

Prometheus 监控服务 实践教学



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分的内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

实践教程

自建 Prometheus 迁入

云服务器场景下自定义接入

容器场景监控

基于 Prometheus 多维能力的告警优化

Prometheus 监控服务如何接入本地 Grafana

配置 Prometheus 公网地址

Prometheus 实例访问公网

使用实例诊断分析 Prometheus 问题

实践教程

自建 Prometheus 迁入

最近更新时间：2024-12-05 16:07:34

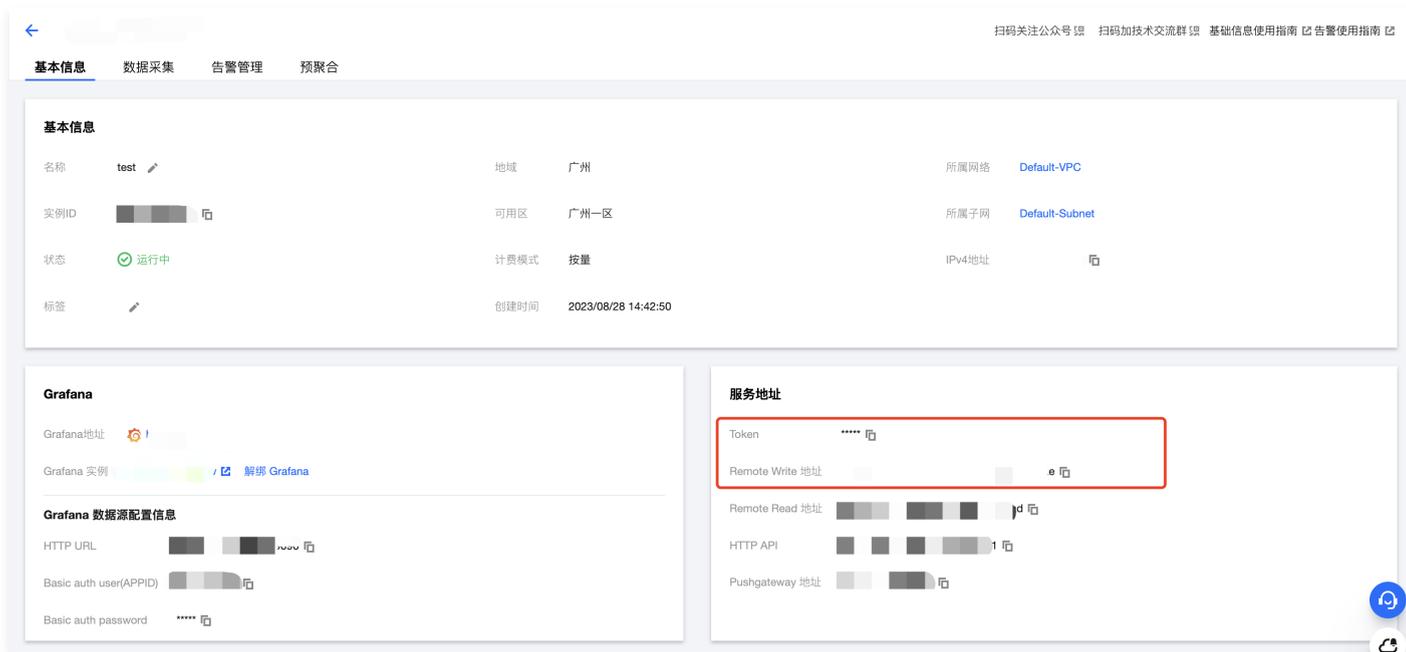
操作场景

已经有自建 Prometheus 需快速迁移到 Prometheus 监控服务。

操作步骤

Prometheus 本身支持 Remote Write 到一个外部存储，因此沿用这个思想，在自建 Prometheus 的配置文件中加一个 Remote Write 配置指向到 Prometheus 监控服务即可。具体操作步骤如下：

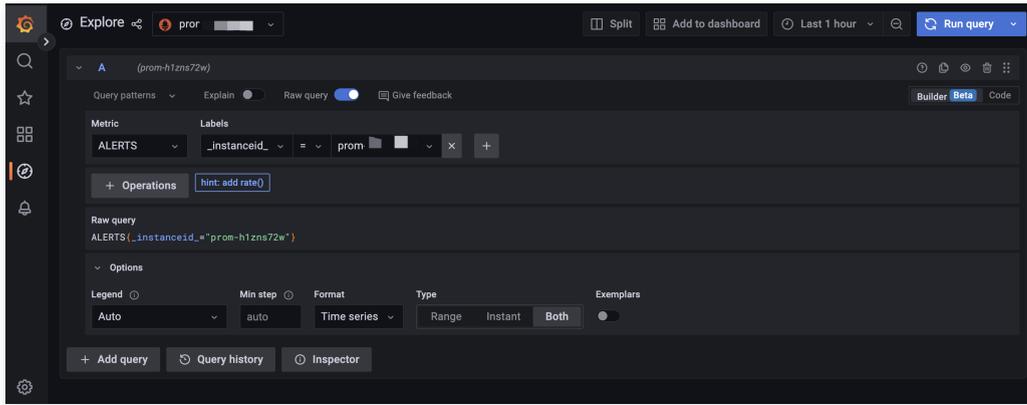
1. 选择对应的 Prometheus 实例，进入实例详情页，单击顶部菜单栏的**基本信息**，在**服务地址**卡片中获取 Prometheus 监控服务的 Remote Write 地址及 Token，如下：



2. 修改 `prometheus.yml`，修改完成 **重启** Prometheus，具体配置如下，如需了解更多 Remote Write 配置参数，请参见 [remote_write](#)。

```
remote_write:
  - name: cm_prometheus # Remote write 的名称
    url: http://ip:port/api/v1/prom/write # 从 Prometheus 基本信息中获取 Remote Write 地址，建议加上双引号避免特殊字符解析错误
    remote_timeout: 30s # 根据实际情况设置
    bearer_token: k32*****trR # 从 Prometheus 基本信息中获取 Token 信息
```

3. 打开 Prometheus 监控服务自带的 Grafana，通过 `Explore` 来验证数据是否写入成功，如下图，也可以 [自定义 Grafana 监控大盘](#)。



4. 也可以通过 Prometheus API 进行自建可视化, 详情请参见 [监控数据查询](#)。

云服务器场景下自定义接入

最近更新：2024-10-18 11:53:11

本文将详细介绍您如何接入腾讯云云服务器（CVM）如何从零开始接入 Prometheus 监控服务。

购买 Prometheus

说明

购买的 Prometheus 实例需跟监控的云服务器同一个 vpc 下，才能实现网络互通。

1. 登录 [Prometheus 监控服务控制台](#)，单击新建购买 Prometheus 实例。

实例ID/名称	监控/状态	可用区	Grafana访问地址	关联集群	网络	收费指标采集速率	配置	标签(key:value)	计费模式	操作
[实例ID]	运行中	广州三区	登录 Grafana	1/1	所属网络: 所属子网:	0个/秒	数据保存: 15天		按量	告警策略 关联集群 更多
[实例ID]	运行中	广州一区	登录 Grafana	1/1	所属网络: 所属子网:	2.27个/秒	数据保存: 15天		按量	告警策略 关联集群 更多

2. 在购买页中，选择合适的实例规格、网络。选择相同 vpc 网段，保证 Prometheus 能与需要采集的云服务器网段相同，才能采集到数据。实例规格，可根据自己的业务上报量进行选择。

Prometheus 监控服务 [返回产品详情](#) 产品文档 计费说明 产品控制台

计费模式: 免费试用 资源包 按量计费

地域及网络配置

地域: 中国 亚太 欧洲和美洲

广州 上海 中国香港 北京 成都 重庆 南京 上海金融 深圳金融
 北京金融 中国台北 上海自动驾驶云

处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择；例如，广州地域的服务无法通过内网上报数据到上海地域的Prometheus。

可用区: 广州三区 广州四区 广州六区 广州七区

网络:

如有私有网络/子网不符合您的要求，可以去控制台 [新建私有网络](#)

实例基础配置

数据存储时间: 15天 30天 45天 90天 180天 1年 2年

实例名称:

Grafana:

如有 Grafana 不符合您的要求，可以去控制台 [新建 Grafana](#)

标签 (选填):

如有标签/标签值不符合您的要求，可以去控制台 [新建](#)

协议条款: 我已阅读并同意相关服务条款 [《腾讯云服务协议》](#)、[《腾讯云 Prometheus 服务等级协议》](#)、[《计费概述》](#) 以及 [《欠费说明》](#)

数据上报费用 **1.45折** 0.60元/百万条/天

3. 选择完后，单击立即购买并支付即可。

说明

如需了解 Prometheus 更多定价规则，请参见 [产品定价](#)。

接入云服务器基础指标

1. 下载安装 node_exporter:

在需要上报数据的主机上，下载并安装 node_exporter，您可以点击进入 Prometheus 开源官网下载地址 [node_exporter](#)，也可以直接执行下列命令：

```
wget https://rig-1258344699.cos.ap-guangzhou.myqcloud.com/prometheus-agent/node_exporter -O node_exporter
```

目录为当前文件夹：

```
[root@VM-0-46-tencentos ~]# ls
node_exporter
[root@VM-0-46-tencentos ~]#
```

2. 运行 node_exporter 采集基础监控数据:

- 赋予权限，执行 node_exporter 并查看日志。

```
chmod +x node_exporter && nohup ./node_exporter &
cat nohup.out
```

如下图所示即为执行成功：

```
[root@M-0-13-tencentos ~]# chmod +x node_exporter && nohup ./node_exporter &
[2] 11678
[root@M-0-13-tencentos ~]# cat nohup.out
ts=2024-09-27T07:42:28.513Z caller=node_exporter.go:182 level=info msg="Starting node_exporter" version="(version=1.3.1, branch=HEAD, revision=a2321e70948ddcf26873612bccd7cd4c42b6b6)"
ts=2024-09-27T07:42:28.513Z caller=node_exporter.go:183 level=info msg="Build context" build_context="(go=1.17.3, user=root@243aaf5525c, date=20211205-11:09:49)"
ts=2024-09-27T07:42:28.513Z caller=node_exporter.go:185 level=warn msg="Node Exporter is running as root user. This exporter is designed to run as unprivileged user, root is not required."
ts=2024-09-27T07:42:28.513Z caller=filesystem_common.go:111 level=info collector=filesystem msg="Parsed flag --collector.filesystem.mount-points-exclude" flag="/dev/proc/mounts|/sysvar/lib/docker|/l(s)?"
ts=2024-09-27T07:42:28.513Z caller=filesystem_common.go:113 level=info collector=filesystem msg="Parsed flag --collector.filesystem.fs-types-exclude" flag="autofs|binfmt_misc|bpf|cgroup2|configs|debugfs|devtmpfs|fusectl|hugetlfs|iso9660|nfs|nfsfs|overlay|proc|procfs|pstore_rpc_ppefs|securityfs|linuxfs|squashfs|sysfs|tracefs"
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:188 level=info msg="Enabled collectors"
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=arp
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=bcache
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=bonding
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=brifs
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=conntrack
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=cpu
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=cpufreq
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=diskstats
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=dmi
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=dmcc
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=entropy
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=fibrechannel
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=filefd
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=filesystem
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=hwmon
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=infiniband
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=ipvs
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=loadavg
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=mdadm
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=meminfo
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=netclass
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=netdev
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=netstat
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=nfs
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=nfsd
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=nmme
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=nvme
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=powersupplyclass
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=pressure
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=rapl
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=schedstat
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=sockstat
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=softnet
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=stat
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=tapestats
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=traffic
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=thermal_zone
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=timex
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=linux
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=udp_queues
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=uname
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=vmstat
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=xfs
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:115 level=info collector=zfs
ts=2024-09-27T07:42:28.514Z caller=node_exporter.go:199 level=info msg="Listening on" address=:9100
ts=2024-09-27T07:42:28.514Z caller=tlscfg.go:195 level=info msg="TLS is disabled." http2=false
nohup: ignoring input and appending output to 'nohup.out'
[root@M-0-13-tencentos ~]#
```

- 可通过下列命令，查看暴露在9100端口的监控数据：

```
curl 127.0.0.1:9100/metrics
```

下图为执行命令后看到的暴露出来的指标监控数据：

```
[root@M-0-13-tenecentos ~]# curl 127.0.0.1:9100/metrics
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 7
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.17.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.38724e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.38724e+06
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.445375e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 753
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 0
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 4.19836e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
```

3. 新增抓取任务：

- i. 登录 [Prometheus 监控服务控制台](#)，选择对应 Prometheus 实例进入管理页面。
- ii. 进入数据收集 > 集成中心,在集成中心页面找到CVM 云服务器卡片并点击弹出集成页面，根据页面提示进行配置即可。

CVM 云服务器 (cvm-instance-exporter)
✕

安装
Dashboard
已集成

🔔 当前子网 [【lyla】](#) 剩余IP数目为: 100

安装方式
一键安装
[安装说明文档](#)

