成
Video Clock	DCGM_FI_DEV_VIDEO_CLOCK{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"} * 1000 * 1000	DCGM_FI_DEV_VIDEO_CLOCK	TKE GPU Exporter 集成
Clock Throttle Reasons	DCGM_FI_DEV_CLOCK_THROTTLE_REASONS{cluster=~"$cluster",nodepool=~"$NodePool",Hostname=~"$GPUNode",UUID=~"$GPU"}	DCGM_FI_DEV_CLOCK_THROTTLE_REASONS	TKE GPU Exporter 集成

数据采集配置

最近更新时间：2024-12-05 16:07:34

操作场景

本文档介绍如何为已完成关联的集群配置监控采集项。

前提条件

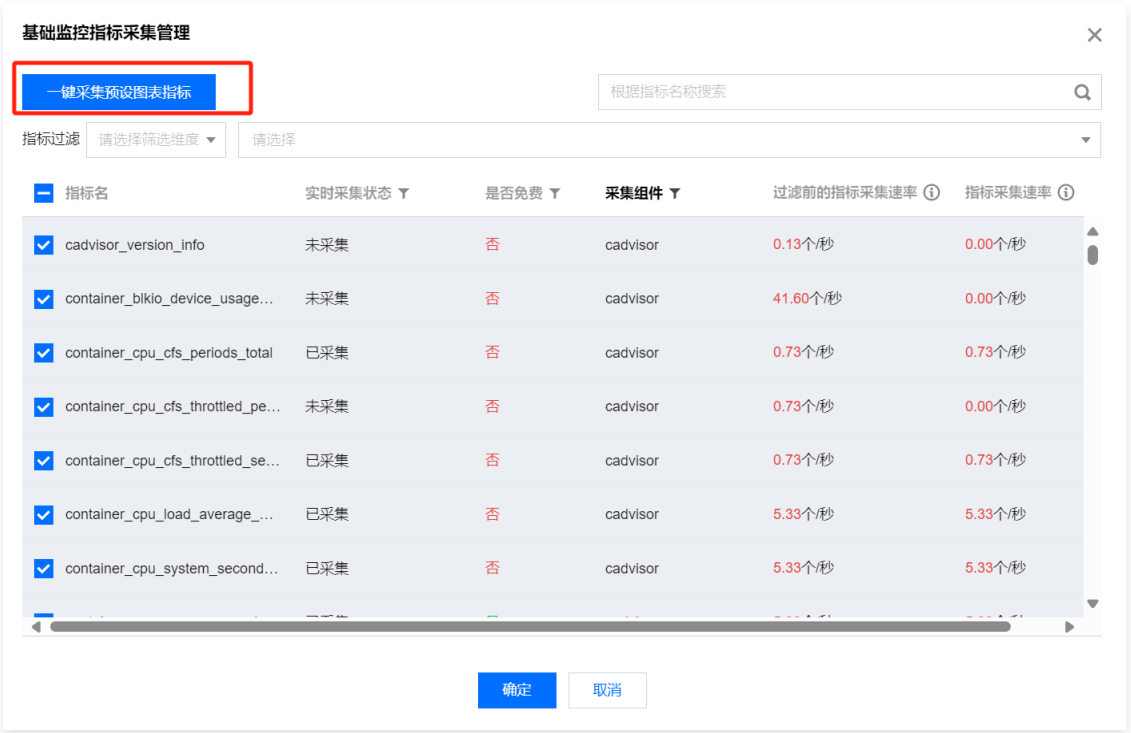
在配置监控数据采集项前，您需要完成以下操作：

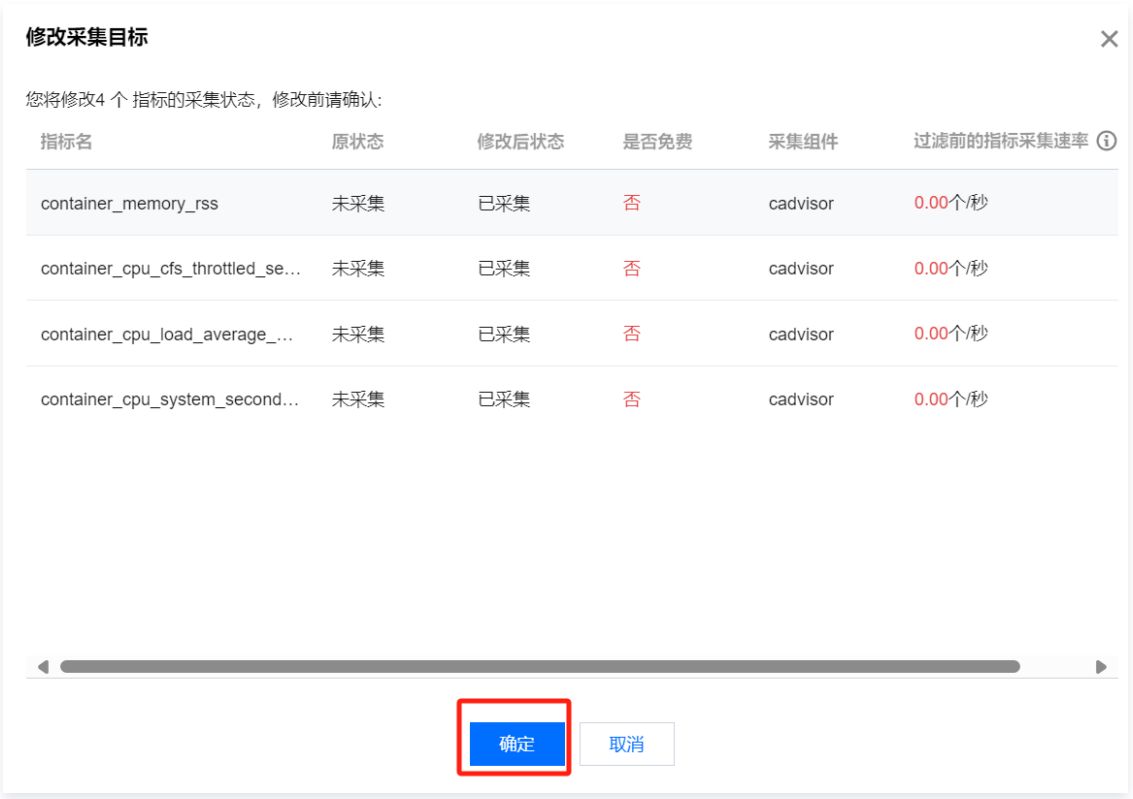
- 已成功创建 Prometheus 监控实例。
- 已将需要监控的集群关联到相应实例中。

