TencentCloud Managed Service for Prometheus

Practical Tutorial

Product Documentation

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Practical Tutorial

Migration from Self-Built Prometheus

Last updated：2024-01-29 15:55:07

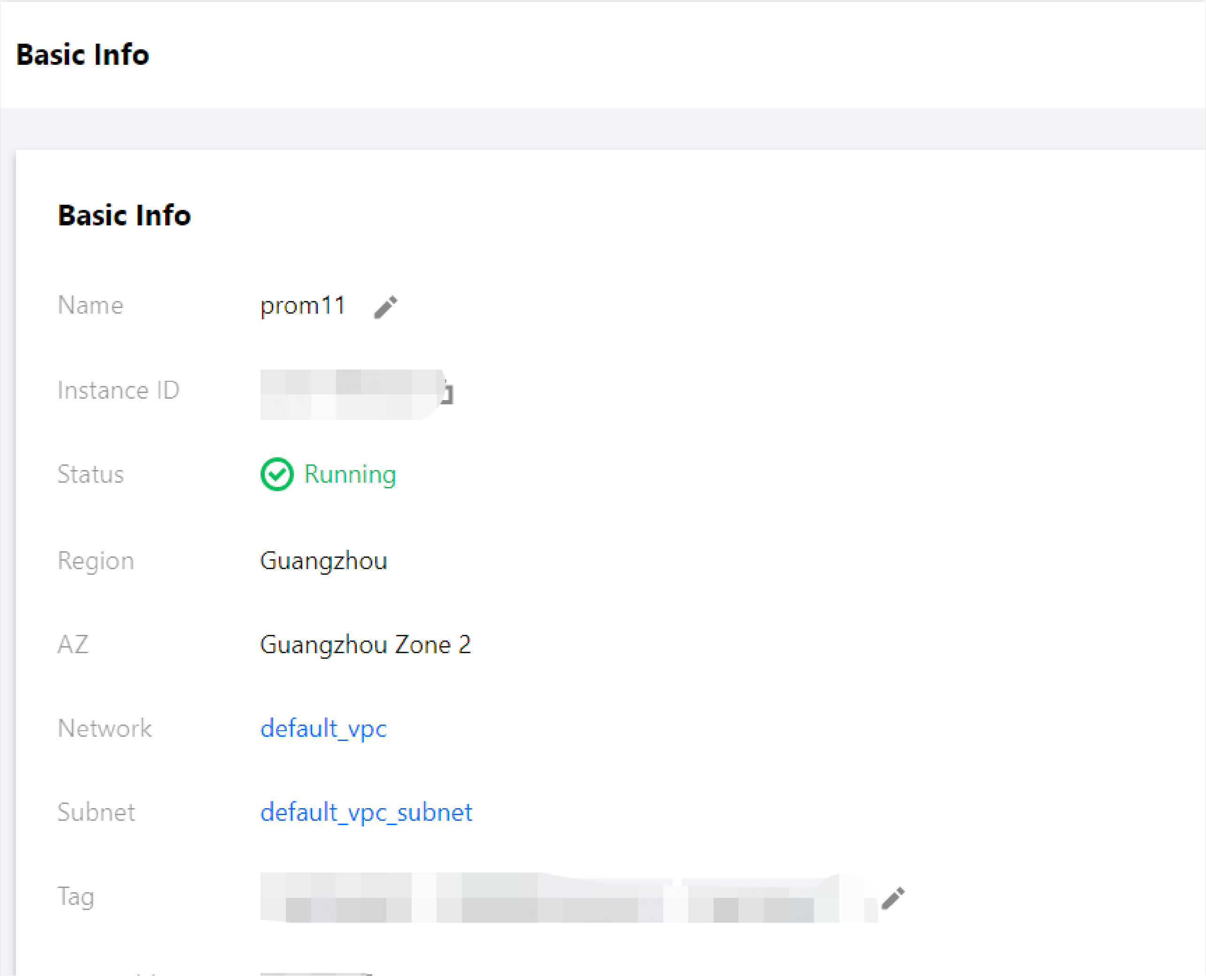
Overview

You can quickly migrate from your self-built Prometheus service to TMP.

Directions

Prometheus itself supports remote write to an external storage; therefore, you can add a remote write configuration pointing to TMP in the configuration file of your self-built Prometheus. The steps are as follows:

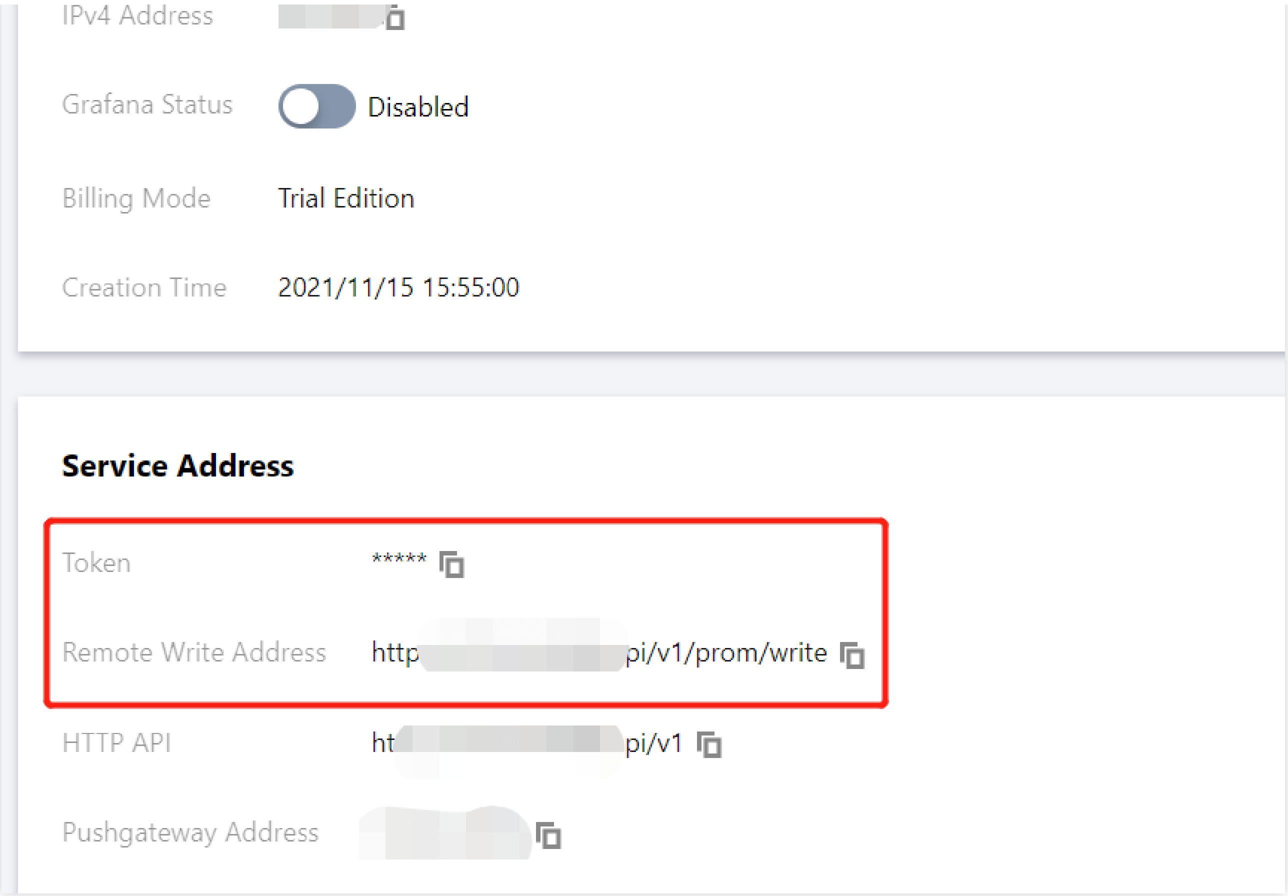
1. Get the remote write address and token of TMP through the basic information of the instance as follows:



Basic Info

Basic Info

Name	prom11
Instance ID	[redacted]
Status	Running
Region	Guangzhou
AZ	Guangzhou Zone 2
Network	default_vpc
Subnet	default_vpc_subnet
Tag	[redacted]



IPv4 Address [redacted]

Grafana Status Disabled

Billing Mode Trial Edition

Creation Time 2021/11/15 15:55:00

Service Address

Token *****

Remote Write Address http[redacted]pi/v1/prom/write

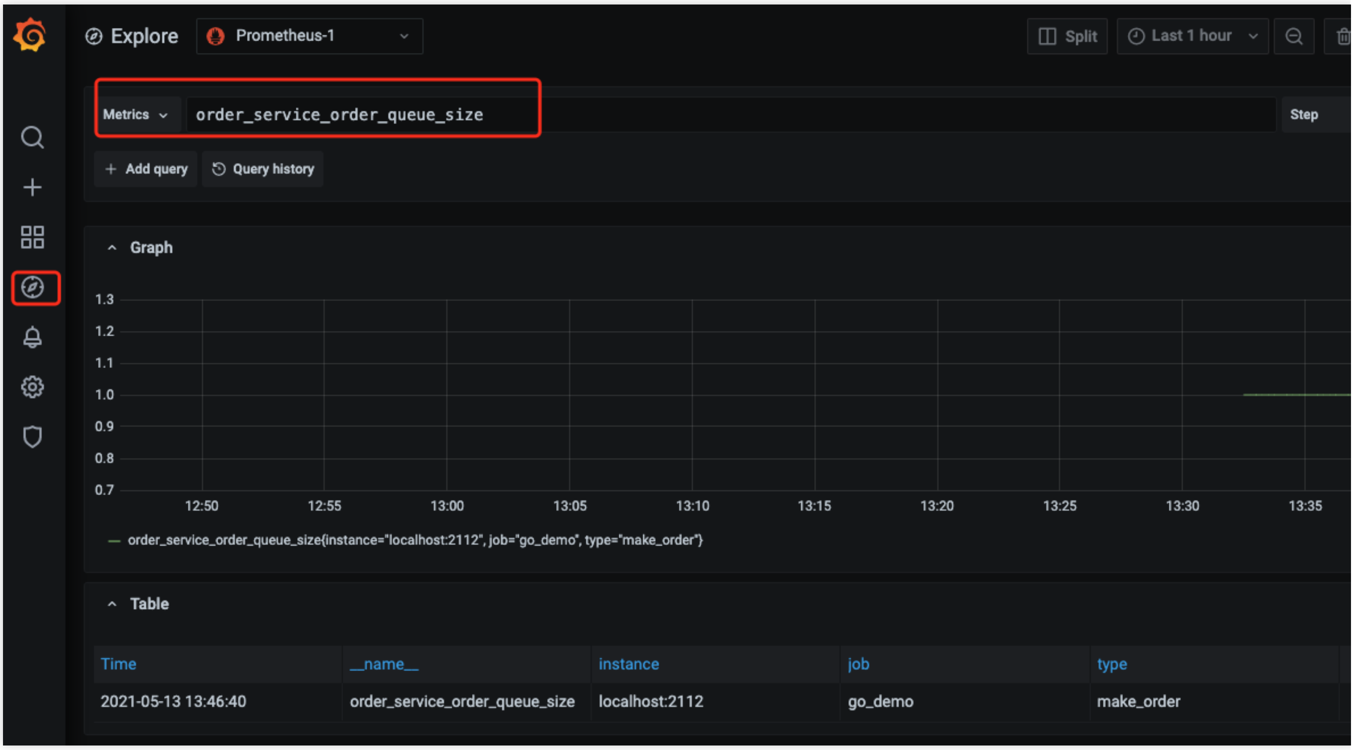
HTTP API ht[redacted]pi/v1

Pushgateway Address [redacted]