CVM 采集任务

1. 指标上报

基础指标，安装 `node_exporter`，[操作指引](#)。

自定义业务指标，[操作指引](#)。

2. 采集配置

[完整配置指引](#)

任务配置 ▾

```

1  job_name: example-job-name
2  metrics_path: /metrics
3  cvm_sd_configs:
4    - region: ap-guangzhou
5      ports:
6        - 9100
7      filters:
8        - name: tag:示例标签键
9          values:
10         - 示例标签值
11  relabel_configs:
12    - source_labels: [__meta_cvm_instance_state]
13      regex: RUNNING
14      action: keep
15    - regex: __meta_cvm_tag_(.*)
16      replacement: $1
17      action: labelmap
18    - source_labels: [__meta_cvm_region]
19      target_label: region
20      action: replace
                
```

采集器预估占用资源 🔔: CPU-0.25核 内存-0.5GB | 配置费用: 0.01元/小时 原价:-0.05元/小时 仅采集免费指标的情况下不收费, [计费说明](#)

保存
取消

iii. 抓取任务参考配置 如下：

```

job_name: example-job-name
metrics_path: /metrics
cvm_sd_configs:
- region: ap-guangzhou
  ports:
  - 9100
  filters:
  - name: tag:示例标签键
    values:
    - 示例标签值
relabel_configs:
- source_labels: [__meta_cvm_instance_state]
  regex: RUNNING
  action: keep
- regex: __meta_cvm_tag_(.*)
  replacement: $1
  action: labelmap
- source_labels: [__meta_cvm_region]
  target_label: region
  action: replace
    
```

如需使用公网 IP 作为采集目标，需要在抓取任务中增加以下 relabel 规则：

```

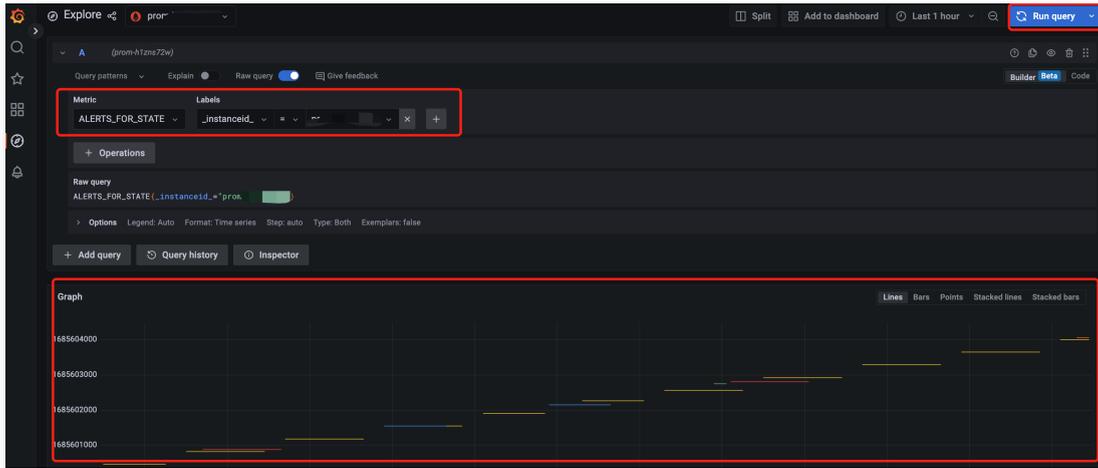
- action: replace
  regex: ([^:]+):(.*):(.+)
  replacement: $3:$2
  source_labels:
  - __address__
  - __meta_cvm_public_ip
  target_label: __address__
    
```

4. 查看数据是否上报成功：

i. 登录 Prometheus 监控服务控制台，单击 Grafana 访问地址列的登录 Grafana。

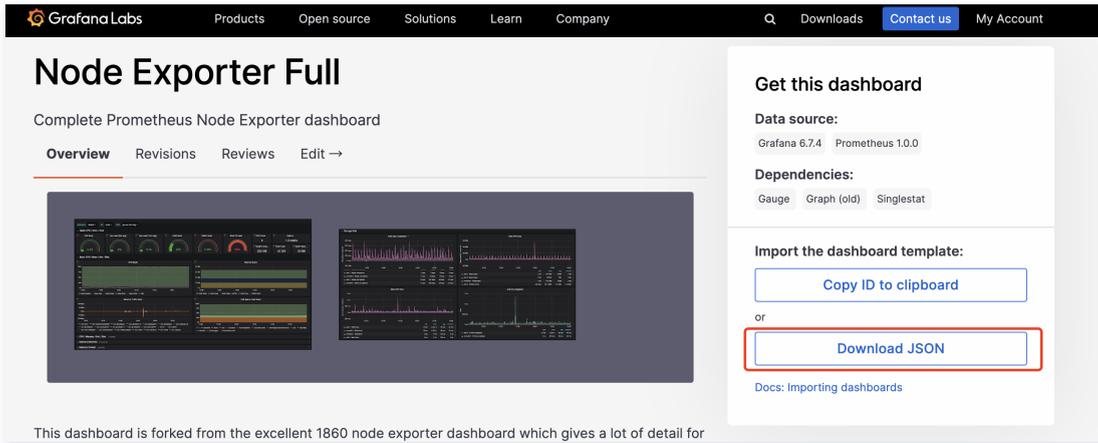
实例ID/名称	监控/状态	可用区	Grafana访问地址	关联集群	网络	收费指标采集速率	配置	标签(key=value)	计费模式	操作
[实例ID]	运行中	广州三区	登录 Grafana	1/1	所属网络: 所属子网:	0个/秒	数据保存: 15天		按量	告警策略 关联集群 更多
[实例ID]	运行中	广州一区	登录 Grafana	1/1	所属网络: 所属子网:	2.27个/秒	数据保存: 15天		按量	告警策略 关联集群 更多

ii. 如下图所示，到 Explore 搜索下 `{job="cvm_node_exporter"}` 查看是否有数据，若有数据，则表示上报成功。

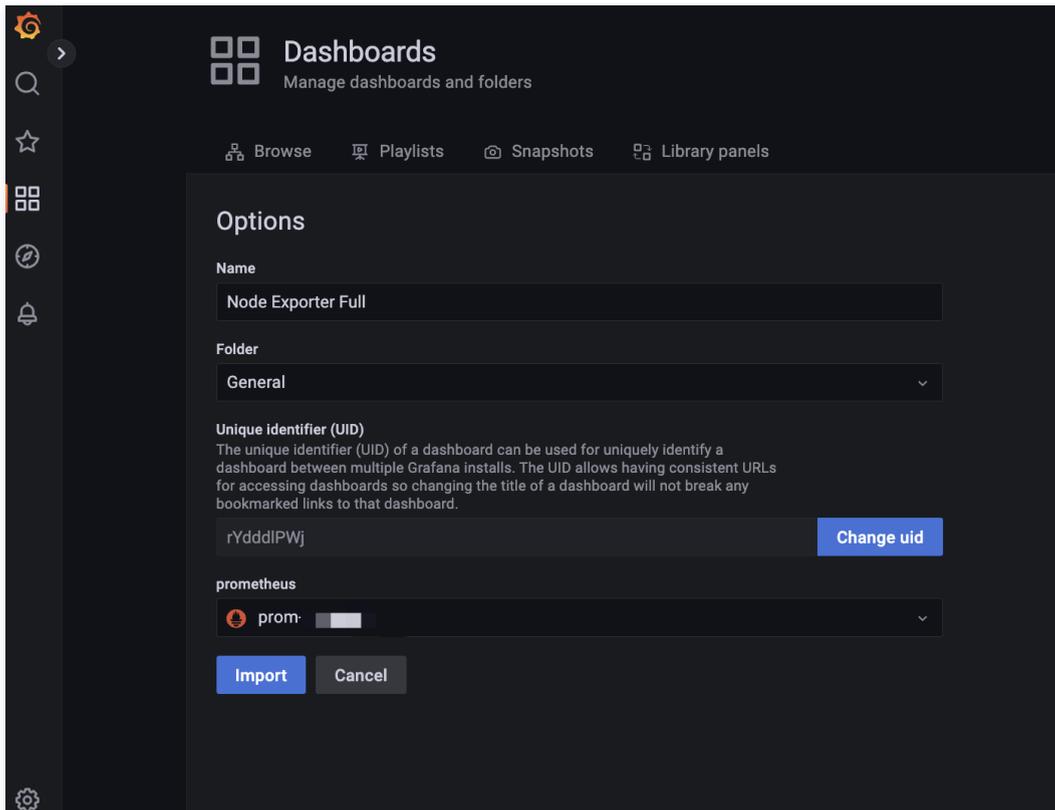


5. 配置 Dashboard 界面: Dashboard 界面每个产品都会有一些现成的 json 文件, 可以直接导入。

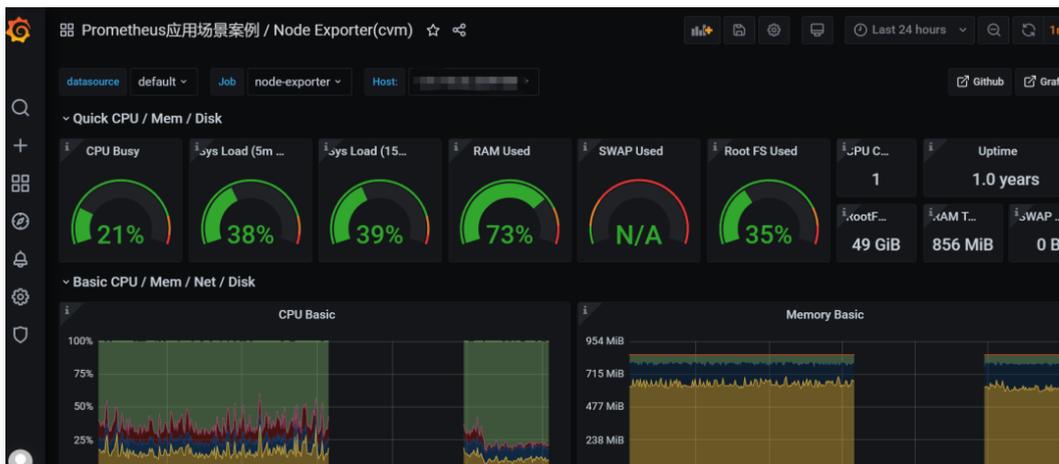
i. 下载 Dashboard 文件: 登录 [Dashboard-Node Exporter Full](#) 页面, 选择最新的 Dashboard 并下载。



ii. 导入 Dashboard 的 json 文件: 登录 [Prometheus 监控服务控制台](#), 进入 [基本信息](#) > [Grafana 地址](#), 单击进入 Grafana, 在 [Grafana 控制台](#) > [Dashboards](#) > [Import](#) > 在 [Upload JSON file](#) 中上传 Dashboard 文件。



配置完后的效果如下：



接入云服务器业务层指标

Prometheus 根据监控的不同场景提供了 Counter、Gauge、Histogram、Summary 四种指标类型。Prometheus 社区提供了多种开发语言的 SDK，每种语言的使用方法基本上类似，主要是开发语言语法上的区别，下面主要以 Go 作为例子，使用 Counter 指标类型如何上报自定义监控指标数据。其余指标类型请参见 [自定义监控](#)。

Counter

计数类型，数据是单调递增的指标，服务重启之后会重置。可以用 Counter 来监控请求数、异常数、用户登录数和订单数等。

1. 如何通过 Counter 来监控订单数：

```
package order

import (
    "github.com/prometheus/client_golang/prometheus"
    "github.com/prometheus/client_golang/prometheus/promauto"
)
```

```
// 定义需要监控 Counter 类型对象
var (
    opsProcessed = promauto.NewCounterVec(prometheus.CounterOpts{
        Name: "order_service_processed_orders_total",
        Help: "The total number of processed orders",
    }, []string{"status"}) // 处理状态
)

// 订单处理
func makeOrder() {
    opsProcessed.WithLabelValues("success").Inc() // 成功状态
    // opsProcessed.WithLabelValues("fail").Inc() // 失败状态

    // 下单的业务逻辑
}
```

例如，通过 `rate()` 函数获取订单的增长率：

```
rate(order_service_processed_orders_total[5m])
```