操作步骤

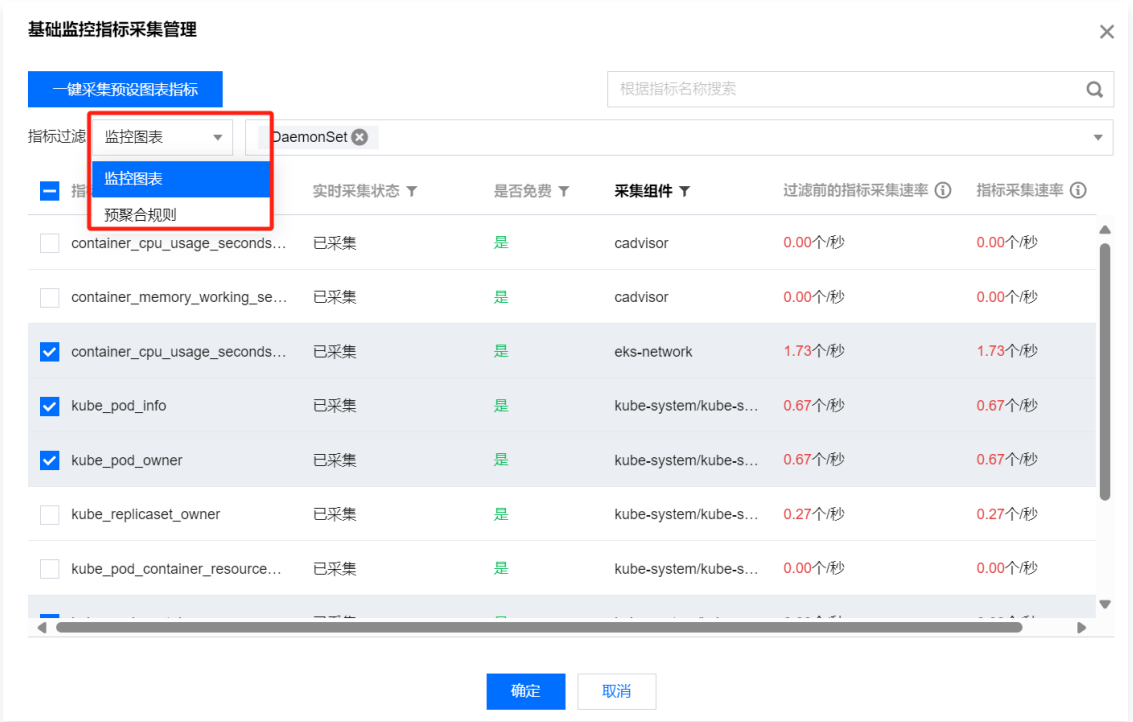
配置数据采集

1. 登录 腾讯云可观测平台，选择左侧导航栏中的 **Prometheus 监控**。
2. 在监控实例列表页，选择需要配置数据采集规则的实例名称，进入该实例详情页。
3. 在**数据采集 > 集成容器服务**页面，单击实例右侧的**数据采集配置**，进入采集配置列表页。
4. 点击页面上方的**采集指标管理**，会弹出基础监控指标采集管理页面。
   - 若需要采集所有容器图表指标，可以在弹出的窗口中点击**一键采集预设图表指标**，在弹出的确认窗口点击**确定**，将会采集所有预设图表指标。





- 若只需要采集部分指标，可以通过**监控图表**和**预聚合规则**进行指标筛选，勾选自己需要的指标后再点击**确定**即可。如下图所示：



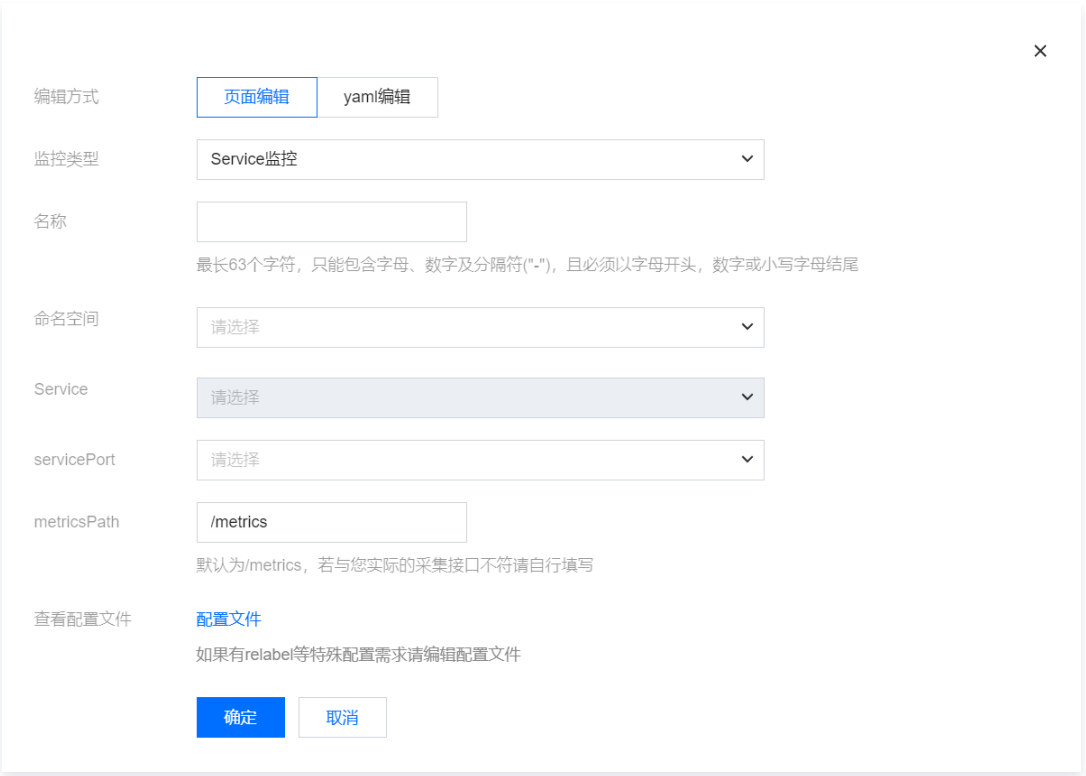
5. 在“数据采集配置”页中，单击**新建自定义监控**，新增数据采集配置。Prometheus 监控服务预置了部分采集配置文件，用于采集常规的监控数据。您可以通过以下两种方式配置新的数据采集规则，以监控您的业务数据。