2. Modify `prometheus.yml` and restart Prometheus. The specific configuration is as follows. For more information on the remote write configuration parameters, please see remote_write.

```
remote_write:

  - name: cm_prometheus # Remote write name
    url: http://ip:port/api/v1/prom/write  # Get the remote write address from the
    remote_timeout: 30s # Set according to the actual situation
    bearer_token: k32*****trR # Get the token information from the basic informatio
```

3. Open the Grafana service that comes with TMP and use `Explore` to verify whether the data is written successfully as shown below. You can also customize Grafana monitoring dashboards.



```
order_service_order_queue_size
```

order_service_order_queue_size{instance="localhost:2112", job="go_demo", type="make_order"}

Time	__name__	instance	job	type
2021-05-13 13:46:40	order_service_order_queue_size	localhost:2112	go_demo	make_order

4. You can also use Prometheus APIs for self-built visualization. For more information, please see Monitoring Data Query.

Custom Integration with CVM

Last updated：2024-01-29 15:55:07

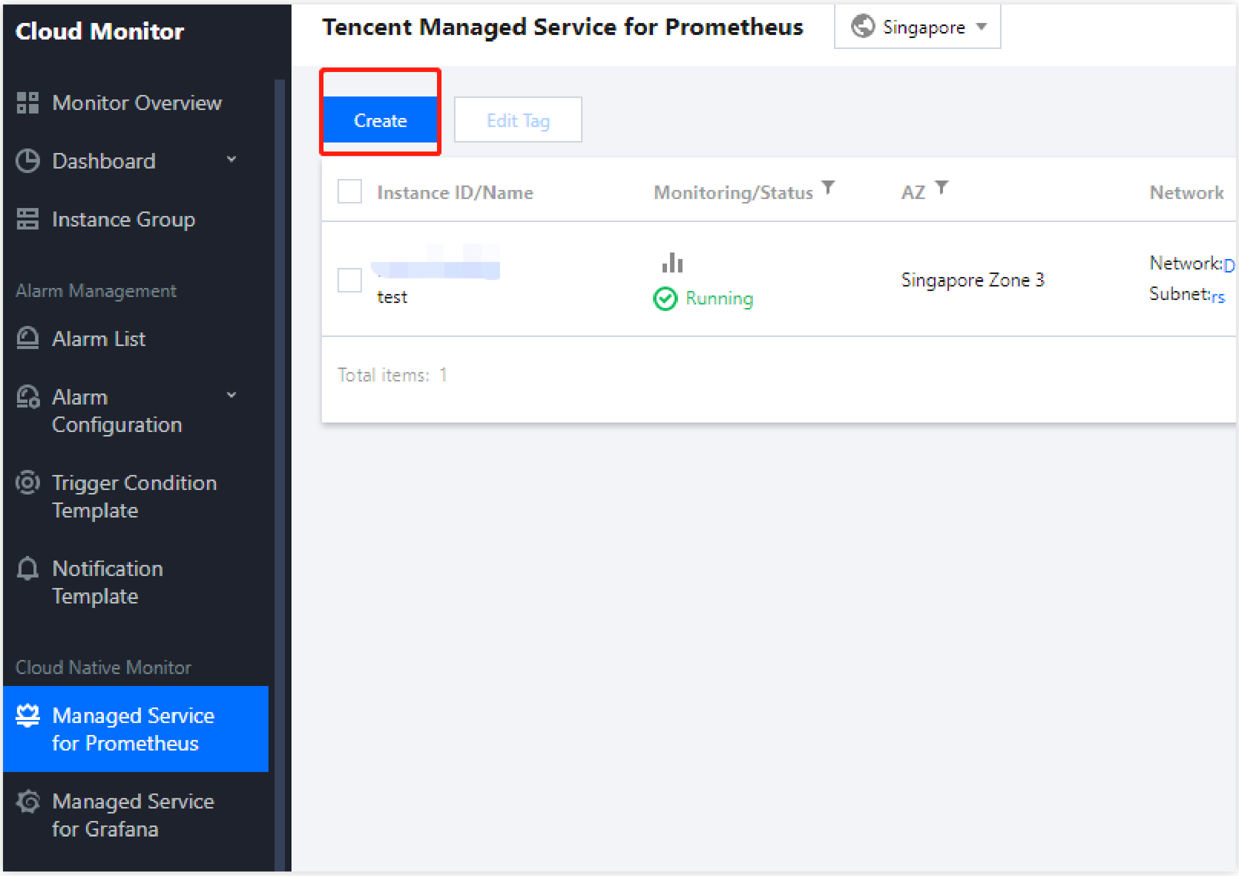
This document describes how to integrate CVM with TMP.

Purchasing a TMP Instance

Note:

The purchased TMP instance must be in the same VPC as the monitored CVM instance for network connectivity.