2. 暴露 Prometheus 指标：

通过 `promhttp.Handler()` 把监控埋点数据暴露到 HTTP 服务上。

```
package main

import (
    "net/http"

    "github.com/prometheus/client_golang/prometheus/promhttp"
)

func main() {
    // 业务代码

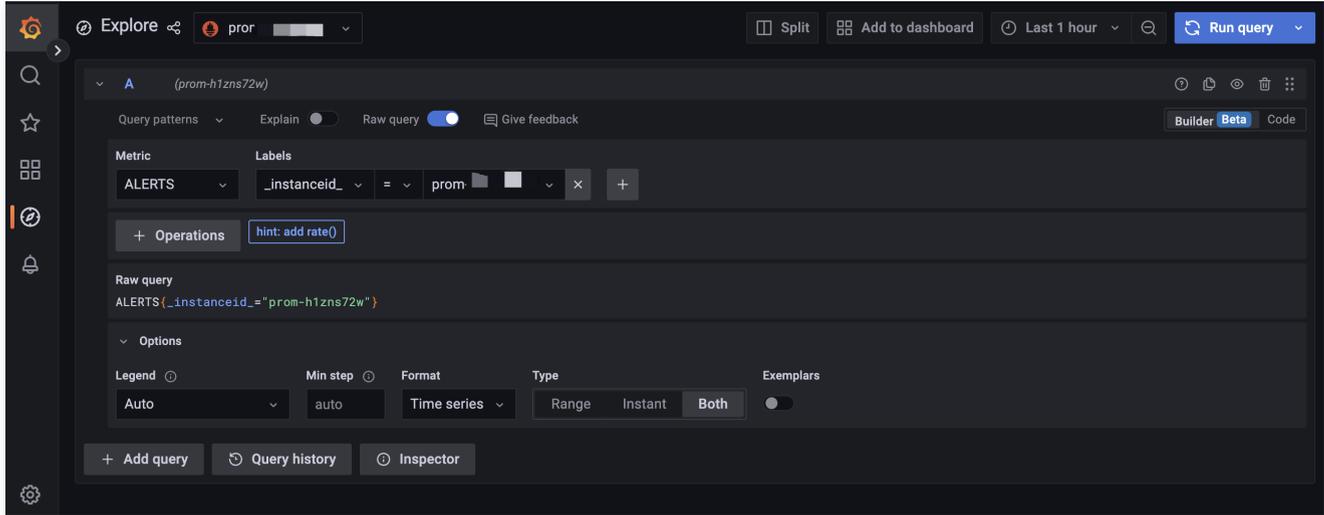
    // 把 Prometheus 指标暴露在 HTTP 服务上
    http.Handle("/metrics", promhttp.Handler())

    // 业务代码
}
```

3. 采集数据：

完成相关业务自定义监控埋点之后，应用发布，即可通过 Prometheus 来抓取监控指标数据。采集完成后，等待数分钟，您可在 Prometheus 监控服务集

成的 Grafana 中查看业务指标监控数据。



容器场景监控

最近更新时间：2024-11-07 10:34:33

实践背景

众所周知，Prometheus 是容器场景的最佳监控工具，但自建 Prometheus 对于运维人力有限的中小型企业而言，成本太高；对于业务发展快速的大企业又容易出现性能瓶颈。因而使用云上托管 Prometheus 已成为越来越多上云企业的第一选择。那我们该如何使用托管 Prometheus 监控 [腾讯云容器服务 \(TKE\)](#)。

实践步骤

步骤1：购买实例

1. 登录 [Prometheus 监控服务控制台](#)。
2. 单击新建，根据您的需求选择购买地域、存储时长并选择需要关联的 Grafana 实例，若无 Grafana 实例，可参见 [相关操作指引](#) 创建（需创建实例并完成购买）。
3. 完成后，单击立即购买即可。详细计费规则请查看 [计费规则](#)。

步骤2：集成容器服务

1. 新建完实例后，在实例列表中单击对应的实例 ID/名称，进入实例详情页面。
2. 在顶部导航栏中单击数据采集 > 集成容器服务 > 关联集群。
3. 在弹窗中选择需要关联的集群，共支持接入4种集群类型（标准集群、弹性集群、注册集群、边缘集群），同时支持集群跨 VPC，不同 VPC 网络不互通的情况需要勾选创建公网 CLB。

关联集群

① 当前子网 [su-...] 目为: 158

- 关联集群后, 会默认采集 [免费的基础指标](#), 如需配置更多指标采集, 可通过 [数据采集配置](#) 功能进行配置。

集群类型: 标准集群

跨VPC关联: 启用
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 [计费说明](#)

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, 否则无法进行数据采集。

集群所在地域: 成都
处在不同地域的云产品内网不通, 购买后不能更换。建议选择靠近您客户的地域, 以降低访问延时、提高下载速度。

集群: **当前地域下有以下可用集群**

ID/集群名	类型	所属VPC	状态
<input checked="" type="checkbox"/> 体验测试-re...	标准...	vpc-...	Run...
<input type="checkbox"/> 测试集群-勿删	标准...	vpc-...	Run...
<input checked="" type="checkbox"/> ceph-...	标准...	vpc-...	Run...
<input type="checkbox"/> ...	标准...	vpc-...	Run...
<input checked="" type="checkbox"/> cls-d1wd08an ceph-low-勿删	标准...	vpc-...	Run...

请为每个集群预留0.5核100M以上资源

全局标记: 启用
标签键名称不超过63个字符, 仅支持英文、数字、'_' (但不允许以 '_' 开头)。支持使用前缀, 更多说明[查看详情](#)
 标签键值只能包含字母、数字及分隔符('-', '_', '.', ':'), 且必须以字母、数字开头和结尾

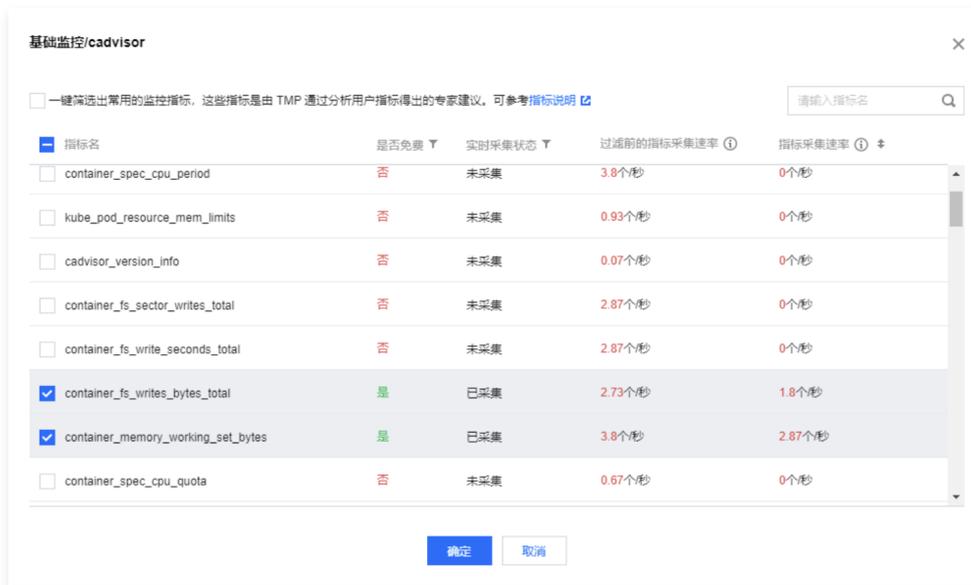
默认预聚合规则: 已全部开启 [预聚合规则管理](#)
用于常用监控指标的预设Dashboard图表展示或告警规则模板, 预聚合规则会生成新指标, 正常计费。默认预聚合规则列表

① 您所选的集群与当前 Prometheus 实例不在同一VPC, 请确认跨 VPC 之间的网络互通性或选择创建公网CLB, 实例VPC为: [vpc-96dygrec](#)

采集器预估占用资源 CPU-3核 内存-6GIB [计费说明](#)

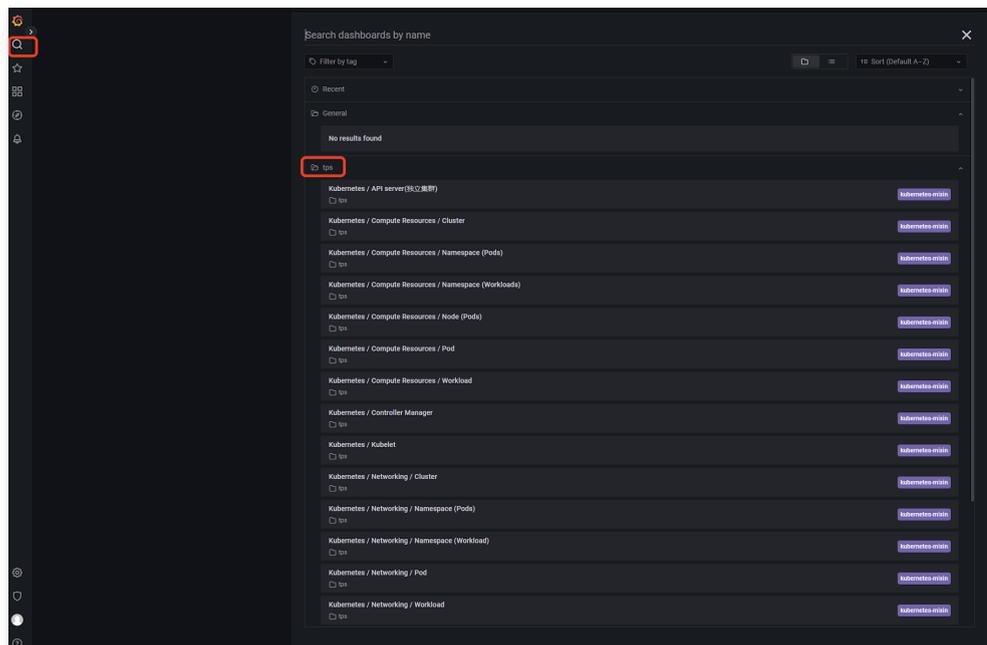
确定
取消

4. 关联集群后, 可以点击已关联集群列表页中操作列[数据采集配置](#), 进入数据采集配置页面, 再点击基础监控操作列的[指标详情](#), 查看默认采集的免费基础指标, 同时可以调整新增或减少采集指标。

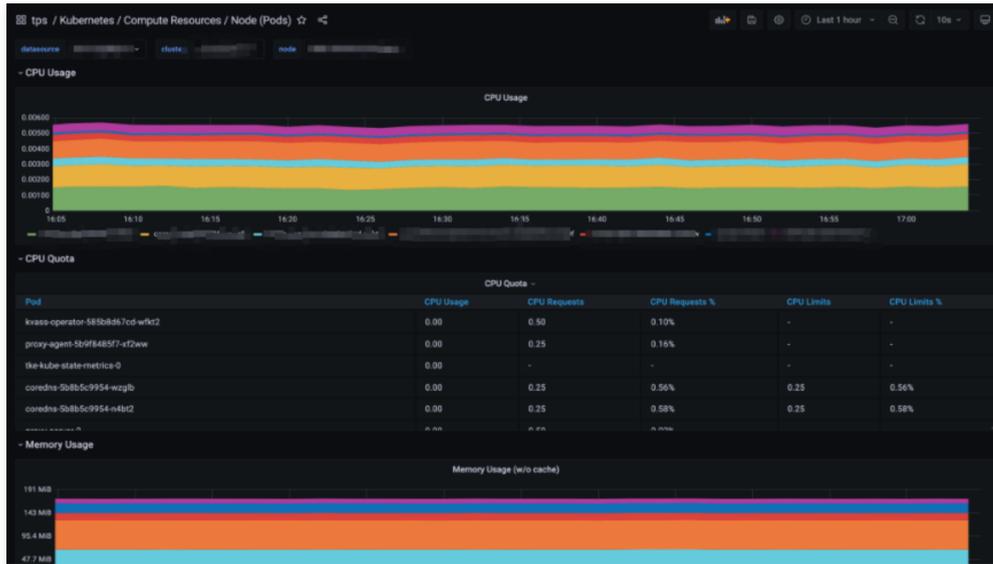


步骤3: Grafana 查看监控数据

1. 在实例列表中单击实例名称右侧的 Grafana 图标，进入 Grafana 服务平台。
2. 在 Grafana 服务平台 > Dashboard 搜索列表，默认预设了容器相关的监控面板，单击某个面板名称。



进入面板页面，可以查看预设好的监控数据图表。



步骤4：配置告警策略

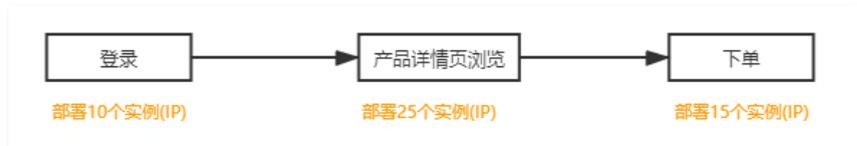
在实例列表中，选择对应的 Prometheus 实例，单击顶部菜单栏的告警管理，进入告警策略页面，单击新建告警策略，详情请参见 [新建告警策略](#)。

基于 Prometheus 多维能力的告警优化

最近更新时间：2024-11-07 10:34:33

实践背景

通常来说，监控系统的四个黄金指标（Four Golden Signals，参考 Google 运维解密）是错误类指标、延迟类指标、流量指标和饱和度指标，可以在服务级别衡量终端用户体验、服务质量、业务影响等层面的问题。以一个典型的电商服务关键路径（登录 > 产品浏览详情页 > 下单）举例。



针对图中三个服务，需要设定成功率等监控指标。传统的以实例(例如 IP)为监控对象的场景下，通常会对每个服务的实例都配置告警策略，用于满足最细粒度的告警对象质量监测，但也会带来一系列痛点：