通过控制台新增配置

监控 Service

1. 单击**页面编辑**。

2. 在“新建采集配置”弹窗中，填写配置信息。如下图所示：

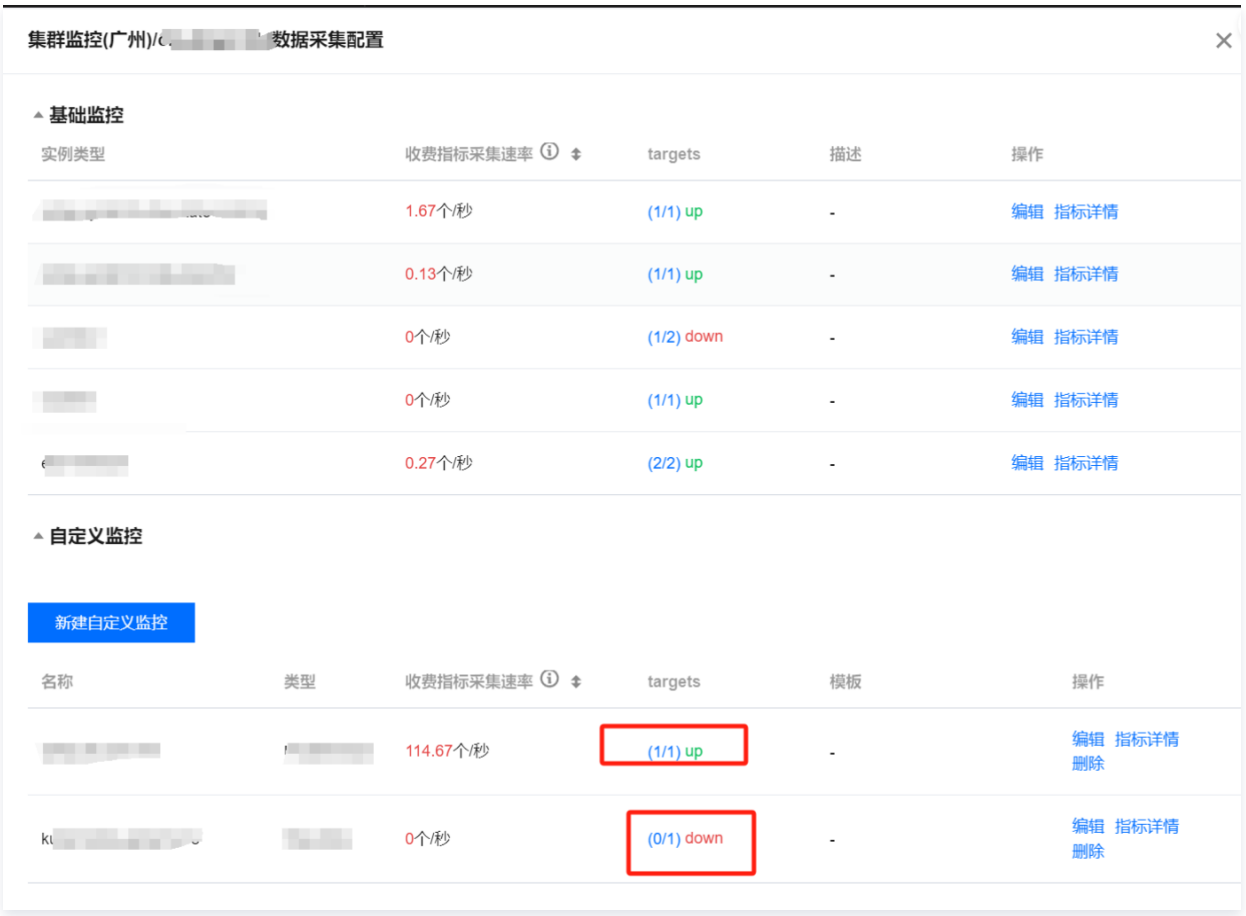


通过 yaml 文件新增配置

1. 单击 **yaml 编辑**。
2. 在弹窗中，选择监控类型，并填写相应配置。
   您可以按照社区的使用方式通过提交相应的 yaml 来完成数据采集的配置。
   - **工作负载监控**：对应配置为 PodMonitors。
   - **service 监控**：对应配置为 ServiceMonitors。
   - **RawJobs 监控**：对应配置为 RawJobs。

6. 单击**确定**完成配置。

7. 在该实例的“数据采集配置”页中，查看采集目标状态。如下图所示：

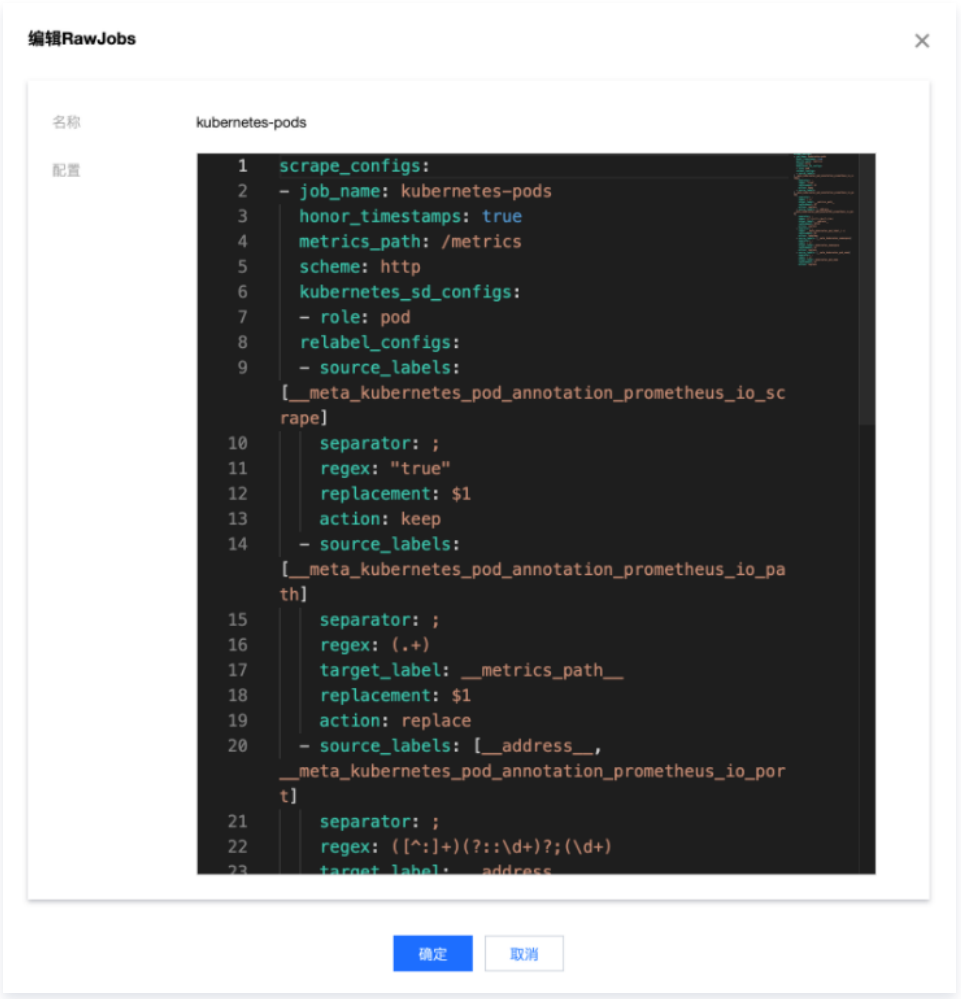


targets（1/1）表示（实际抓取的 targets 数为1 / 探测的采集目标数为1）。当实际抓取数和探测数的数值相等时，显示为 **up**，即表示当前抓取正常。
当实际抓取数小于探测数时，显示为 **down**，即表示有部分 endpoints 抓取失败。
单击上图中的字段值（**0/1**）即可查看采集目标的详细信息。如下图所示 down 的失败状态：

查看已有配置

1. 登录 腾讯云可观测平台，选择左侧导航栏中的 **Prometheus 监控**。
2. 在监控实例列表页，选择需要配置数据采集规则的实例名称，进入该实例详情页。
3. 在**数据采集 > 集成容器服务**页面，单击实例右侧的**数据采集配置**，进入采集配置列表页。选择**基础监控**或者**自定义监控**，单击右侧的**编辑**。

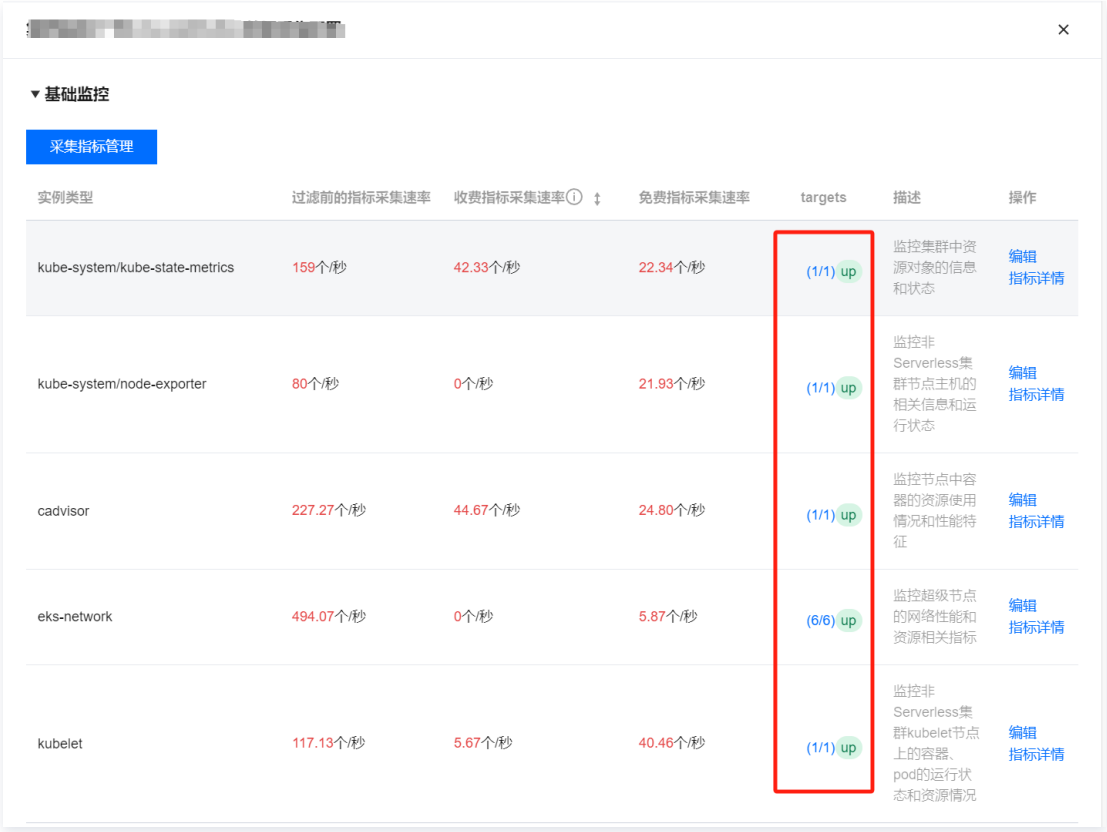
4. 在弹出的窗口查看 yaml 文件中当前配置的所有监控对象。如下图所示：



查看采集目标

1. 登录 腾讯云可观测平台，选择左侧导航栏中的 **Prometheus 监控服务**。
2. 在监控实例列表页，选择需要查看 Targets 的实例名称，进入该实例详情页。
3. 在**数据采集 > 集成容器服务**页面，单击实例右侧的**数据采集配置**。