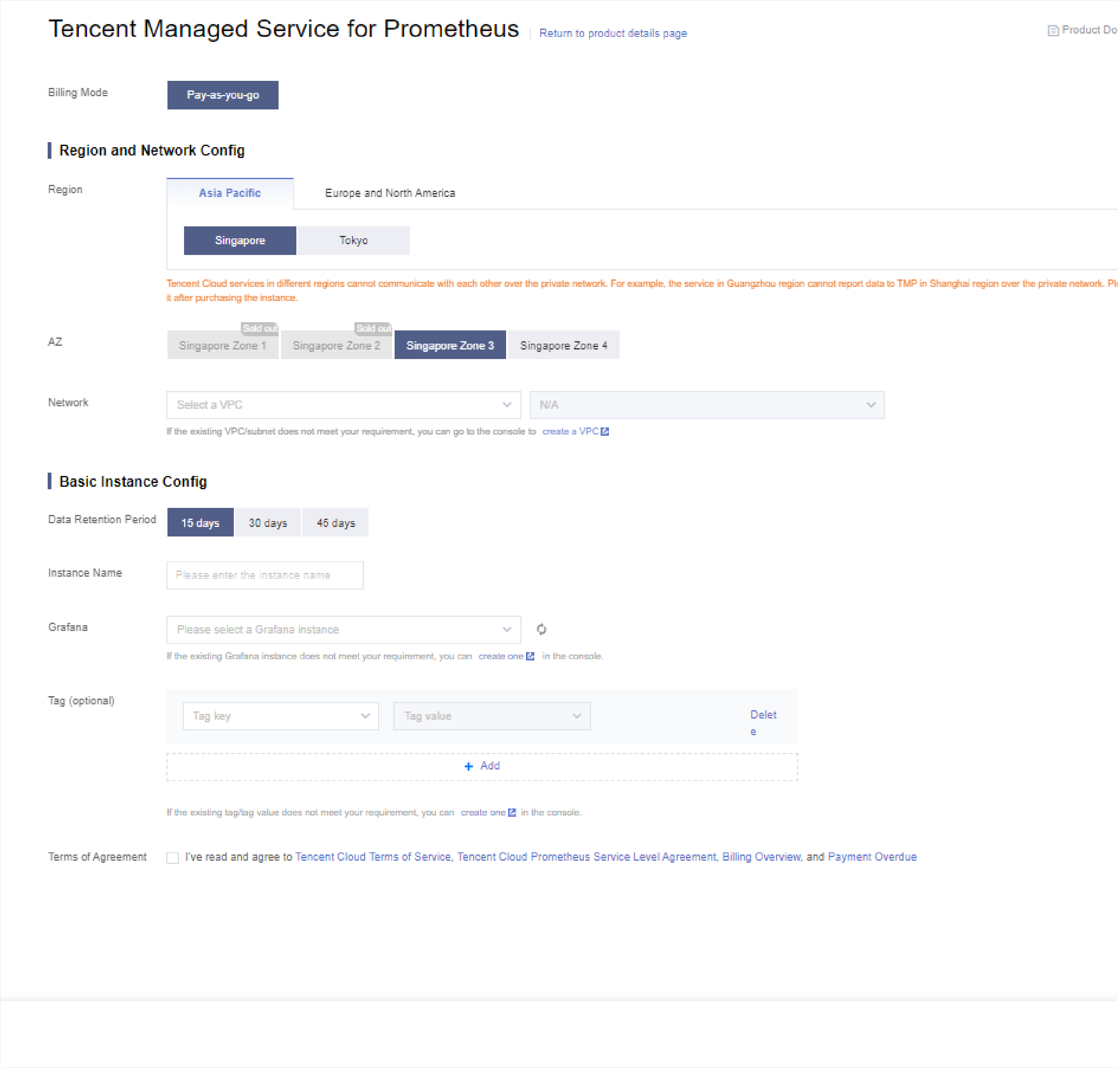
1. Log in to the TMP console and click **Create** to purchase a TMP instance.



□	Instance ID/Name	Monitoring/Status	AZ	Network
□	[redacted] test	Running	Singapore Zone 3	Network:D Subnet:rs

Total items: 1

2. On the purchase page, select the target instance specification and network. Make sure that the TMP and CVM instances have the same VPC IP range so that data can be collected. Select the instance specification based on your reported data volume.



Tencent Managed Service for Prometheus   Return to product details page   Product Do

Billing Mode: Pay-as-you-go

Region and Network Config

Region: Asia Pacific | Europe and North America

Singapore | Tokyo

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

AZ: Singapore Zone 1 (Sold out) | Singapore Zone 2 (Sold out) | Singapore Zone 3 | Singapore Zone 4

Network: Select a VPC | N/A

If the existing VPC/subnet does not meet your requirement, you can go to the console to create a VPC

Basic Instance Config

Data Retention Period: 15 days | 30 days | 45 days

Instance Name: Please enter the instance name

Grafana: Please select a Grafana instance

If the existing Grafana instance does not meet your requirement, you can create one in the console.

Tag (optional): Tag key | Tag value | Delete

+ Add

If the existing tag/tag value does not meet your requirement, you can create one in the console.