- 需要针对每个实例都配置一条告警策略，告警策略太多，维护成本高。
- 缺乏整个业务层面的监控(监控对象是3个业务，而不是每个实例)。
- 业务异常时，多个实例同时触发告警，容易引起告警风暴。
- 对于其他维度的业务质量监控能力不足，例如业务分区部署，需要看某个区的质量情况等。

针对上述痛点，Prometheus 监控服务可以提供“优雅”的解决方案。通过将监控对象从实例扩展为不同标签（label），利用Prometheus 监控服务的多维能力，可以聚合出服务层面的宏观监控指标。详见下文。

监控场景

服务指标的宏观维度监控&告警场景

以上述电商服务关键路径举例。服务上报指标为成功率，上报标签为：服务名、IP 和区域。



相比传统只上报实例（IP）的单一标签，这里还扩展了服务名等其他标签。

在设置告警策略时，通过不同标签聚合，来解决上述提到的传统实例监控中碰到的痛点。具体步骤如下：

1. 按服务维度的成功率看整体情况。
2. 按服务&地区维度聚合，查看某个更细维度的质量 情况。
3. 利用 PromQL 可以大幅降低告警策略数量(对比传统实例监控)。

实践步骤

步骤1: 指标定义&服务部署

1. 指标定义

定义服务需要上报的指标和标签。举例，这里上报指标为：成功率(svcSuccessRate)。标签为：服务名(svc)，IP(ip)，区域(region)。

```
svcSuccessRate = prometheus.NewGaugeVec(  
    prometheus.GaugeOpts{  
        Name: "svcSuccessRate",  
        Help: "service success rate",  
    },  
    []string{"svc", "region", "ip"})
```

2. 服务部署

服务可以通过云服务器(CVM)的形式部署在云上(本文采用 CVM 的部署形式),也可以通过容器(TKE)的形式部署。部署到 CVM 上,并检查是否正常暴露指标(示例中通过8581端口暴露指标)。

```
svcSuccessRate{ip="192.168.1.2", region="ap-shanghai", svc="svc.prod1"} 0.9535
svcSuccessRate{ip="192.168.1.2", region="ap-singapore", svc="svc.deal"} 0.987
svcSuccessRate{ip="192.168.1.2", region="ap-singapore", svc="svc.log1n"} 0.94
svcSuccessRate{ip="192.168.1.2", region="ap-singapore", svc="svc.prod1"} 0.9242
[root@VM-16-14-tlinux /usr/local/testEsen]# curl http://localhost:8581/metrics
```

说明:

- Prometheus go client 提供了完善的指标接入库,详情请查看 [官方文档](#)。
- 更多语言 SDK 可参考 [自定义监控](#)。

步骤2: 配置抓取任务

1. 登录 [Prometheus 监控服务控制台](#), 点击**新建**, 创建 Prometheus 实例。
2. 点击新建的 Prometheus 实例, 进入实例详情页, 点击**数据采集 > 集成中心**, 搜索“抓取任务”, 再点击**一键安装**, 如下图:



3. 安装成功后再进行配置, 详细配置信息请参考 [cvm_sd_config 配置](#)。

新建
✕

• 当前子网 剩余IP数目为: 162

自定义采集任务

采集配置

[见完整配置指引](#)

任务配置 ⓘ

1

保存

取消

会产生额外费用, 计费概述 [🔗](#)

4. 验证数据是否抓取成功。通过 Grafana 查询对应的指标和标签。



步骤3: 配置告警策略

1. 配置服务宏观层面的告警：当服务的成功率低于两个9(0.99)时告警。



效果如下图所示，当上报的三个服务(登录，查看产品详情页，下单)成功率低于0.99时，就会触发告警。



2. 配置服务更细维度的告警：当服务某个地区的成功率低于0.99时，触发告警。



效果如下图所示，当上报的三个服务(登录，查看产品详情页，下单)在某个地区的成功率低于0.99时，就会触发告警。



Prometheus 监控服务如何接入本地 Grafana

最近更新时间：2024-11-08 18:03:02

若您需要在本地的 Grafana 系统中查看 Prometheus 监控服务相关数据，可以利用 Prometheus 监控服务提供的 HTTP API 来实现。本文介绍如何将 Prometheus 监控服务数据接入本地 Grafana 的实现方法。

操作步骤

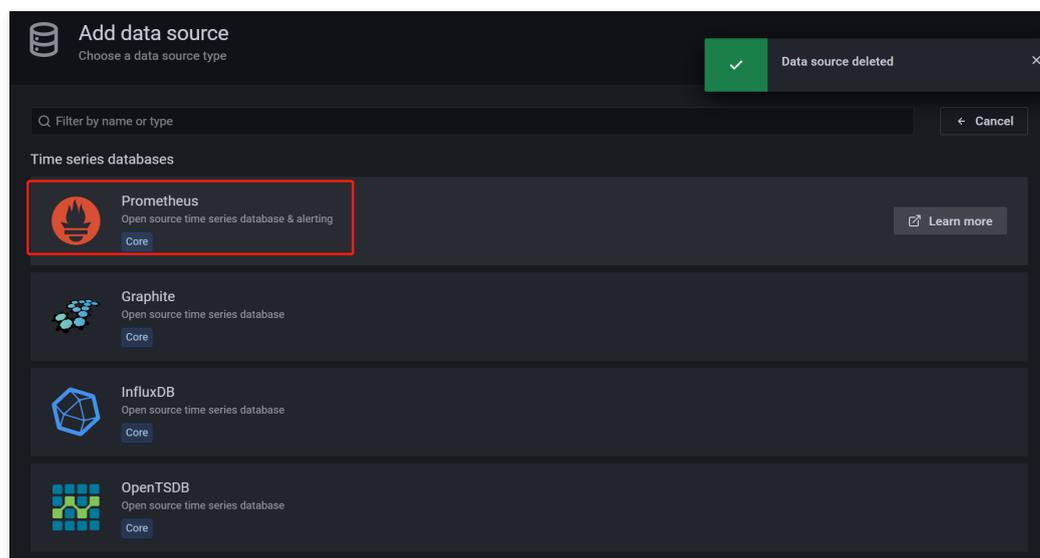
步骤1: 获取 Prometheus 监控服务提供的 HTTP API

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. 单击对应的按量付费实例，进入 Prometheus 基本信息页。
4. 在服务地址模块获取 HTTP API 地址。若您需要提高 Grafana 数据读取的安全性，可获取 Prometheus 实例的鉴权 Token，按照**步骤2**的指引填入即可。



步骤2: 在本地 Grafana 添加数据源

1. 以管理员账号登录本地 Grafana 系统。
2. 在左侧导航栏中选择 **Configuration > Data Sources**（非管理员无法查看该菜单）。
3. 在 Data Sources页中单击 **Add data source**。
4. 在 Add data source 页面中选择 **Prometheus**。



5. 在 Settings 页签的 Name 字段输入自定义的名称，在 URL 字段中粘贴步骤1: 「Grafana 数据源配置信息」中的 HTTP API。
6. 在 Auth 区域开启 Basic Auth 开关，设置 User 为「Grafana 数据源配置信息」中的 **Basic auth user**，设置 Password 为「Grafana 数据源配置信息」中的 **Basic auth password**。

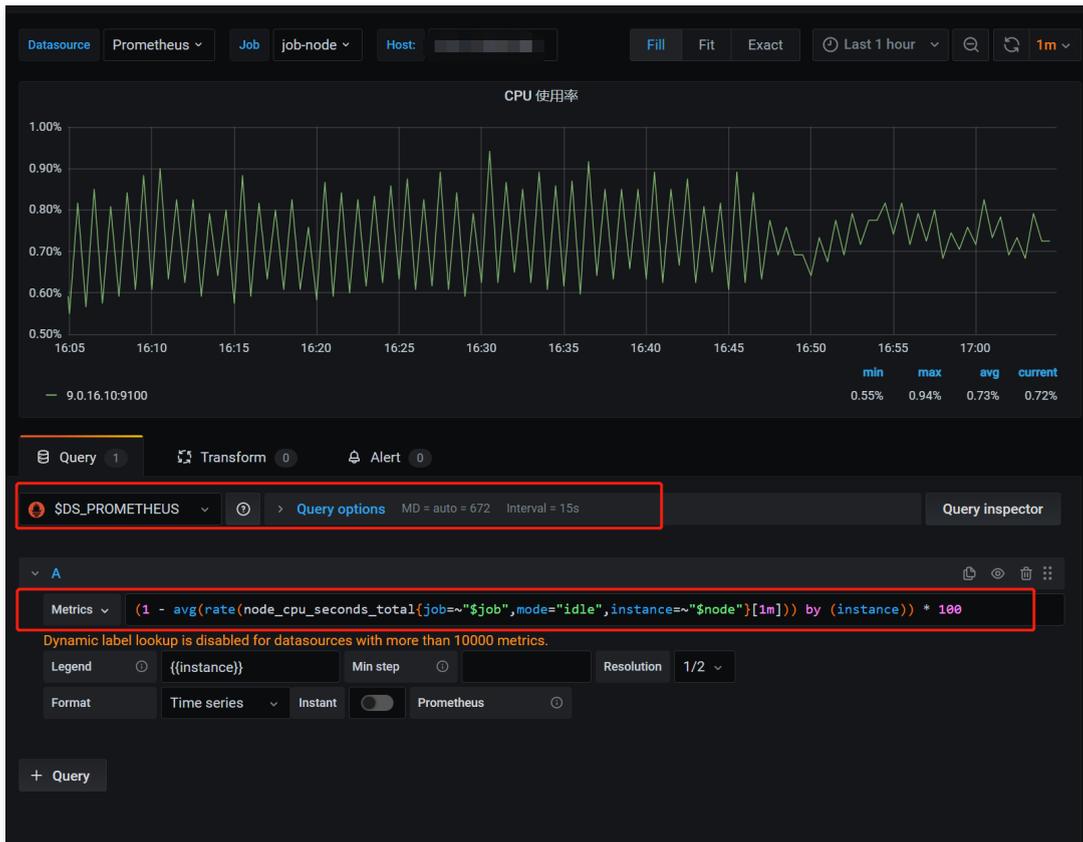
7. 完成后单击 **save & test** 即可。

The screenshot displays the Prometheus configuration interface in Grafana. It is divided into three main sections: HTTP, Auth, and Basic Auth Details. In the HTTP section, the URL field is highlighted with a red box and contains the text 'http://10.0.0.1:9090'. The Auth section has several toggle switches, with 'Basic auth' being turned on. The Basic Auth Details section has two input fields: 'User' containing '125' and 'Password' containing 'configured', both of which are highlighted with red boxes. There is also a 'Reset' button next to the password field.

步骤3: 验证是否添加成功

请按照以下步骤验证 Prometheus 监控服务是否成功接入本地 Grafana:

1. 登录您本地 Grafana 系统。
2. 在左侧导航栏中选择  > **New dashboard**。
3. 在 New dashboard 页面中单击 **Add a new panel**。
4. 在 Edit Panel 页面的 Query 页签的下拉框中, 选择步骤2中所添加的数据源(如下图红框1处)。
5. 在 A 区域的 Metrics 字段输入指标名称并按回车。
6. 若能显示出相应指标的图表数据, 则说明操作成功。否则请检查填写的接口地址或 Token 是否正确, 以及步骤2中添加的数据源是否有 Prometheus 监控服务数据。



配置 Prometheus 公网地址

最近更新時間：2024-12-24 14:16:32

本文將為您介紹如何為 Prometheus 監控配置公網地址。

操作步驟

步驟1：購買 Prometheus 實例

1. 登錄 [Prometheus 監控服務控制台](#)。
2. 单击左上角的新建，進入 Prometheus 購買頁，可根據自己的實際情況購買對應的實例，詳情請參見 [創建實例](#)。
3. 成功購買後，单击創建的實例 ID/名稱，進入實例詳情頁的**基本信息**，獲取 Prometheus 所屬網絡以及IPV4 地址。

基本信息	數據採集	告警管理	預聚合	實例診斷	
名稱	test	地域	廣州	所屬網絡	VPC-...
實例ID		可用區	廣州四區	所屬子網	subnet-...-4
狀態	運行中	計費模式	按量	IPv4地址	192.168.1.37
標籤		創建時間	2024/01/22 16:05:19		

步驟2：新建公網 CLB 實例

1. 進入 [負載均衡控制台](#)，新建 CLB 實例。
2. 根據提示選擇並填寫信息，具體操作說明可參見 [創建負載均衡實例](#)。網絡類型選擇公網，來開通 CLB 公網訪問。所屬網絡選擇 [步驟1](#) 獲取的 Prometheus 所屬網絡。

選擇配置

計費模式：按量計費

地域：廣州

網絡類型：公網 (已選中) | 內網

彈性公網IP：不選擇 (已選中)

IP版本：IPv4 (已選中) | IPv6 NAT64

所屬網絡：[VPC-...] (已選中)