4. 在弹出的窗口中点击 targets 下方的显示状态，即可跳转到当前数据的详情页。



相关操作

挂载文件到采集器

在配置采集项的时候，如果您需要为配置提供一些文件，例如证书，可以通过以下方式向采集器挂载文件，文件的更新会实时同步到采集器内。

- **prometheus.tke.tencent.cloud.com/scrape-mount = "true"**
  prom-xxx 命名空间下的 configmap 添加如上 label，其中所有的 key 会被挂载到采集器的路径 `/etc/prometheus/configmaps/[configmap-name]/` 。
- **prometheus.tke.tencent.cloud.com/scrape-mount = "true"**
  prom-xxx 命名空间下的 secret 添加如上 label，其中所有的 key 会被挂载到采集器的路径 `/etc/prometheus/secrets/[secret-name]/` 。

精简监控指标

最近更新时间：2024-12-05 16:07:34

本文档介绍如何精简 Prometheus 监控服务的**采集指标**，避免不必要的费用支出。

前提条件

在配置监控数据采集项前，您需要完成以下操作：

- 已成功 创建 Prometheus 监控实例 。
- 已将需要 监控的集群关联到相应实例 中。

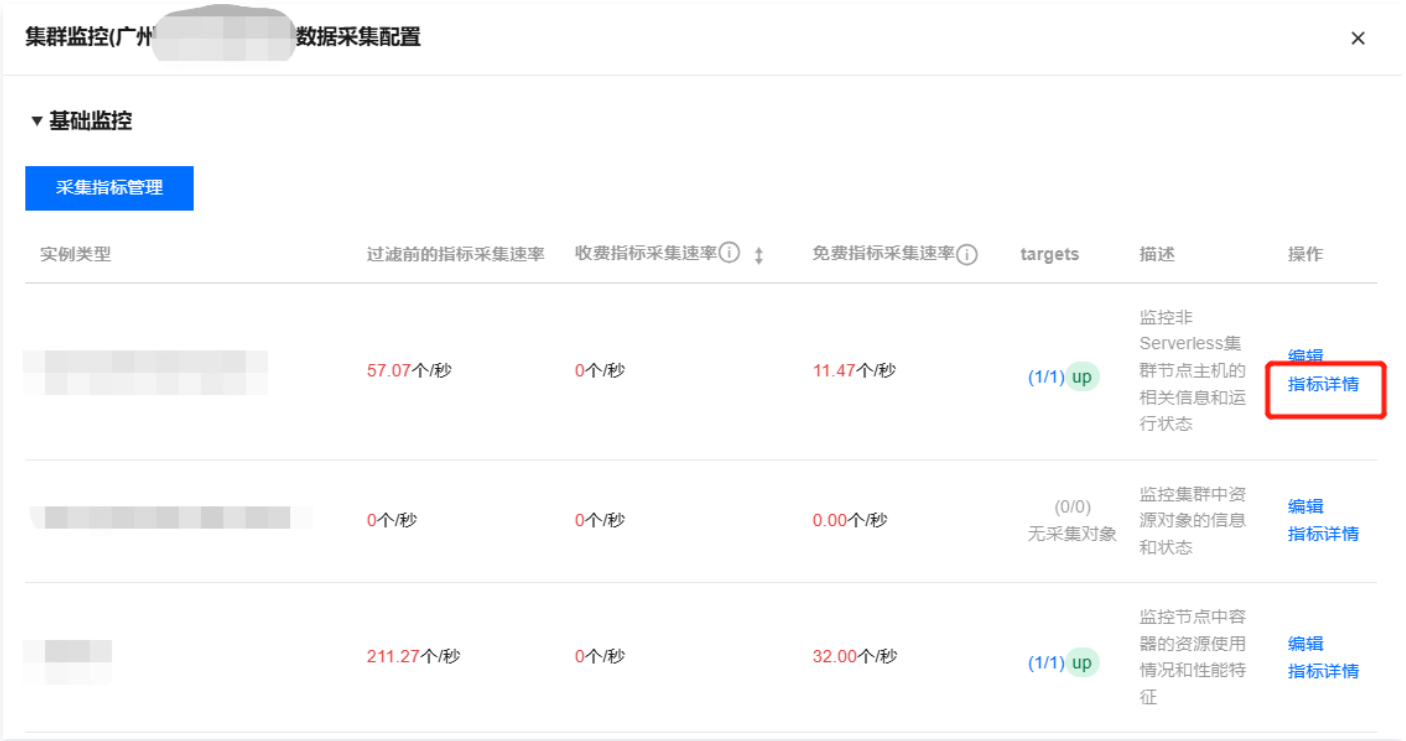
精简指标

通过控制台精简指标

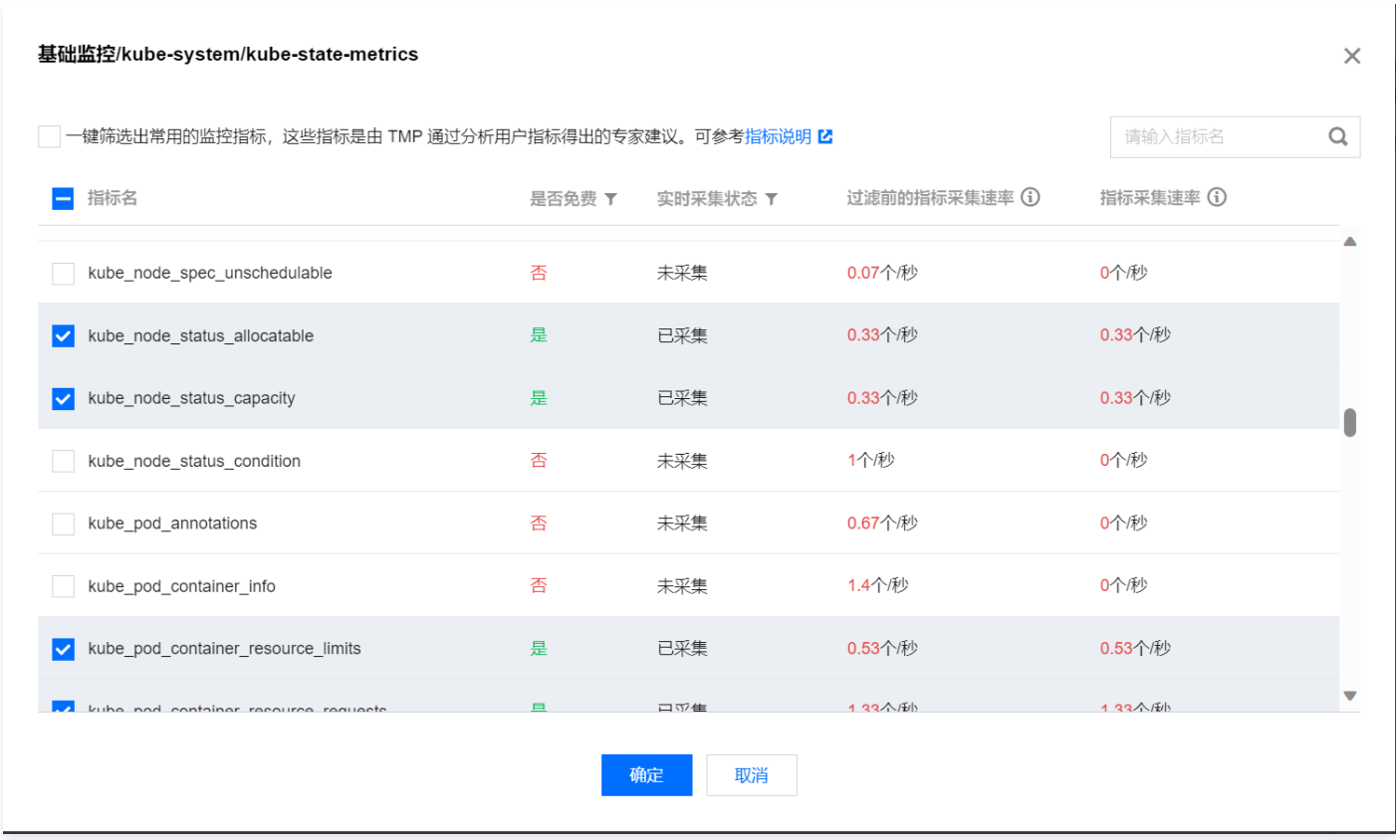
Prometheus 监控服务提供了一百多个免费的基础监控指标，完整的指标列表可查看 按量付费免费指标 。