Terms of Agreement: □ I've read and agree to Tencent Cloud Terms of Service, Tencent Cloud Prometheus Service Level Agreement, Billing Overview, and Payment Overdue

3. Click **Buy Now** and make the payment.

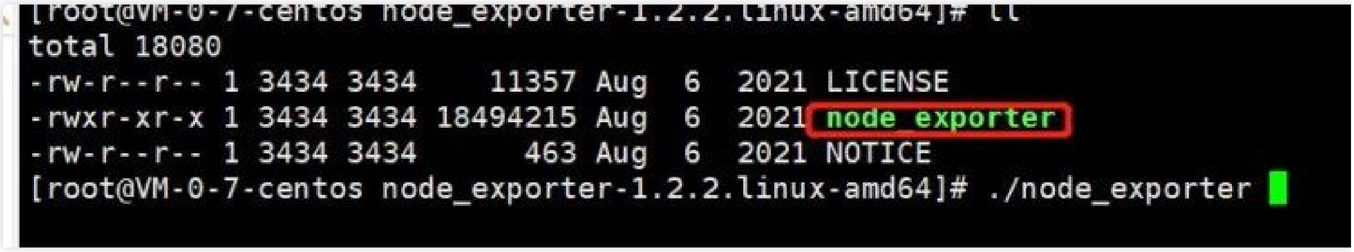
Integrating CVM Basic Metrics

1. Download and install Node Exporter.

Download and install Node Exporter (used to collect basic metric data) in the target CVM instance. Click here or run the following command for download:

```
wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_expo
```

The file directory is as follows:



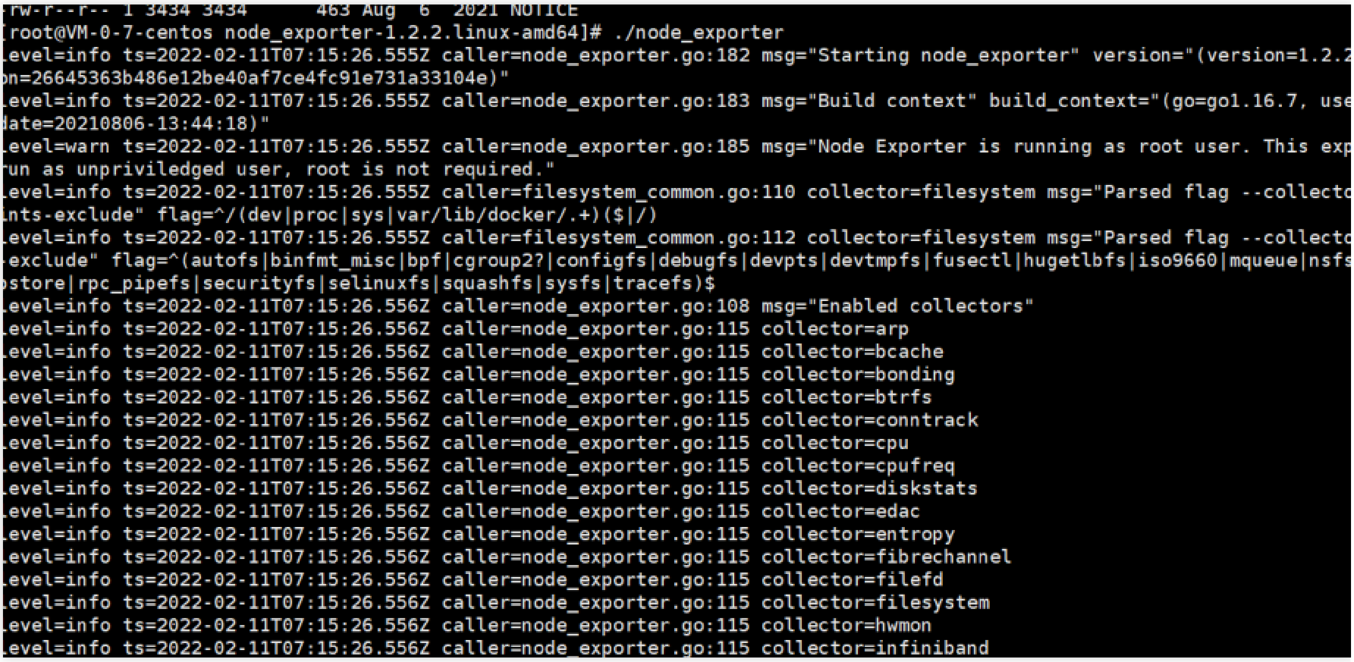
```
[root@VM-0-7-centos node_exporter-1.2.2.linux-amd64]# ll
total 18080
-rw-r--r-- 1 3434 3434    11357 Aug  6  2021 LICENSE
-rwxr-xr-x 1 3434 3434 18494215 Aug  6  2021 node exporter
-rw-r--r-- 1 3434 3434      463 Aug  6  2021 NOTICE
[root@VM-0-7-centos node_exporter-1.2.2.linux-amd64]# ./node_exporter
```

2. Run Node Exporter to collect basic monitoring data.

2.1 Go to the target folder and run Node Exporter.

```
cd node_exporter-1.3.1.linux-amd64
./node_exporter
```

If the following result is displayed, basic monitoring data has been collected successfully.