實例規格：共享型

網絡計費模式：按帶寬計費 | 按使用流量 (已選中) | 共享帶寬包

帶寬上限：5 Mbps

數量：1

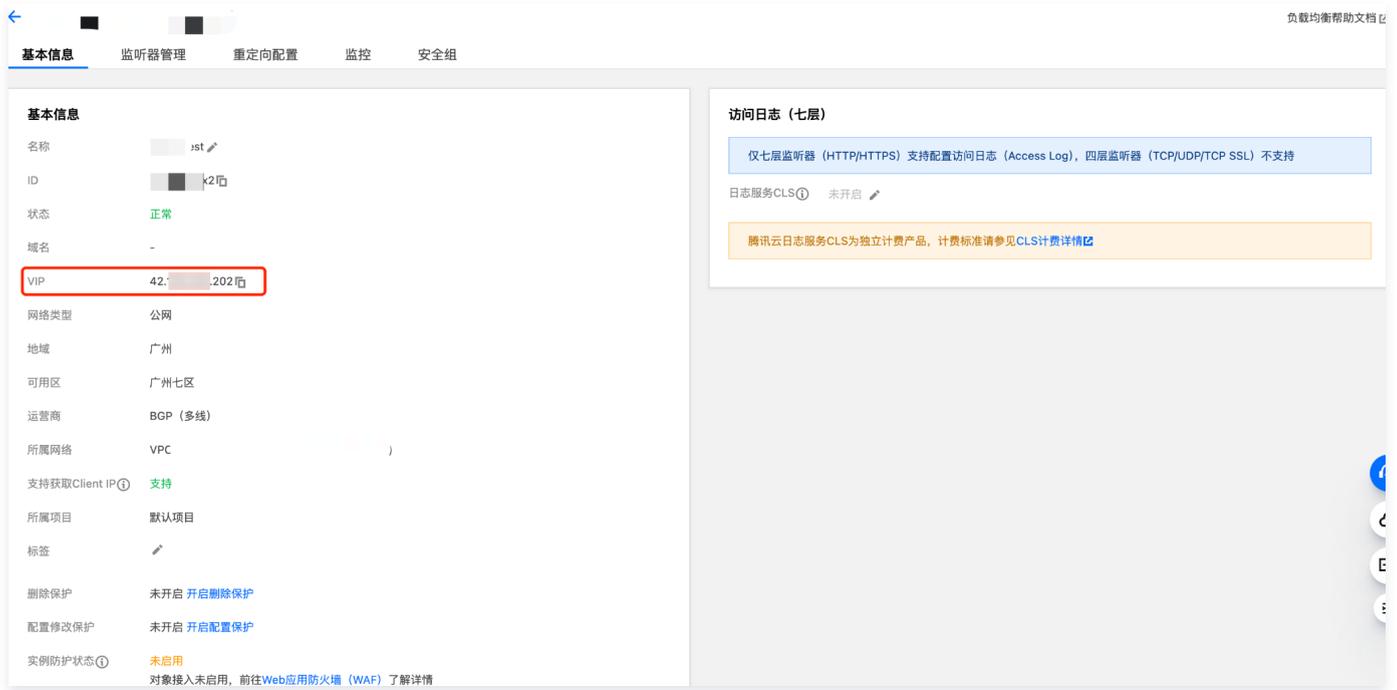
實例費用 | IP資源費 | 網絡費用

立即購買

說明：

需創建與 Prometheus 同 VPC 下的 CLB 實例，才可綁定 Prometheus 內網 IP。若已有公網的 CLB 實例且與 Prometheus 同 VPC，可復用已有公網 CLB。

3. 創建完後進入實例基本信息頁，可查看 CLB 對應的公網 IP。



步骤3: 绑定后端服务

1. 进入监听器管理页面。
2. 单击 TCP/UDP/TCP SSL/QUIC 监听器下的新建。



可参见 [负载均衡监听器](#) 操作指引新建监听器。

创建监听器 ✕

1 基本配置 >
 2 健康检查 >
 3 会话保持

名称
不能超过60个字符，只能使用中文、英文、数字、下划线、分隔符“-”、小数点、冒号

监听协议端口 TCP : 8080
端口范围: 1 - 65535

均衡方式① 加权轮询
WRR 根据新建连接数来调度，权重越高的后端服务器被轮询到的概率越高

[隐藏高级选项](#) ▲

双向RST① 解绑后端服务
勾选后，对应操作会向两端（客户端和服务端）发送RST报文来关闭连接；若不勾选，则不发送双向RST，长连接仍然存在直至超时

关闭
下一步

3. 新建完监听器后，单击监听器名称。在子窗口中单击**绑定**，绑定后端服务。



- 目标类型：选择 IP 类型。
- IP 地址：需填写步骤1获取的 Prometheus IPV4 地址。
- 端口：填写 Prometheus 默认开放端口9090。

绑定后端服务 ✕

目标类型① 实例 IP类型

默认端口 默认权重

① 注意：绑定内网IP支持绑定云联网关联的跨VPC、跨地域绑定后端云服务器，但不支持将CLB实例作为后端IP绑定。

IP	端口	权重①	
不支持ipv4地址	1-65535	- 10 +	添加端口 删除

添加内网IP

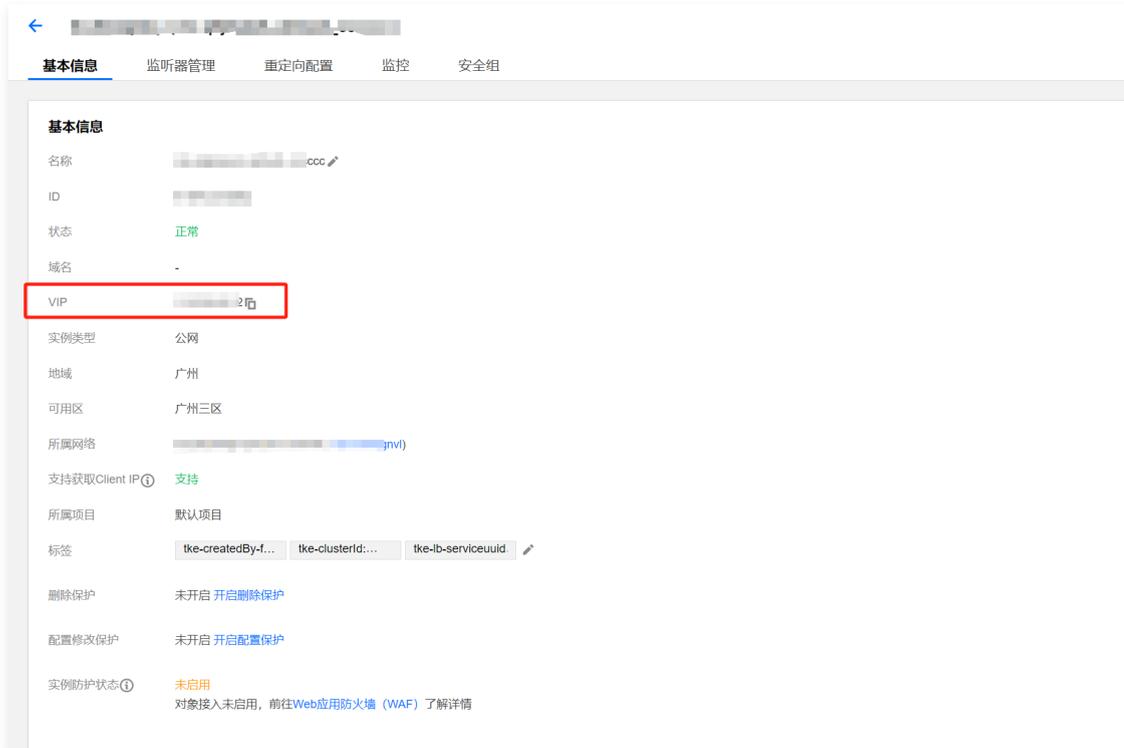
确认
取消

4. 单击监听器名称，查看是否监听正常。



步骤4：测试是否配置成功

1. 查看公网 CLB 的地址，这里假设查看的地址为：192.168.1.1。



2. 查看监听配置端口。如下列端口为：8080。



根据上述两个信息，确定 Prometheus 转发的公网地址为如下新 IP:PORT 地址为：192.168.1.1:8080。

3. 到浏览器或者机器上查看是否可以通过这个 IP 获取 UP 数据。

HTTP API 地址：

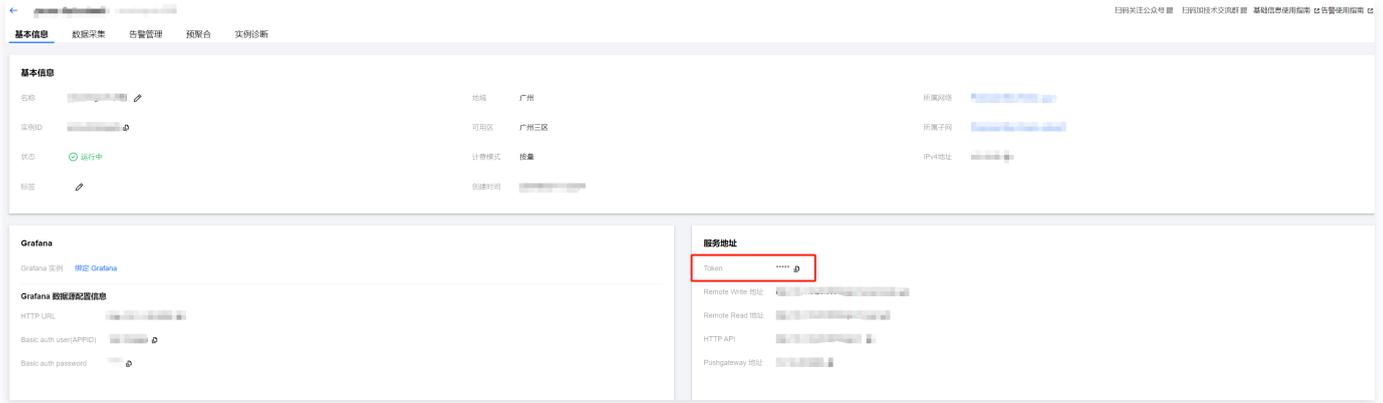
```
http://IP:PORT/api/v1/query?query=up
```

用对应 CLB 的公网 IP 和端口 替换 IP:PORT 后如下：

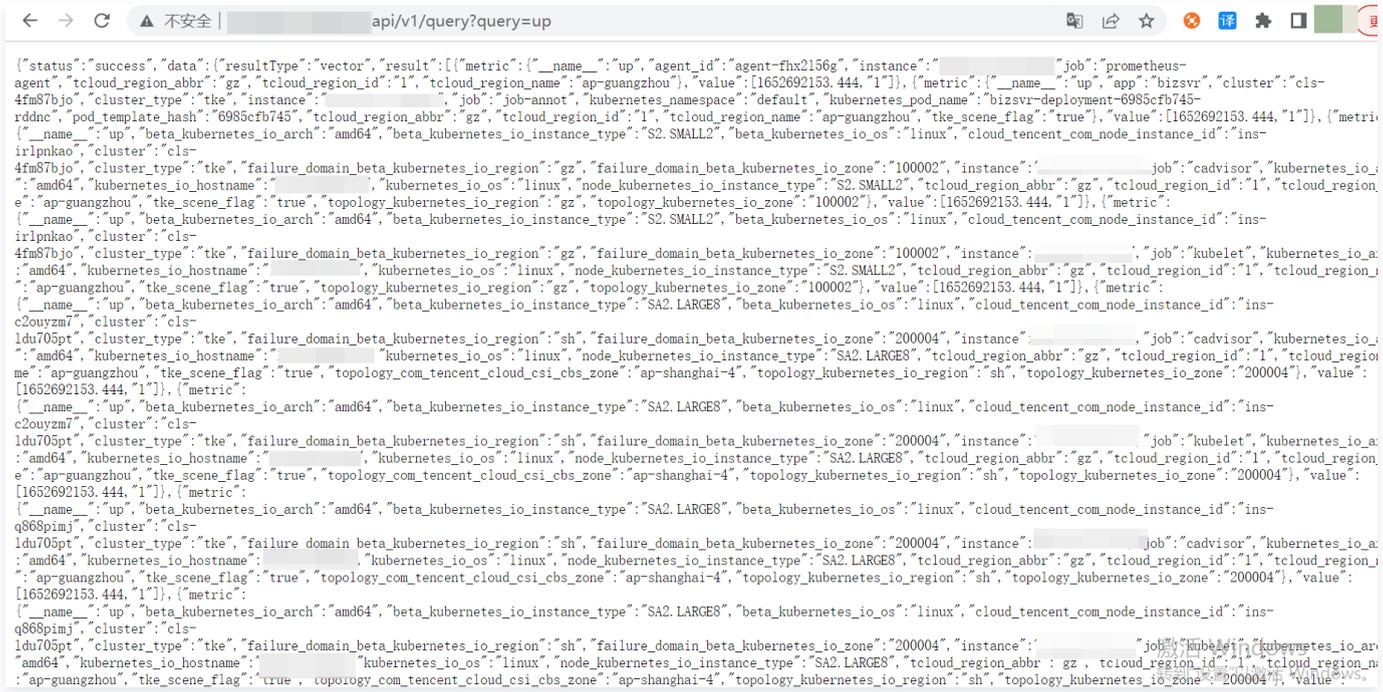
```
http://81.71.21.123:8080/api/v1/query?query=up
```