1. 登录 腾讯云可观测平台控制台 ，选择左侧导航栏中的 Prometheus 监控 。
2. 在监控实例列表页，选择需要配置数据采集规则的实例名称，进入该实例详情页。
3. 在**数据采集**页面，选择**集成容器服务**，单击集群右侧的**数据采集配置**，如下图所示：

4. 进入采集配置列表页后，基础指标支持通过产品化的页面增加/减少采集对象，单击右侧的**指标详情**，如下图所示：



5. 在以下页面您可以查看到每个指标是否免费，指标勾选表示会采集这些指标，您可按需勾选使用。仅基础监控提供免费的监控指标，完整的免费指标详情请参见 按量付费免费指标 。付费指标计算详情见 Prometheus 监控服务按量计费 。



通过 YAML 精简指标

Prometheus 目前收费模式为按监控数据的点数收费，为了最大程度减少不必要的浪费，建议您针对采集配置进行优化，只采集需要的指标，过滤掉非必要指标，从而减少整体上报量。详细的计费方式和相关云资源的使用请查看 文档 。
以下步骤将分别介绍如何在自定义指标的 ServiceMonitor、PodMonitor，以及原生 Job 中加入过滤配置，精简自定义指标。

1. 登录 腾讯云可观测平台控制台，选择左侧导航栏中的 Prometheus 监控 。
2. 在监控实例列表页，选择需要配置数据采集规则的实例名称，进入该实例详情页。
3. 在**数据采集**页面，选择**集成容器服务**，单击集群右侧的**数据采集配置**。
4. 单击实例右侧的**编辑**查看指标详情。

ServiceMonitor 和 PodMonitor

ServiceMonitor 和 PodMonitor 的过滤配置字段相同，本文以 ServiceMonitor 为例。
ServiceMonitor 示例：

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/version: 1.9.7
  name: kube-state-metrics
  namespace: kube-system
spec:
  endpoints:
  - bearerTokenSecret:
      key: ""
    interval: 15s # 该参数为采集频率，您可以调大以降低数据存储费用，例如不重要的指标可以改为 300s，可以降低20倍的监控数据采集量
    port: http-metrics
```

```
    scrapeTimeout: 15s # 该参数为采集超时时间，Prometheus 的配置要求采集超时时间不能超过采集间隔，即：
scrapeTimeout <= interval
  jobLabel: app.kubernetes.io/name
  namespaceSelector: {}
  selector:
    matchLabels:
      app.kubernetes.io/name: kube-state-metrics
```

若要采集 `kube_node_info` 和 `kube_node_role` 的指标，则需要在 ServiceMonitor 的 endpoints 列表中，加入 `metricRelabelings` 字段配置。注意：是 **`metricRelabelings`** 而不是 `relabelings` 。
添加 `metricRelabelings` 示例：

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app.kubernetes.io/name: kube-state-metrics
    app.kubernetes.io/version: 1.9.7
  name: kube-state-metrics
  namespace: kube-system
spec:
  endpoints:
  - bearerTokenSecret:
      key: ""
    interval: 15s # 该参数为采集频率，您可以调大以降低数据存储费用，例如不重要的指标可以改为 300s，可以降低20倍的
监控数据采集量
    port: http-metrics
    scrapeTimeout: 15s
    # 加了如下四行：
    metricRelabelings: # 针对每个采集到的点都会做如下处理
    - sourceLabels: ["__name__"] # 要检测的label名称，__name__ 表示指标名称，也可以是任意这个点所带的label
      regex: kube_node_info|kube_node_role # 上述label是否满足这个正则，在这里，我们希望__name__满足
kube_node_info或kube_node_role
      action:  keep # 如果点满足上述条件，则保留，否则就自动抛弃
  jobLabel: app.kubernetes.io/name
  namespaceSelector: {}
  selector:
```

原生 Job

如果使用的是 Prometheus 原生的 Job，则可以参考以下方式进行指标过滤。
Job 示例：

```
scrape_configs:
  - job_name: job1
    scrape_interval: 15s # 该参数为采集频率，您可以调大以降低数据存储费用，例如不重要的指标可以改为 300s，可以降
低20倍的监控数据采集量
    static_configs:
      - targets:
          - '1.1.1.1'
```

若只需采集 `kube_node_info` 和 `kube_node_role` 的指标，则需要加入 `metric_relabel_configs` 配置。注意：是 **`metric_relabel_configs`** 而不是 `relabel_configs` 。
添加 `metric_relabel_configs` 示例：

```
scrape_configs:
  - job_name: job1
    scrape_interval: 15s # 该参数为采集频率，您可以调大以降低数据存储费用，例如不重要的指标可以改为 300s，可以降低20倍的监控数据采集量
    static_configs:
    - targets:
      - '1.1.1.1'
    # 加了如下四行：
    metric_relabel_configs: # 针对每个采集到的点都会做如下处理
    - source_labels: ["__name__"] # 要检测的label名称，__name__ 表示指标名称，也可以是任意这个点所带的label
      regex: kube_node_info|kube_node_role # 上述label是否满足这个正则，在这里，我们希望__name__满足kube_node_info或kube_node_role
      action: keep # 如果点满足上述条件，则保留，否则就自动抛弃
```

5. 单击**确定**。

屏蔽部分采集对象

屏蔽整个命名空间的监控

Prometheus 关联集群后，默认会纳管集群中所有 ServiceMonitor 和 PodMonitor，若您想屏蔽某个命名空间下的监控，可以为指定命名空间添加 label：`tps-skip-monitor: "true"`，关于 label 的操作请 参考 。

屏蔽部分采集对象

Prometheus 通过在用户的集群里面创建 ServiceMonitor 和 PodMonitor 类型的 CRD 资源进行监控数据的采集，若您想屏蔽指定 ServiceMonitor 和 PodMonitor 的采集，可以为这些 CRD 资源添加 label：`tps-skip-monitor: "true"`，关于 label 的操作请 参考 。