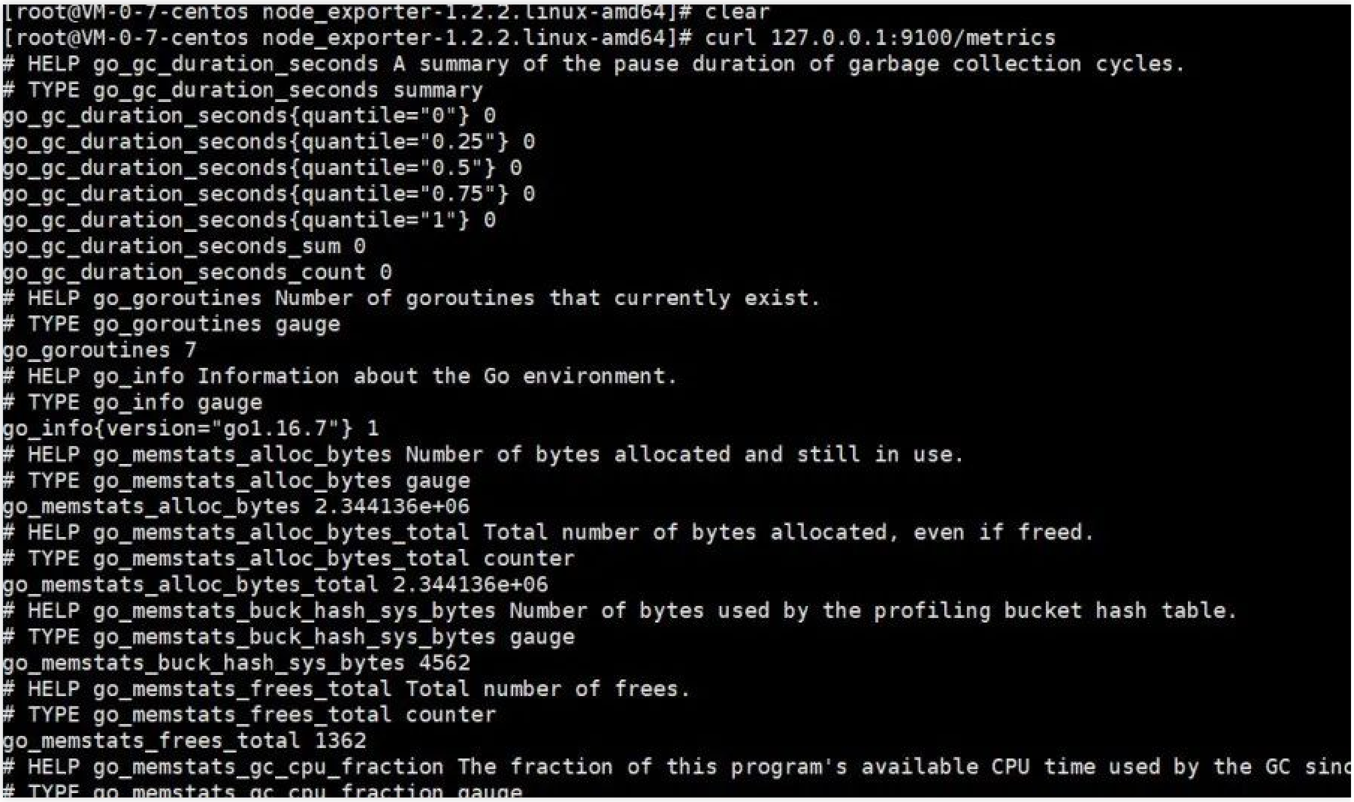


```
rw-r--r-- 1 3434 3434      463 Aug  6  2021 NOTICE
[root@VM-0-7-centos node_exporter-1.2.2.linux-amd64]# ./node_exporter
level=info ts=2022-02-11T07:15:26.555Z caller=node_exporter.go:182 msg="Starting node_exporter" version="(version=1.2.2
on=26645363b486e12be40af7ce4fc91e731a33104e)"
level=info ts=2022-02-11T07:15:26.555Z caller=node_exporter.go:183 msg="Build context" build_context="(go=go1.16.7, use
date=20210806-13:44:18)"
level=warn ts=2022-02-11T07:15:26.555Z caller=node_exporter.go:185 msg="Node Exporter is running as root user. This exp
run as unprivileged user, root is not required."
level=info ts=2022-02-11T07:15:26.555Z caller=filesystem_common.go:110 collector=filesystem msg="Parsed flag --collecto
ints-exclude" flag=^/(dev|proc|sys|var/lib/docker/.+)($|/)
level=info ts=2022-02-11T07:15:26.555Z caller=filesystem_common.go:112 collector=filesystem msg="Parsed flag --collecto
exclude" flag=^(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue|nsfs
pstore|rpc_pipefs|securityfs|selinuxfs|squashfs|sysfs|tracefs)$
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:108 msg="Enabled collectors"
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=arp
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=bcache
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=bonding
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=btrfs
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=conntrack
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=cpu
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=cpufreq
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=diskstats
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=edac
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=entropy
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=fibrechannel
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=filefd
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=filesystem
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=hwmon
level=info ts=2022-02-11T07:15:26.556Z caller=node_exporter.go:115 collector=infiniband
```

2.2 Run the following command to expose the basic monitoring data to port 9100:

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

You can see the following metric monitoring data that is exposed after the command is executed.



```
[root@VM-0-7-centos node_exporter-1.2.2.linux-amd64]# clear
[root@VM-0-7-centos node_exporter-1.2.2.linux-amd64]# 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="go1.16.7"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.344136e+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 2.344136e+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 4562
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1362
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC sinc
# TYPE go_memstats_gc_cpu_fraction gauge
```

3. Add a scrape task.

Log in to the TMP console, select **Integration Center** > **CVM**, and configure the information in **Task Configuration** as prompted.