4. 进入链接地址 `http://81.71.21.123:8080/api/v1/query?query=up`。

- 用户名：填写您的主账号 ID (APPID)。
- 密码：在实例的基本信息页面获取 Token。



如下图所示，Prometheus 配置公网地址成功。



Prometheus 实例访问公网

最近更新时间：2024-11-08 18:03:02

操作背景

在一些场景下，您可能需要您的 Prometheus 实例具备访问公网的能力，例如：

- Prometheus 集成对云上 COS 产品（对象存储）的监控时，由于 COS 产品不支持内网访问，所以需要 Prometheus 实例具备访问公网的能力。
- 使用 Prometheus 告警策略里的 webhook，而该 webhook 为公网地址时。

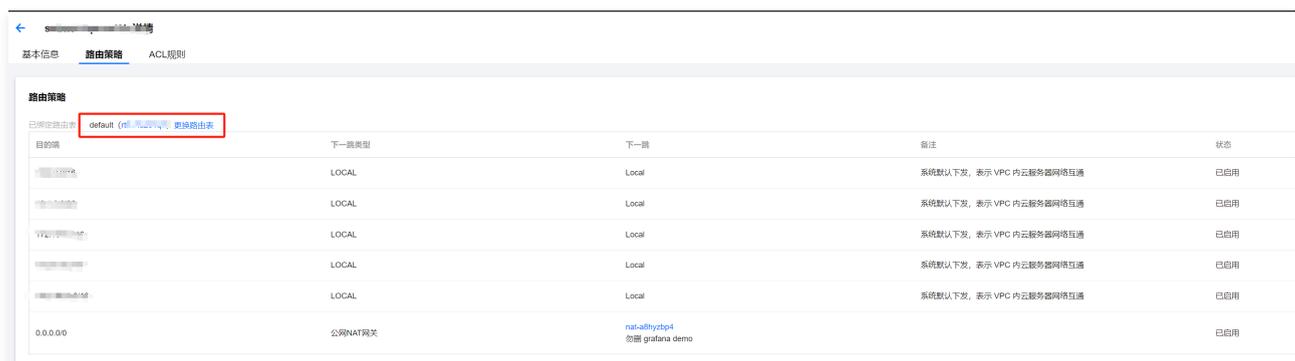
在这类情形下，您需要为 Prometheus 开通公网访问（底层是为 Prometheus 实例组件所在的 TKE 托管集群开通公网访问）。

操作步骤

1. 登录 [Prometheus 监控服务控制台](#)。
2. 单击对应的实例，进入实例管理页，单击基本信息页的**所属子网**。



3. 在子网顶部菜单，切换为**路由策略**栏。单击路由表链接（列表上的 `rtb-xxx`）进入路由表界面。



4. 在路由表界面单击**新增路由策略**。
 - **目的端**：填写 `0.0.0.0/0`。
 - **下一跳类型**：选择 NAT 网关。
 - **下一跳**：选择对应的网关，若无网关，请参见 [NAT 网关](#) 指引进行创建。



5. 填写完成后，单击**创建**。创建成功后表示 TKE Serverless 开通公网访问。

使用实例诊断分析 Prometheus 问题

最近更新时间：2024-10-25 18:18:01

为了提升用户在使用 Prometheus 监控采集功能时的体验，Prometheus 监控团队特推出了新的采集架构，且为用户提供了名为**实例诊断**的控制台页面。此功能有助于用户对当前 Prometheus 实例采集和存储功能的运行情况有更加细致的了解，并在出现问题时可以更加快速地定位和排查。

说明：

若用户名下有存量的 Prometheus 实例，则需要先将其升级到新的采集架构再来使用实例诊断的功能。详情可参考 [实例诊断](#)。

问题一：采集组件升级

现象

实例诊断页面中，提示采集组件有新版本可升级。

The screenshot displays the '实例诊断' (Instance Diagnosis) page for a Prometheus instance named 'cls'. The left sidebar shows resource usage (Pod 数 1/1, 1核 2 G), configuration, and status (4 Up, 1 Agent). The main area features a '采集架构图' (Collection Architecture Diagram) and a '版本' (Version) section. The version section lists components with upgrade prompts: proxy-agent (v1.0.3->v1.0.8), tmp-agent (v1.0.1->v1.1.1), proxy-server (v1.0.3->v1.0.7), and tmp-operator (v1.0.1->v1.1.1). The architecture diagram shows the flow from '采集组件托管集群' (Collection Component Managed Cluster) to 'tmp-operator', 'tmp-agent', and 'proxy-server', which then connects to '用户集群' (User Cluster) via '内网代理' (Intranet Proxy) to reach '集群内采集目标' (Cluster Internal Collection Targets).

解释和处理

Prometheus 采集端主要包含四个组件，用来实现采集目标的发现和调度、采集操作的进行、采集集群和用户集群之间的通信等功能：

- **tmp-operator**：负责采集目标的发现和调度，以及采集分片 tmp-agent 的安装和配置生成，并以集群化的形式管理所有采集分片。
- **tmp-agent**：负责实施实际的采集操作，并将处理后的指标推送到指定的远端地址。
- **proxy-server**：为采集组件访问用户集群提供代理功能，是通过多个端口实现多路代理的七层代理器，支持多个 agent 连接同一个 server。
- **proxy-agent**：与 proxy-server 共同提供完整的代理功能，该组件安装在用户集群中。

可以在**实例诊断**页面查看各个组件的当前版本、功能描述以及不同版本的更新信息。在版本的更新中，会对组件可能存在的问题进行处理修复。因此，用户使用的过程中可以尽量保持组件升级到最新的版本。当实例诊断中有新版本提示时，可根据自身需要对采集组件进行升级。其中 tmp-operator、tmp-agent 和 proxy-agent 的升级不会影响数据采集，即不会产生数据断点；而 proxy-server 的升级则可能会出现数据断点，具体的恢复时间和底层集群 Pod 的调度时间有关，需要谨慎升级。

The screenshot displays the Prometheus monitoring console interface. It is divided into several sections:

- 基本信息 (Basic Information):** Shows resource usage (Pod count, memory), collection configuration, target distribution (4 targets assigned), target status (4 Up), agent status (1 Up), and version information for tmp-agent, proxy-agent, tmp-operator, and proxy-server.
- 采集诊断 (Collection Diagnosis):** Displays a '指标上报总速率' (Metric Reporting Total Rate) of 90.53 and a 'series 数量 Top10 指标' (Top 10 Metrics by Series Count) table.
- tmp-agent (可升级):** Component description, current version (v1.0.1), and latest version (v1.0.7).
- proxy-agent (可升级):** Component description, current version (v1.0.3), and latest version (v1.0.8).
- tmp-operator (可升级):** Component description, current version (v1.0.1), and latest version (v1.0.1).
- proxy-server (可升级):** Component description, current version (v1.0.1), and latest version (v1.0.1).

问题二：网络异常

现象

- 在查询 Prometheus 指标时陆续出现断点；
- 在控制台上显示采集目标一直因为 timeout 而处于 down 的状态；
- 在实例诊断页面则可能会显示 proxy-server 和 proxy-agent 的连接异常。



解释和处理

Prometheus 采集涉及到集群内、跨集群、跨 VPC 的通信，不同对象之间的网络异常可能会导致通信异常，使得表现出来采集功能也存在异常。



这种情况，通常是带宽问题或安全组问题，特别是带宽问题。当链路上某一节点的带宽占满之后，请求或响应在该节点被阻塞，最终导致请求或响应超时。

1. 在跨 VPC 采集的情况下，如果使用的是公网 CLB，那么可以优先排查公网 CLB 的链路是否通畅、带宽是否占满；如果使用的是云联网连接两个集群，那么可以优先排查云联网的链路是否通畅、带宽是否占满。



2. 在排除跨 VPC 通信的问题后，可以对采集组件 proxy-server 的带宽是否占满进行排查（进入采集集群，并在 Pod 中查看监控即可）。



3. 如果已排除带宽的问题，那么可能是安全组的问题。安全组未放行一般在集群关联时就会暴露出来，但不排除后续被修改的可能，被安全组阻挡的请求就无法到达目标 Pod，需要按照实例诊断中的安全组要求进行配置。



问题三：数据多写堆积

现象

- 用户短信、站内信等收到数据多写配置异常的告警；
- 在实例诊断页面显示数据多写异常堆积。

[扫码关注公众号](#) [扫码加](#)

实例诊断

当前实例地址: 写入正常

数据多写地址:
http://ip:9090/api/v1/prom/write: 写入堆积

当前实例 10.0.0.17:9090 (2/2)

采集架构图

采集组件托管集群

可用 IP 178 个
安全组 sg
Pod 数 19/20 (11 核 20.25 G)

分配 Target → tmp-operator-544cc74c8d-grsnl (运行中)

采集 → tmp-agent tke-cl... 1/1 (运行中)

采集 → proxy-server 10.0.0.91 (运行中)

远程写入
411.20/s

内网代理

采集

用户集群

tke/cls

proxy-agent 2/2 (运行中)

集群内采集目标

[集群采集组件说明](#)

解释和处理

数据多写功能支持将集成容器服务和集成中心的数据写到自建 Prometheus 或其他 Prometheus 监控实例中。如果多写异常，说明采集分片在向目标地址推送指标的时候出现了问题；采集分片如果推送失败，会将指标保存在本地进行重试，如果数据堆积量过大，还可能导致 Pod 异常影响指标采集。

当出现这样的问题，可参考如下步骤排查：

1. 在实例诊断中确认具体的目标地址，短信告警中没有具体的地址；
2. 查看对应的 Prometheus 是否可以正常写入、是否有异常的日志，以排除目标 Prometheus 的问题；
3. 可以查看采集分片的日志，是否存在报错，以及分析具体报错的信息。

The screenshot shows the Prometheus monitoring console interface. On the left, there are sections for '采集诊断' (Collection Diagnosis) and 'Agent状态' (Agent Status). The 'Agent状态' section shows '1个' (1) agent is up. In the center, there is a '采集架构图' (Collection Architecture Diagram) showing the 'tmp-operator-7cfs' cluster. On the right, there is a table titled 'Agent状态' with columns: Pod, 空闲时间 (Idle Time), 当前处理量 (Current Processing Volume), CPU/Memory, 创建时间 (Creation Time), and 操作 (Action). A red arrow points to the '操作' column, which contains a '查看日志' (View Logs) button. Below the table, there is a '存储诊断' (Storage Diagnosis) section.

The screenshot shows the '日志' (Logs) view for the 'tmp-agent' container. The 'Container' dropdown is set to 'tmp-agent'. Below it, there are buttons for log counts: 50, 100, 200, 500, 1000, and 2000. A toggle switch is set to '查看已退出的容器' (View exited containers). At the bottom right, there is an '自动刷新' (Auto Refresh) toggle. The log content is displayed in a dark box with a red border, showing three entries:

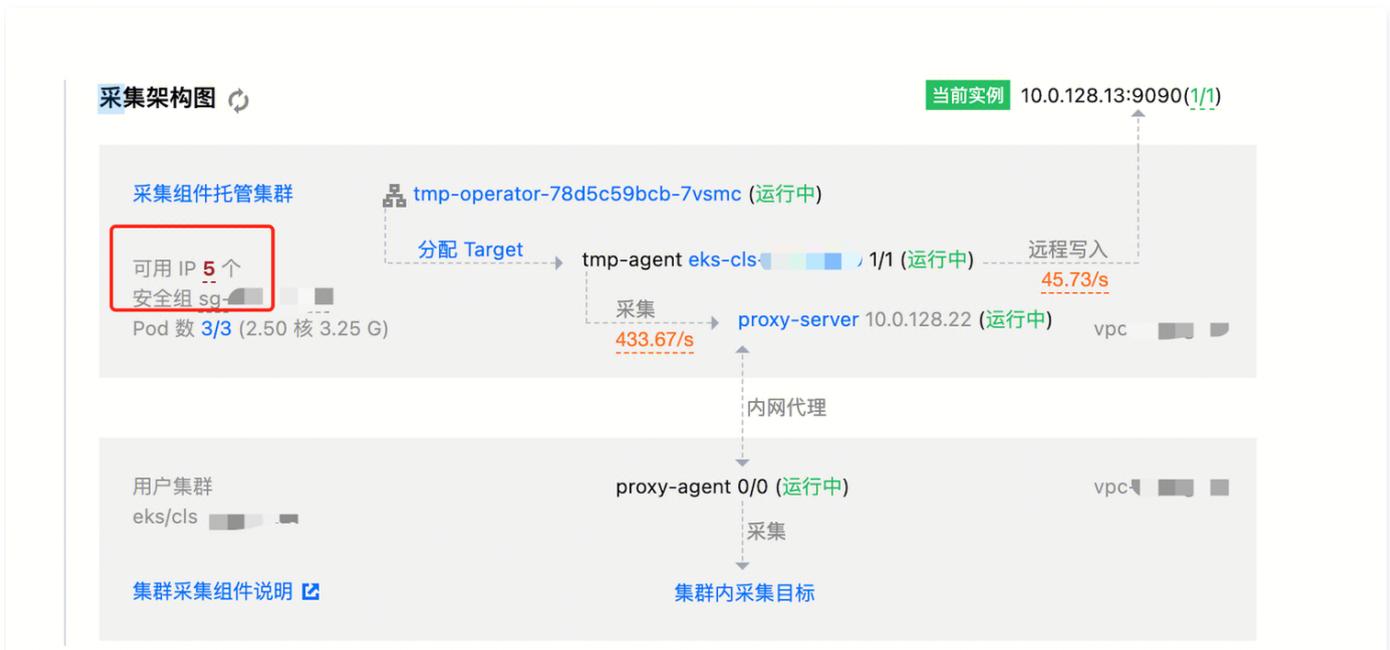
```

1 2024-06-20T03:00:54.452169901Z 2024-06-20T11:00:54.452+0800 warn /go/pkg/mod/github.com/victoria!metrics/!victoria!metrics@v1.92.1/app/vmagent/remotewrite/client.go:370 couldn't send a block with size 11270 bytes to "2:secret-url": Post "http://ip:9090/api/v1/prom/write": dial tcp4: lookup ip on 172.16.192.200:53: no such host; re-sending the block in 60.000 seconds
2 2024-06-20T03:00:54.452193292Z 2024-06-20T11:00:54.452+0800 warn /go/pkg/mod/github.com/victoria!metrics/!victoria!metrics@v1.92.1/app/vmagent/remotewrite/client.go:370 couldn't send a block with size 3393 bytes to "2:secret-url": Post "http://ip:9090/api/v1/prom/write": dial tcp4: lookup ip on 172.16.192.200:53: no such host; re-sending the block in 60.000 seconds
3 2024-06-20T03:01:54.455461582Z 2024-06-20T11:01:54.455+0800 warn /go/pkg/mod/github.com/victoria!metrics/!victoria!metrics@v1.92.1/app/vmagent/remotewrite/client.go:370 couldn't send a block with size 3393 bytes to "2:secret-url": Post "http://ip:9090/api/v1/prom/write": dial tcp4: lookup ip on 172.16.192.200:53: no such host; re-sending the block in 60.000 seconds
    
```

问题四：可用 IP 数量不足

现象

实例诊断页面提示 IP 数量不足（红色并闪烁）。



解释和处理

采集组件托管集群所关联的子网可用 IP 数量不足，可能会导致如下问题：

- 当采集规模不断增大，达到单个采集分片无法承受时，调度器会对采集群片进行扩容，扩容出来的新分片 Pod 会占用子网内部的 IP；如果可用 IP 数量不足，会出现无法扩容的情况，已有的采集分片持续超负荷运转，会影响采集的稳定性。
- 当进行采集组件升级时，部分组件会先产生新的正常运行的副本，再替换旧的副本；如果可用 IP 数量不足，会出现新的副本无法调度成功，导致组件升级异常。

因此，建议始终保持一定数量的可用 IP。当出现上述实例诊断页面显示可用 IP 数量较少时，可以在页面中直接添加子网，具体操作如下：

1. 点击子网数字，进入**托管集群子网管理**页面；
2. 显示当前 VPC 内已添加到托管集群的子网和未添加的子网，可根据规划点击启用子网启用新的子网；
3. 若没有可用子网，可以先 **添加子网** 后再在当前页面中启用。

诊断

采集架构图

采集组件托管集群

可用 IP 5 个
安全组 sg-
Pod 数 3/3 (2.50 核 3.25 G)

tmp-operator-78d5

分配 Target

tmp-agent eks-cl...

远程写入 45.73/s

采集 433.67/s

proxy-server 10.0.128.22 (运行中)

内网代理

proxy-agent 0/0 (运行中)

采集

集群内采集目标

集群采集组件说明

托管集群子网管理 vpc

• 当前剩余IP数目为: 5

• 当前已启用子网剩余IP数目过少，可能会导致扩容失败或集成安装失败，请启用新子网

子网 ID	IP 范围	剩余 IP 数量/IP 总数	是否启用	操作
subnet-		0/5 0.00%	已启用	-
subnet-		0/5 0.00%	已启用	-
subnet-		5/13 38.46%	已启用	-
subnet-		1015/1021 99.41%	未启用	启用子网
subnet-		209/253 82.61%	未启用	启用子网
subnet-		111/125 88.80%	未启用	启用子网
subnet-		123/125 98.40%	未启用	启用子网

问题五：指标标签数量过大导致写入异常

现象

- 存储诊断中，指标最大 Label 数量显示异常速率；
- 同时，Prometheus 实例基本信息页面的实例监控中，“因不合法被丢弃的 Samples”图标中显示有数据被丢弃；
- 在查询时有部分预期的指标查不出来。

存储诊断 🔄

指标上报总速率①

90.53

收费指标速率 9.11
免费指标速率 81.42

实例 series 存储上限

单指标 series 存储上限

指标 Label 名称的最大长度

指标 Label 值的最大长度

指标最大 Label 数量 34
异常速率22.56

指标时间戳允许的最老范围

指标时间戳允许的最新范围

单次查询的最大 series 数量

单位时间告警数量上限①

单位时间告警字节大小上限①

series 数量 Top10 指标	数量
container_fs_writes_bytes_total	247
container_fs_reads_bytes_total	247
kube_pod_status_phase	115
container_memory_working_set_bytes	88
container_cpu_usage_seconds_total	71
container_fs_limit_bytes	66
container_fs_usage_bytes	66
container_network_receive_bytes_total	16
container_network_receive_packets_total	11
kube_pod_container_status_restarts_total	10

解释和处理

指标最大 Label 数量代表写入指标标签的最大数量 **限制**，如果使用过程中，在指标中加入大量无用标签，就可能会超过限制导致需要的指标写入失败，出现上述的异常提示。

在存储诊断中点击**异常速率**弹出窗口时，可以看到对应目标集群内尝试写入存储的**最大 Label 数量**，以及对应不同采集任务 Job 的详细信息。

采集诊断 cls-

资源占用 Pod 数 1/1 (1核 2 G)

采集配置 已分配 4 个 未分配 0 个

Target 分配情况 已分配 4 个 未分配 0 个

Target 状态 4 Up

Agent 状态 1 个

版本 tmp-agent(v1.0.1->v1.0.7) proxy-agent(v1.0.3->v1.0.8) tmp-operator(v1.0.1->v1.1.0) proxy-server(v1.0.3->v1.0.7)

Label 数量超限指标 最大数量34

Cluster	最大 Label 数量
cls-	46

Job 名称	最大 Label 数量	超限指标详情	采集配置	元数据
kube-system/kube-state-metrics	46	查看详情	配置	元数据

存储诊断

指标上报总速率 90.53

收费指标速率 9.11

免费指标速率 81.42

实例 series 存储上限

单指标 series 存储上限

指标 Label 名称的最大长度

指标 Label 值的最大长度

指标最大 Label 数量 34 异常速率 22.56

指标时间戳允许的最老范围

指标时间戳允许的最新范围

单次查询的最大 series 数量

单位时间告警数量上限

单位时间告警字节大小上限

series 数量 Top10 指标	数量
container_fs_writes_bytes_total	247
container_fs_reads_bytes_total	247
kube_pod_status_phase	115
container_memory_working_set_bytes	88
container_cpu_usage_seconds_total	71
container_fs_limit_bytes	66
container_fs_usage_bytes	66
container_network_receive_bytes_total	16
container_network_receive_packets_total	11
kube_pod_container_status_restarts_total	10

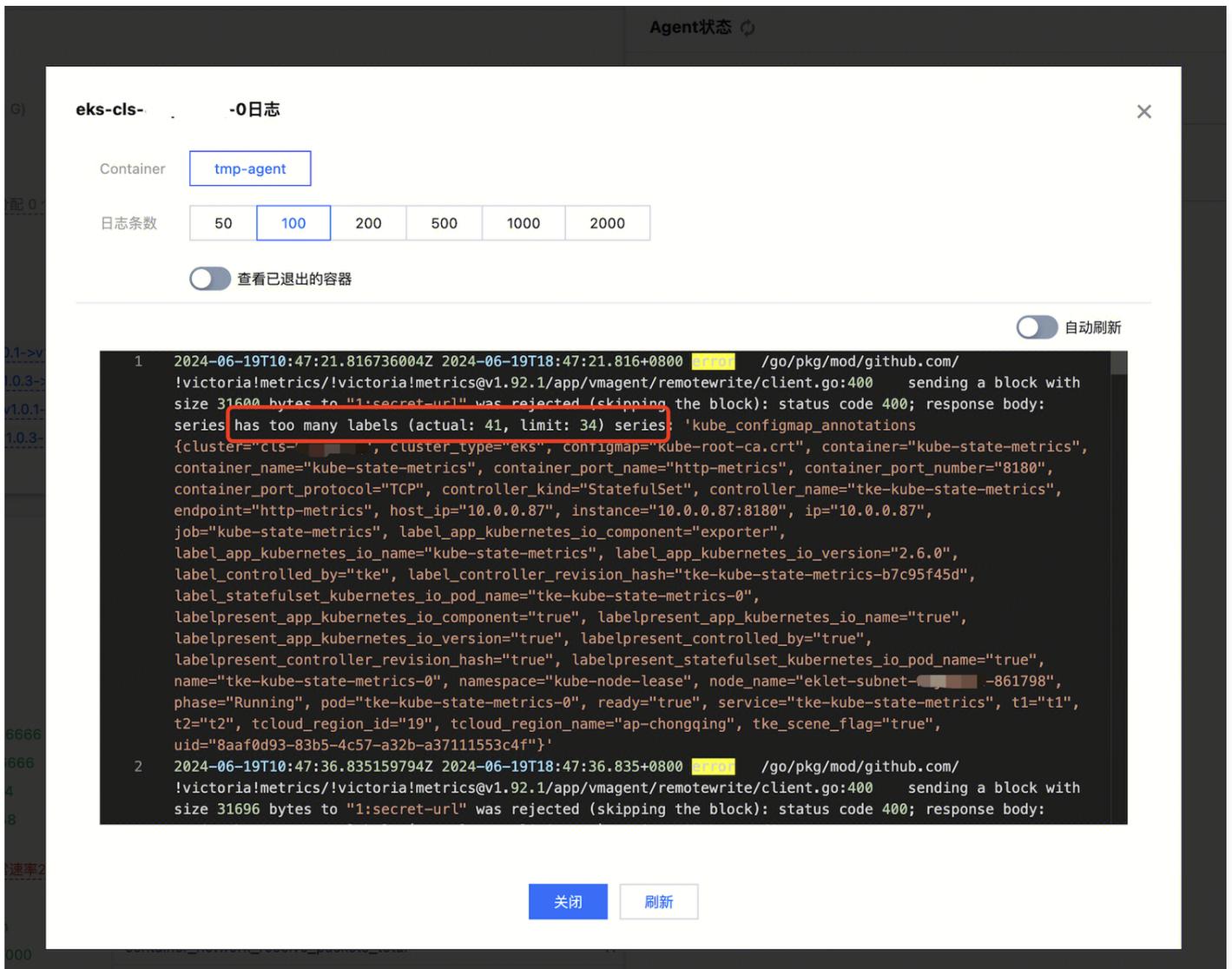
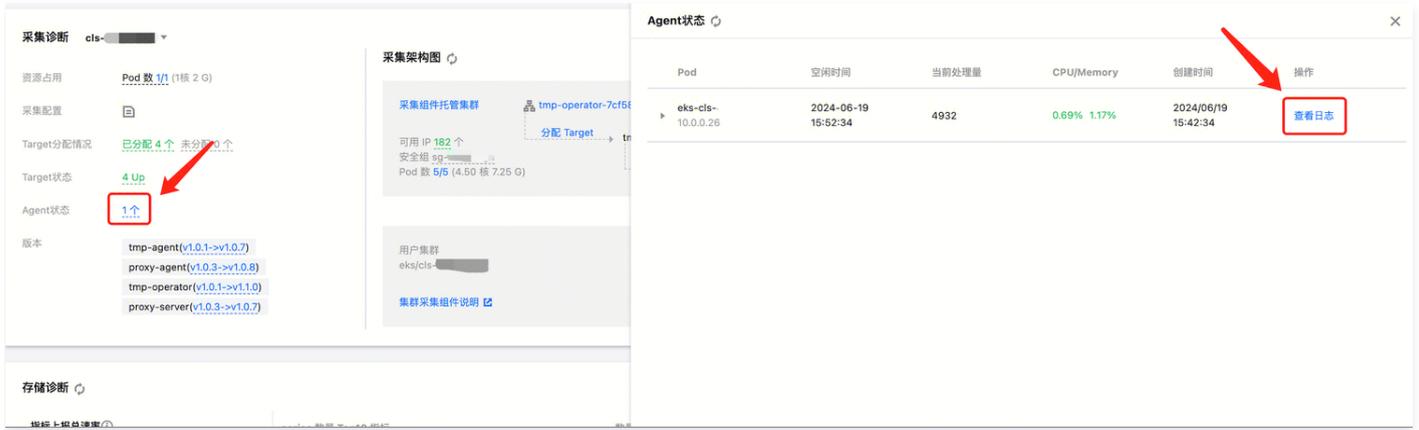
在 Label 数量超限指标详情中，可以看到对应指标的总 Label 数量，以及采集 Label 数、全局标记和存储附加 Label 的信息。