调整采集间隔

最近更新时间：2025-06-17 15:50:53

本文档介绍如何调整 Prometheus 监控服务的**采集间隔**，避免不必要的费用支出。

前提条件

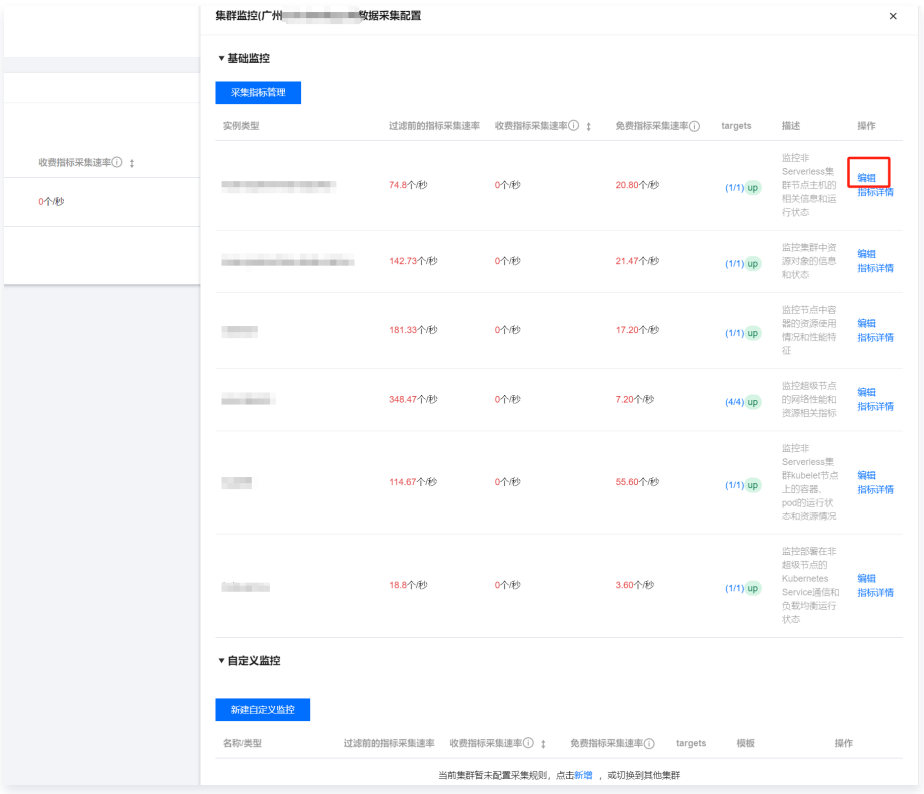
在配置监控数据采集项前，您需要完成以下操作：

- 已成功 创建 Prometheus 监控实例 。
- 已将需要 监控的集群关联到相应实例 中。

通过集成容器服务的控制台调整采集间隔

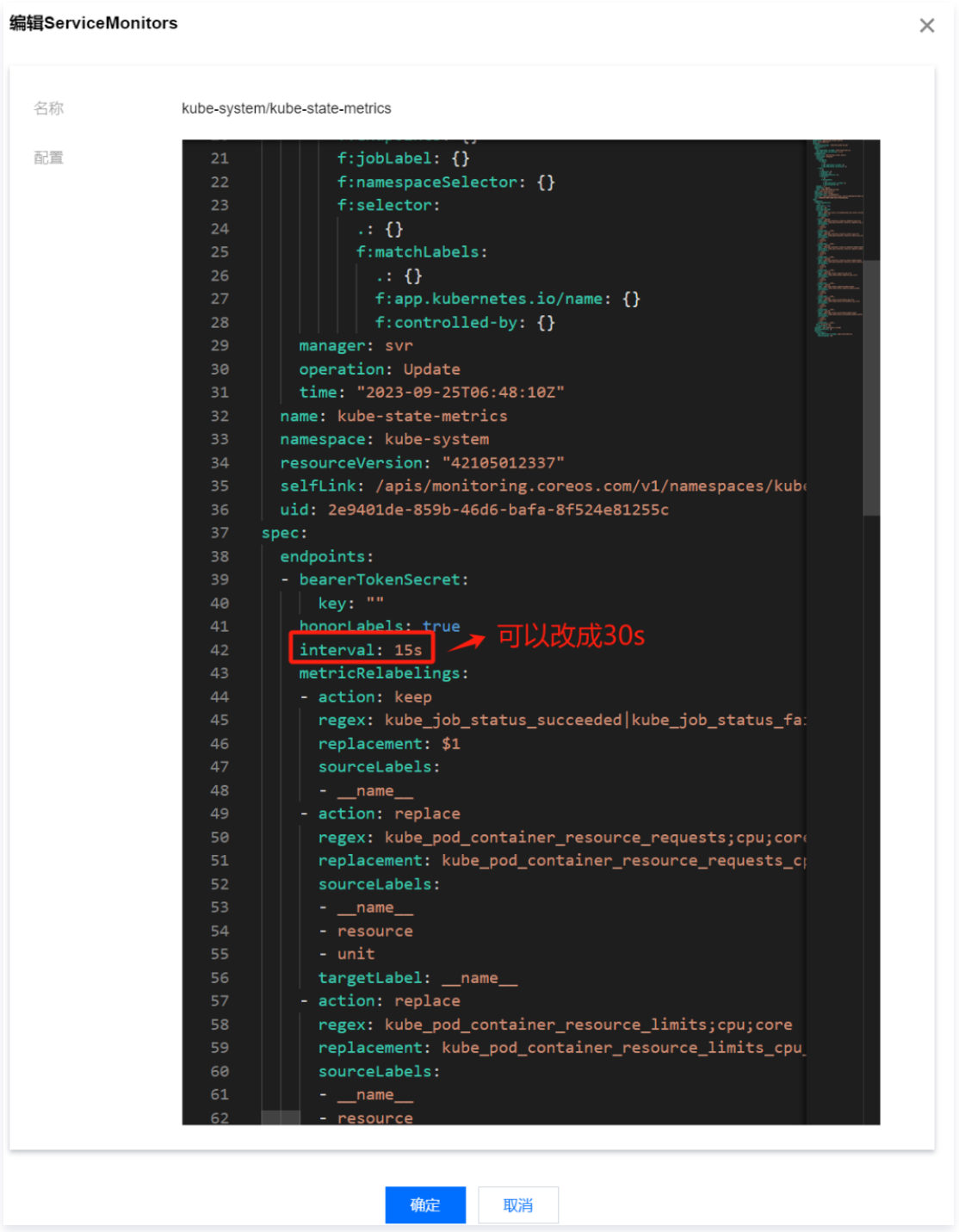
1. 登录 腾讯云可观测平台控制台 ，选择左侧导航栏中的 Prometheus 监控 。
2. 在监控实例列表页，选择需要配置数据采集规则的实例名称，进入该实例详情页。
3. 在**数据采集**页面，选择**集成容器服务**，单击集群右侧的**数据采集配置**， 如下图所示：