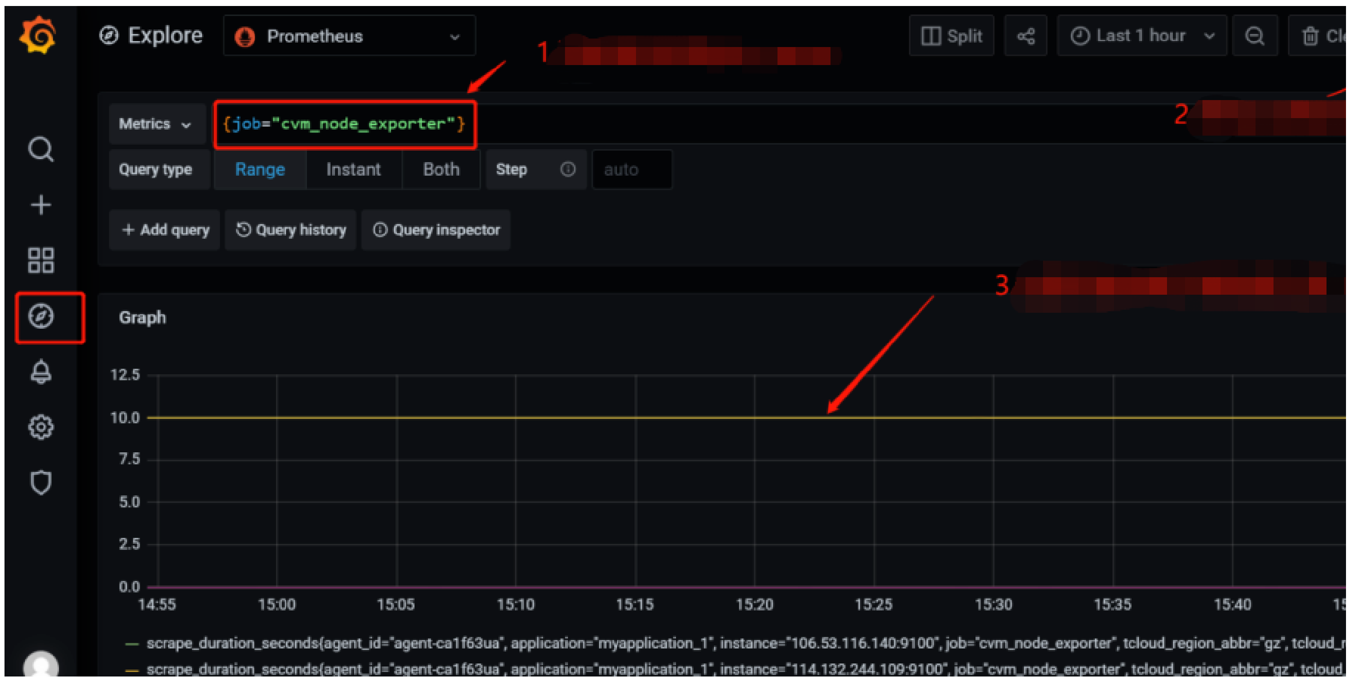
Below is a sample configuration of a scrape task:

```
job_name: example-job-name
metrics_path: /metrics
cvm_sd_configs:
- region: ap-guangzhou
  ports:
  - 9100
  filters:
  - name: tag:Sample tag key
    values:
    - Sample tag value
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
```

4. Check whether data is reported successfully.

Log in to the TMP console and click the Grafana icon to enter Grafana.

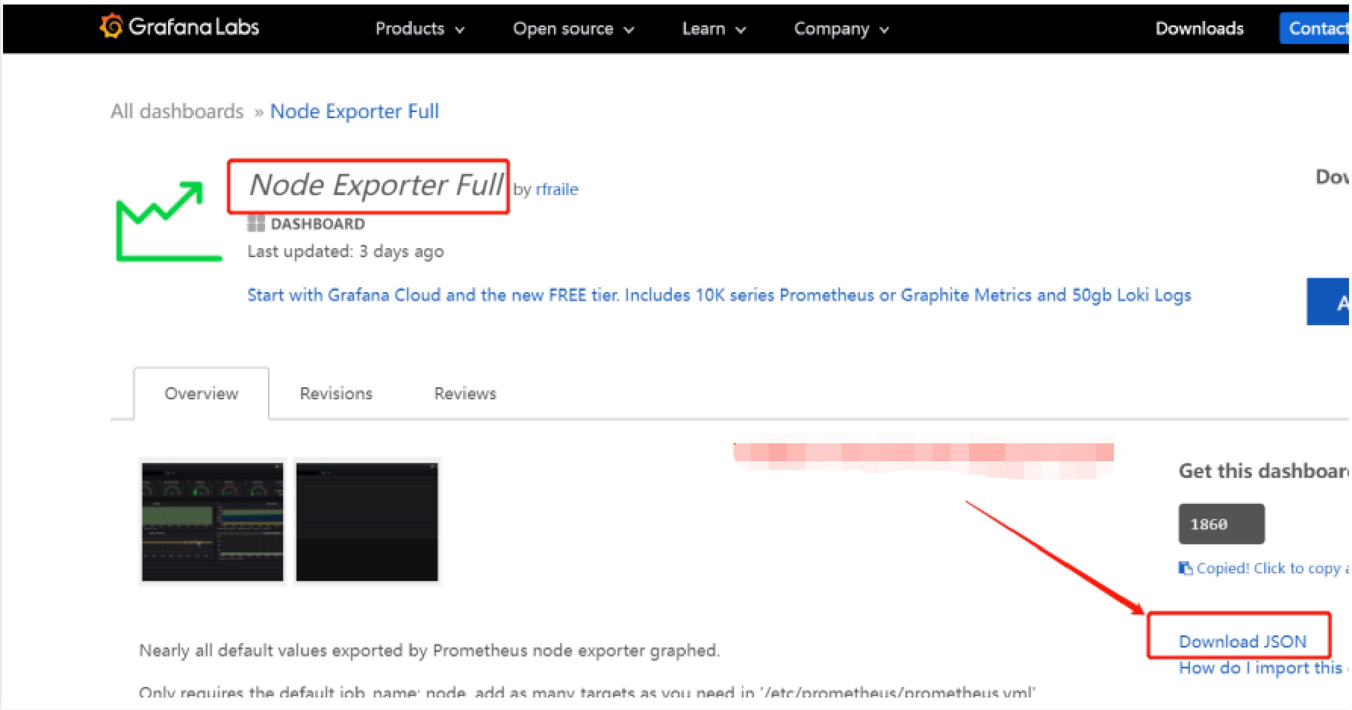
Search for `{job="cvm_node_exporter"}` in **Explore** to see whether there is data, and if so, data is reported successfully.



```
— scrape_duration_seconds{agent_id="agent-ca1f63ua", application="myapplication_1", instance="106.53.116.140:9100", job="cvm_node_exporter", tcloud_region_abbr="gz", tcloud_r
— scrape_duration_seconds{agent_id="agent-ca1f63ua", application="myapplication_1", instance="114.132.244.109:9100", job="cvm_node_exporter", tcloud_region_abbr="gz", tcloud_
```

5. Configure the dashboard page: Every product has some existing JSON files that can be directly imported into the dashboard.