Label 数量超限指标详情

指标名	受限 Label 数	采集 Label 数	全局标记	存储附加 Label	指标采集速率
kube_configmap_annotations	41	34	tke_scene_flag: true cluster:cls-6zq cluster_type:eks t1:t1 t2:t2	tcloud_region_id:19 tcloud_region_name:ap-chongqing	1.13个/秒
kube_configmap_info	41	34	tke_scene_flag: true cluster:cls-6zq cluster_type:eks	tcloud_region_id:19 tcloud_region_name:ap-	1.13个/秒

- 采集 Label 数:** 根据采集配置，对采集目标服务发现的标签进行 relabel 之后的标签，加上指标本身的标签之后的标签数量，不包括全局标记以及存储附加标签。
- 全局标记:** 用户额外配置的，在当前集群所有采集任务上都会添加的标签；默认情况下会在关联集群的时候，添加包含集群信息的“cluster_type”和“cluster”全局标记；
- 存储附加 Label:** 由存储侧默认写入的标签，无法修改，用于表示存储本身的信息。

因此，要解决指标标签超限的问题，可以从采集 Label 数和全局标记两个方面处理。处理之前，可以通过查看日志了解被丢弃的指标在写入之前包含的标签信息，在采集分片的日志中进行查看，并确认需要丢弃哪些标签。



- 对于采集 Label 数，可以在采集配置中的 `metric_relabel_configs` 或 `metricRelabelings` 中增加 `labeldrop` 配置来丢弃掉。

```
metricRelabelings:
- action: labeldrop
  regex: label_(.+)
```

正则匹配需要去掉的标签

- 对于全局标记，可以在控制台的数据采集中进行编辑，有不需要的全局标记可以在这里去掉，不过需要注意的是，全局标记的变更会影响到整个集群的采集。



问题六：指标数量达到存储上限导致写入异常

现象

- 存储诊断中，“实例 series 存储上限”或“单指标 series 存储上限”显示“异常速率”。
- 同时，Prometheus 实例基本信息页面的实例监控中，“因限流丢弃的 samples”图表中显示有 Samples 被丢弃；在查询时有部分预期的指标查不出来。

解释和处理

在存储诊断中，series 存储上限分别代表了整个 Prometheus 实例和单个指标的 series 存储 **上限**，当存储中的 series 达到上限之后，会出现新的 series 无法写入的问题，详情如下。（图中的存储上限仅供演示）

存储诊断 🔄

指标上报总速率 6.92

收费指标速率 0.66
免费指标速率 6.26

实例 series 存储上限 160
异常速率21.60

单指标 series 存储上限 16
异常速率3.60

指标 Label 名称的最大长度 1024

指标 Label 值的最大长度 2048

指标最大 Label 数量

指标时间戳允许的最老范围 5h

指标时间戳允许的最新范围 10m

单次查询的最大 series 数量 100000

单位时间告警数量上限 2000

单位时间告警字节大小上限 20971520

series 数量 Top10 指标	数量
container_fs_writes_bytes_total	16
container_fs_reads_bytes_total	16
container_fs_limit_bytes	16
container_memory_working_set_bytes	16
container_cpu_usage_seconds_total	15
container_fs_usage_bytes	13
container_network_receive_bytes_total	7
container_network_receive_packets_dropped_total	7
container_network_receive_packets_total	7
container_network_transmit_packets_total	7

单指标存储上限

通常，单个指标 series 数量庞大，是因为某些标签值变化过于频繁，例如将用户的 id、时间戳等作为标签值，或许这在日志集中很常见，但对于 Prometheus 来说，这种使用方式造成的高基数问题，会严重影响 Prometheus 系统的性能。

解决该问题，即解决标签值频繁变化的情况。在存储诊断中，可以看到“series 数量 Top10 指标”，当出现异常的时候，这里的数值和“单指标 series 存储上限”是对应的。

存储诊断

指标上报总速率 

6.92

收费指标速率 0.66
免费指标速率 6.26

实例 series 存储上限 **160**
异常速率**21.60**

单指标 series 存储上限 **16**
异常速率**3.60**

指标 Label 名称的最大长度 **1024**

指标 Label 值的最大长度 **2048**

指标最大 Label 数量

指标时间戳允许的最老范围 **5h**

指标时间戳允许的最新版范围 **10m**

单次查询的最大 series 数量 **100000**

单位时间告警数量上限  **2000**

单位时间告警字节大小上限  **20971520**

series 数量 Top10 指标

数量

container_fs_writes_bytes_total	16
container_fs_reads_bytes_total	16
container_fs_limit_bytes	16
container_memory_working_set_bytes	16
container_cpu_usage_seconds_total	15
container_fs_usage_bytes	13
container_network_receive_bytes_total	7
container_network_receive_packets_dropped_total	7
container_network_receive_packets_total	7
container_network_transmit_packets_total	7

只要找到了对应指标，看是具体哪个标签有问题，在业务暴露指标中去掉这个标签，或是在采集配置的 `metric_relabel_configs`（原生 Job 配置）或 `metricRelabelings`（`servicemonitor`、`podmonitor` 等 crd，以及集成中心的云监控集成）将该标签 `labeldrop` 即可。

```
metricRelabelings:
- action: labeldrop
  regex: label_(.+) # 正则匹配需要去掉的标签
metric_relabel_configs:
- action: labeldrop
  regex: label_(.+) # 正则匹配需要去掉的标签
```

实例存储上限

在该处显示异常，说明预期写入的 series 数量超过了单实例可以存储的数量；而绝大部分情况下，该限制已经满足普通业务量的需要了。您可从如下方面考虑：

1. 确认是否因为某些标签值变化过于频繁导致的整体存储 series 数量过多，毕竟 series 数量和费用关联紧密，如果是因为不需要的标签值变化导致 series 数量较大，最好将该标签去除；
2. 根据业务需要，去掉不需要写入的 series，精简指标体系；
3. 如果业务真的需要上报如此大量的 series，请 [联系我们](#)。

问题七：关联失败

现象

在 [Prometheus 控制台](#) > [点击实例 ID/名称](#) > [数据采集](#) > [集成容器服务](#) 中关联集群时，偶尔会出现“状态异常”关联失败的情况；在实例诊断中也会显示“采集组件关联异常”，这个异常出现的时候采集功能是没有正常发挥作用的。

The screenshot shows the Prometheus monitoring service interface. At the top, there is a navigation bar with five tabs: "基本信息", "数据采集", "告警管理", "预聚合", and "实例诊断". The "实例诊断" tab is currently selected. Below the navigation bar, there is a section titled "采集诊断" with a dropdown menu showing "cls". A red box highlights a warning message: "采集组件关联异常, 查看绑定状态 刷新". Below this, there is a section titled "存储诊断" with a refresh icon.

解释和处理

关联成功是正常使用采集容器指标的必要条件，出现关联失败问题的暴露时间点位于实际采集和存储发生之前。您可在[集成容器服务](#)或[实例诊断](#)中，查看整个关联过程的详情，包含具体出现异常的环节。

绑定状态 ✕				
任务	状态	启动时间	完成时间	详情
boundClusterEnvPrepare	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
initializeEKSCluster	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
waitClusterRunning	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
用户集群安装 CRD	完成	2024-06-14 16:16:23	2024-06-14 16:16:25	N/A
用户集群安装 RBAC	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
安装 proxy-server 服务	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
安装预聚合规则	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
kvass-operator 扩容	完成	2024-06-14 16:16:23	2024-06-14 16:16:23	N/A
用户集群安装 proxy-agent	完成	2024-06-14 16:16:23	2024-06-14 16:17:14	N/A
安装基础采集配置	完成	2024-06-14 16:16:25	2024-06-14 16:16:27	N/A
安装 tmp-agent CR	异常	2024-06-14 16:17:14		{Reason:get resourceInformer failed,Object:TMPAgent/prom-jtl3hj3g/eks-cls- ,Message:failed to sync cache for ServiceMonitor informer}: context deadline exceeded

[用户集群内安装组件说明](#)

确定
刷新

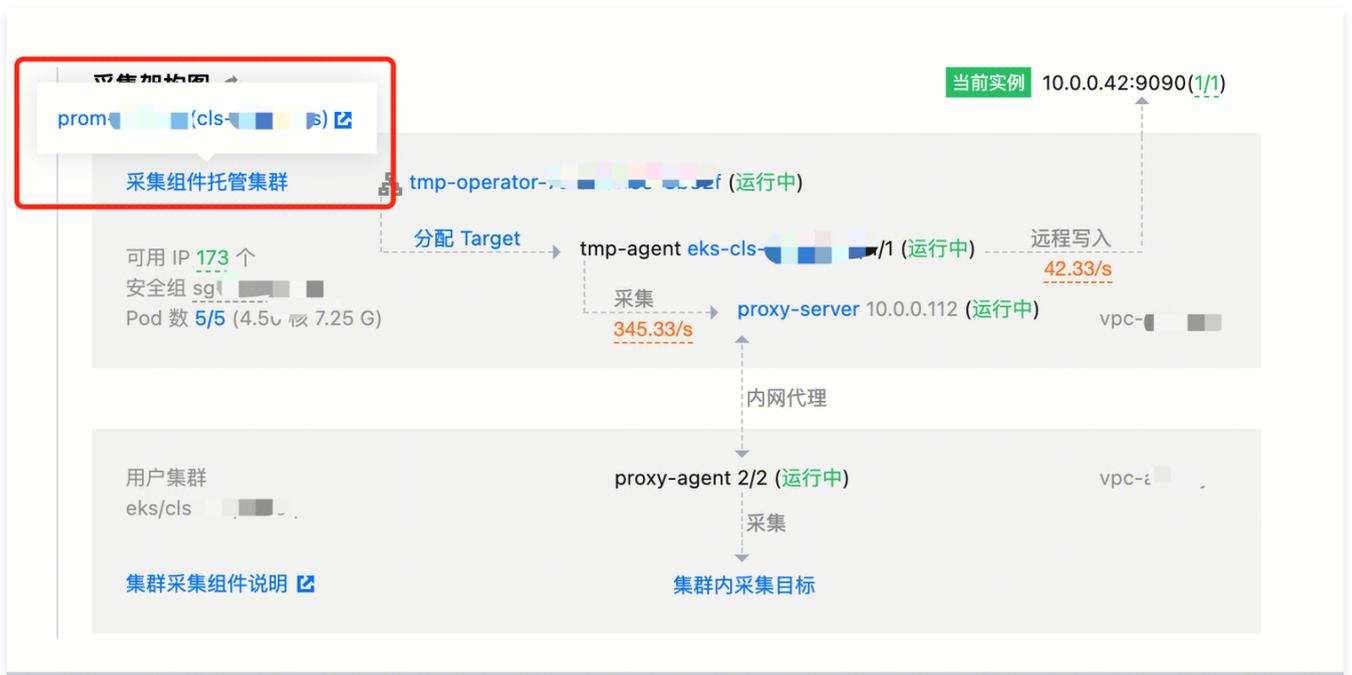
常见的异常及原因通常包括：

● 集群内剩余可用 IP 不足

- 这在异常消息中通常会以以下形式出现：
- “network-unavailable” 的 taint 提示，是 IP 不足时 EKS 集群自动给 node 添加的标记；
- “insufficient IP available” 提示可用 IP 不足；
- 其他与 IP 不足相关的提示。

解决方法：

- 如果出现异常的是 initializeEKSCluster 或安装 proxy-server 阶段，需要给采集集群（即同地域下名称为 Prom 实例 ID 的 Serverless 集群）添加有足够 IP 的子网对应的超级节点。



如果出现异常的是安装 proxy-agent 阶段，则需要给被关联的用户集群添加可用的 IP。

● 集群内资源不足：

- 提示 “nodes are available”，没有足够的 CPU 或内存；
- 提示 “max node group size reached”，节点池无法再扩充节点。

解决方法：这通常发生在目标集群内，需要您增加集群内资源后重试关联。

● 网络不通：

一般异常消息会包含 “failed to sync cache”，由于网络问题导致采集集群无法 watch 目标集群。

解决方法：可进入目标集群以当前 Prom 实例 ID 为名称的命名空间下，查看 proxy-agent 的日志，通常会有无法和对端连接的日志，如 “connect to server failed”。

如果是同 VPC，可能是 proxy-agent 的安全组限制，具体放通要求可以参考如下：

- 入站放通 TCP：9093,9090,10901,10902,9990,3000,8080,8008,8100-8200。
- 出站放通 TCP：ALL



如果是跨 VPC，则需要查看公网 CLB 是否正确配置。例如：如果集群是 EKS 集群是否配置 NAT 网关，或是对应的云联网是否正常等。