4. 进入采集配置列表页后，单击右侧的**编辑**，如下图所示：

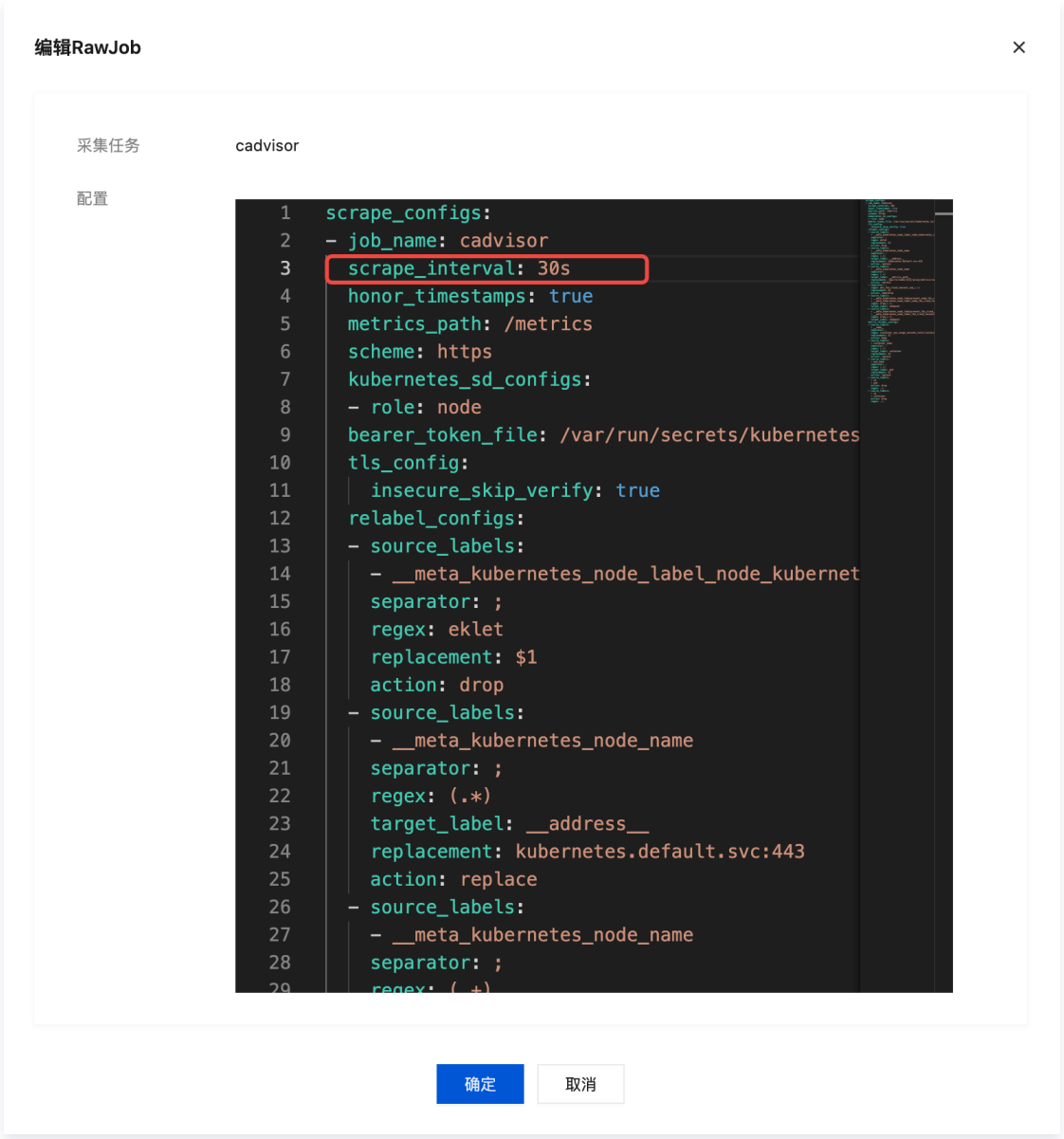


5. 采集任务配置包含 PodMonitor、ServiceMonitor、RawJob 三种格式。其中，PodMonitor 和 ServiceMonitor 的采集频率参数为 **interval**，RawJob 的采集频率参数为 **scrape_interval**。通过调整采集频率，您可以降低数据存储费用，调整如下图所示：

- PodMonitor/ServiceMonitor 格式配置。



- RawJob 格式配置。



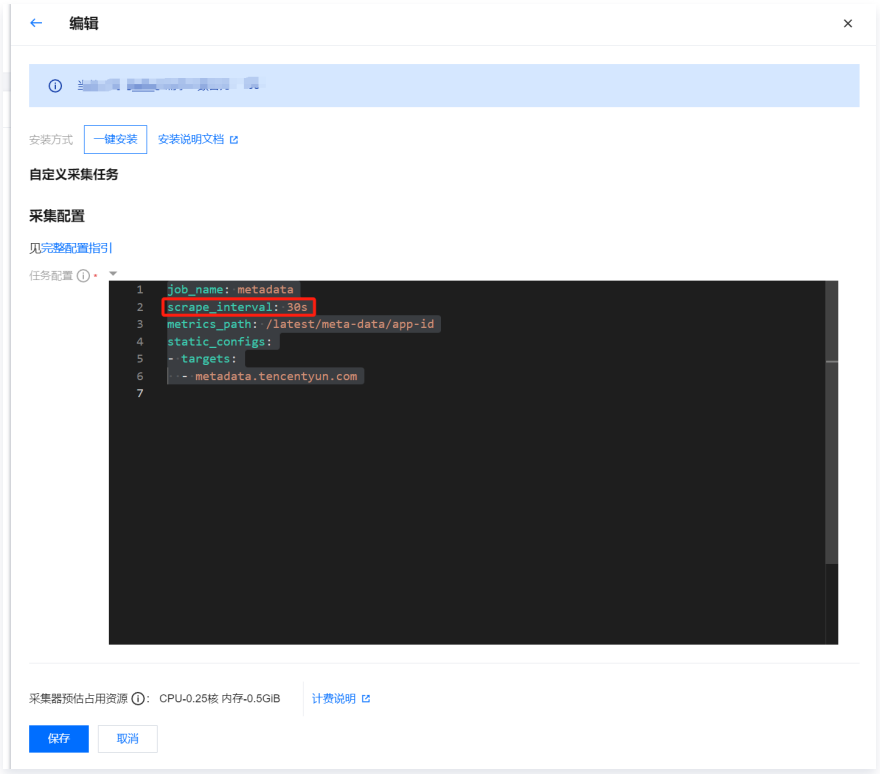
6. 配置完成后单击**确定**。

通过集成中心的控制台调整采集间隔

> **说明：**
> “抓取任务”集成已经一键安装成功，若未安装需要先单击右侧的**一键安装**后再开始下列操作。

1. 登录 腾讯云可观测平台控制台，选择左侧导航栏中的 Prometheus 监控 。
2. 在监控实例列表页，选择需要配置数据采集规则的实例名称，进入该实例详情页。
3. 在**数据采集**页面，单击**集成中心**，选择**抓取任务**，在弹出的窗口中选择**已集成**， 如下图所示：

4. 在集成列表页面，单击任意一个名称，进入采集任务配置页，找到 **scrape_interval** 字段，该参数为采集频率，您可以调大以降低数据存储费用，如下图所示：



5. 配置完成后单击**保存**。

组件版本升级说明

最近更新时间：2024-05-28 18:04:11

以下是针对 kube-state-metrics 组件版本升级（ 1.9.7 - 2.7.0 ）带来的指标变化以及兼容方案的说明。

新增加指标

serviceaccount 相关指标

kube_serviceaccount_info
kube_serviceaccount_created
kube_serviceaccount_deleted
kube_serviceaccount_secret
kube_serviceaccount_image_pull_secret
kube_serviceaccount_annotations
kube_serviceaccount_labels

clusterrole 相关指标

kube_clusterrole_annotations
kube_clusterrole_labels
kube_clusterrole_info
kube_clusterrole_created
kube_clusterrole_metadata_resource_version

role 相关指标

kube_role_annotations
kube_role_labels
kube_role_info
kube_role_created
kube_role_metadata_resource_version

rolebinding 相关指标

kube_rolebinding_annotations
kube_rolebinding_labels
kube_rolebinding_info
kube_rolebinding_created
kube_rolebinding_metadata_resource_version

certificatesigningrequest 相关指标

kube_certificatesigningrequest_annotations

kube_certificatesigningrequest_created
kube_certificatesigningrequest_condition
kube_certificatesigningrequest_labels
kube_certificatesigningrequest_cert_length

ingressclass 相关指标

kube_ingressclass_annotations
kube_ingressclass_info
kube_ingressclass_labels
kube_ingressclass_created

其他指标

kube_daemonset_annotations
kube_configmap_annotations
kube_configmap_labels
kube_cronjob_annotations
kube_cronjob_status_last_successful_time
kube_cronjob_metadata_resource_version
kube_cronjob_spec_successful_job_history_limit
kube_cronjob_spec_failed_job_history_limit
kube_deployment_annotations
kube_deployment_status_replicas_ready
kube_endpoint_annotations
kube_endpoint_address_not_ready
kube_endpoint_address_available
kube_endpoint_ports
kube_endpoint_address
kube_horizontalpodautoscaler_info
kube_horizontalpodautoscaler_annotations
kube_horizontalpodautoscaler_status_target_metric
kube_ingress_annotations
kube_ingress_info
kube_job_annotations
kube_namespace_annotations
kube_networkpolicy_annotations

kube_poddisruptionbudget_annotations
kube_poddisruptionbudget_labels
kube_replicaset_annotations
kube_secret_annotations
kube_statefulset_annotations
kube_statefulset_status_replicas_available
kube_statefulset_status_replicas_available
kube_storageclass_annotations

变更指标

变更前	变更后
kube_pod_container_resource_requests_cpu_cores	kube_pod_container_resource_requests{resource="cpu", unit="core"}
kube_pod_container_resource_limits_cpu_cores	kube_pod_container_resource_limits{resource="cpu", unit="core"}
kube_pod_container_resource_requests_memory_bytes	kube_pod_container_resource_requests{resource="memory", unit="byte"}
kube_pod_container_resource_limits_memory_bytes	kube_pod_container_resource_limits{resource="memory", unit="byte"}
kube_node_status_capacity_pods	kube_node_status_capacity{resource="pods", unit="integer"}
kube_node_status_capacity_cpu_cores	kube_node_status_capacity{resource="cpu", unit="core"}
kube_node_status_capacity_memory_bytes	kube_node_status_capacity{resource="memory", unit="byte"}
kube_node_status_allocatable_pods	kube_node_status_allocatable{resource="pods", unit="integer"}
kube_node_status_allocatable_cpu_cores	kube_node_status_allocatable{resource="cpu", unit="core"}
kube_node_status_allocatable_memory_bytes	kube_node_status_allocatable{resource="memory", unit="byte"}

兼容方案

目前已将变更指标中的8个免费指标添加入指标列，如存在旧实例升级的需求，可以参考如下两种方案，视需求选择一种即可：

采集解决方案

如集群较少，可以直接修改集群内的 kube-state-metrics 采集任务，名称为 kube-system/kube-state-metrics，类型为 servicemonitor，在 spec.endpoints[0].metricRelabelings 下面添加如下 relabel 规则，将新指标替换为旧指标。

```
- action: replace
  regex: kube_pod_container_resource_requests;cpu;core
  targetLabel: __name__
  replacement: kube_pod_container_resource_requests_cpu_cores
  sourceLabels: [__name__, resource, unit]
- action: replace
  regex: kube_pod_container_resource_limits;cpu;core
  targetLabel: __name__
  replacement: kube_pod_container_resource_limits_cpu_cores
  sourceLabels: [__name__, resource, unit]
- action: replace
  regex: kube_pod_container_resource_requests;memory;byte
```

```
    targetLabel: __name__
    replacement: kube_pod_container_resource_requests_memory_bytes
    sourceLabels: [__name__, resource, unit]
  - action: replace
    regex: kube_pod_container_resource_limits;memory;byte
    targetLabel: __name__
    replacement: kube_pod_container_resource_limits_memory_bytes
    sourceLabels: [__name__, resource, unit]
  - action: replace
    regex: kube_node_status_capacity;cpu;core
    targetLabel: __name__
    replacement: kube_node_status_capacity_cpu_cores
    sourceLabels: [__name__, resource, unit]
  - action: replace
    regex: kube_node_status_capacity;memory;byte
    targetLabel: __name__
    replacement: kube_node_status_capacity_memory_bytes
    sourceLabels: [__name__, resource, unit]
  - action: replace
    regex: kube_node_status_allocatable;cpu;core
    targetLabel: __name__
    replacement: kube_node_status_allocatable_cpu_cores
    sourceLabels: [__name__, resource, unit]
  - action: replace
    regex: kube_node_status_allocatable;memory;byte
    targetLabel: __name__
    replacement: kube_node_status_allocatable_memory_bytes
    sourceLabels: [__name__, resource, unit]
```

预聚合解决方案

如实例中绑定了多个集群，不便于修改每个集群中的 kube-state-metrics 的采集配置时，可以考虑如下预聚合的方式，还原旧指标。
注意：如下预聚合规则中的间隔 interval 要和采集任务对齐，如用户无任何修改则默认为15s。

```
name: kube-state-metrics-compatibility.rules
interval: 15s
rules:
  - expr: kube_pod_container_resource_requests{resource="cpu", unit="core"}
    record: kube_pod_container_resource_requests_cpu_cores
  - expr: kube_pod_container_resource_limits{resource="cpu", unit="core"}
    record: kube_pod_container_resource_limits_cpu_cores
  - expr: kube_pod_container_resource_requests{resource="memory", unit="byte"}
    record: kube_pod_container_resource_requests_memory_bytes
  - expr: kube_pod_container_resource_limits{resource="memory", unit="byte"}
    record: kube_pod_container_resource_limits_memory_bytes
  - expr: kube_node_status_capacity{resource="cpu", unit="core"}
    record: kube_node_status_capacity_cpu_cores
  - expr: kube_node_status_capacity{resource="memory", unit="byte"}
    record: kube_node_status_capacity_memory_bytes
  - expr: kube_node_status_allocatable{resource="cpu", unit="core"}
    record: kube_node_status_allocatable_cpu_cores
  - expr: kube_node_status_allocatable{resource="memory", unit="byte"}
    record: kube_node_status_allocatable_memory_bytes
```

如果预聚合如无数据，请检查是否有采集对应新指标。默认采集覆盖这两个指标的功能目前正在发布，存量可能存在未采集的情况。

相关资源使用及计费说明

最近更新时间：2024-12-05 16:07:34

当您使用 Prometheus 监控服务（TMP）服务时，可能会使用到 **EKS 集群**、**Grafana 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

配置数据采集，需要进行初始化，创建采集端所需的 EKS 集群，包括以下使用场景：

- 关联容器集群来监控容器服务。
- 在集成中心安装集成常用组件。

初始化入口如下图所示：

在 集群列表页 可查看已创建的 EKS 集群，如下图所示：

特殊说明：如果账号类型是非带宽上移账号，需要先 升级 为带宽上移账号后，才能进行初始化并创建 EKS 集群。

注意事项

该 EKS 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见下表：

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
<50w	1.25核 1.6GiB	1.3元
100w	0.5核1.5GiB*2	5.5元
500万	1核3GiB*3	11元
2000万	1核6GiB*5	30元
3000万	1核6GiB*8	48元

EKS 集群成本示例
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：0.12 x 24 + 0.05 x 24 = 4.08 元。

Grafana 服务

使用场景

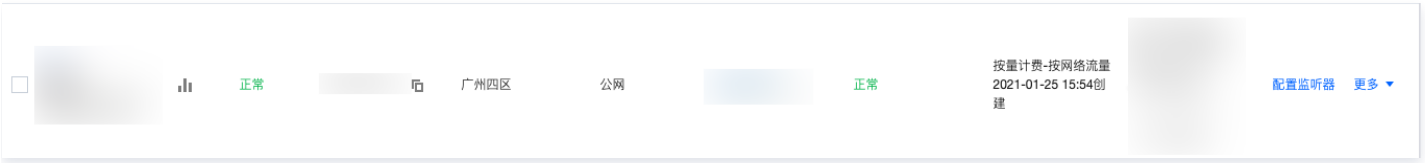
当您创建 Prometheus 实例时，需要绑定一个地域相同的 Grafana 实例，用于可视化展示 Prometheus 采集的监控数据。详细说明请参见 Grafana 服务计费说明 。

负载均衡 CLB

使用场景

若用户使用 Prometheus 监控服务时，关联了边缘集群或跨 VPC 关联了未打通网络的集群，需要创建公网的 CLB 进行网络联通，此时会创建一个公网 CLB。

我们将会对这些 CLB 资源收取费用，可在 负载均衡控制台 查看创建的公网 LB 信息，如下图所示：



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。