5.1 **Download a dashboard file**: Go to the **Dashboard** page, search for `node_exporter` , and select the latest dashboard for download.



All dashboards » Node Exporter Full

Node Exporter Full by rfraile

DASHBOARD

Last updated: 3 days ago

Start with Grafana Cloud and the new FREE tier. Includes 10K series Prometheus or Graphite Metrics and 50gb Loki Logs

[redacted]

Get this dashboar

1860

Copied! Click to copy a

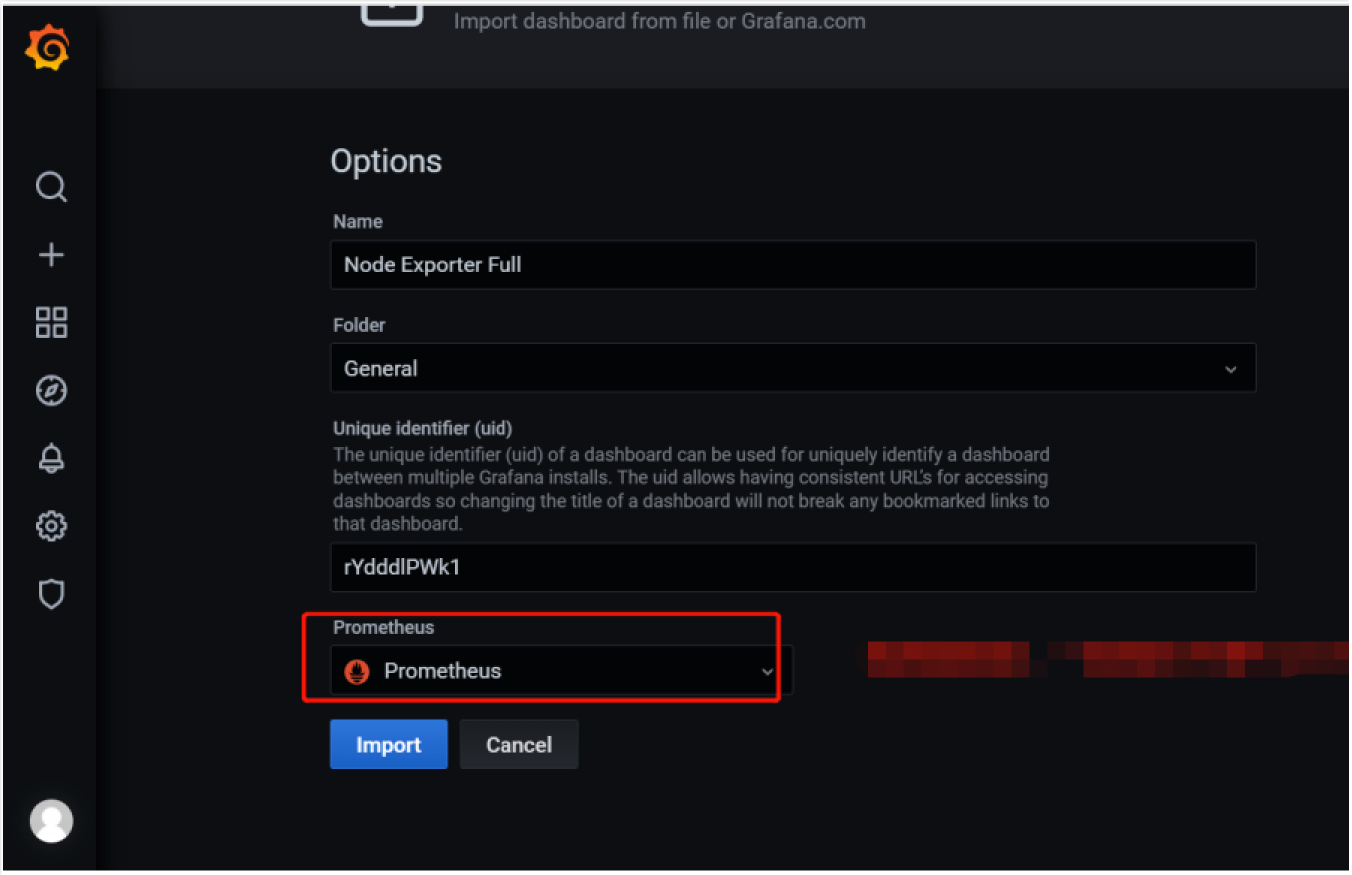
Download JSON

How do I import this

Nearly all default values exported by Prometheus node exporter graphed.

Only requires the default job_name: node, add as many targets as you need in '/etc/prometheus/prometheus.yml'

5.2 **Import a JSON file into the dashboard**: Log in to the TMP console, select **Basic Info** > **Grafana Address** to enter Grafana. In the Grafana console, select **Create** > **Import** and upload the dashboard file in **Upload JSON file**.



Import dashboard from file or Grafana.com

Options

Name: Node Exporter Full

Folder: General

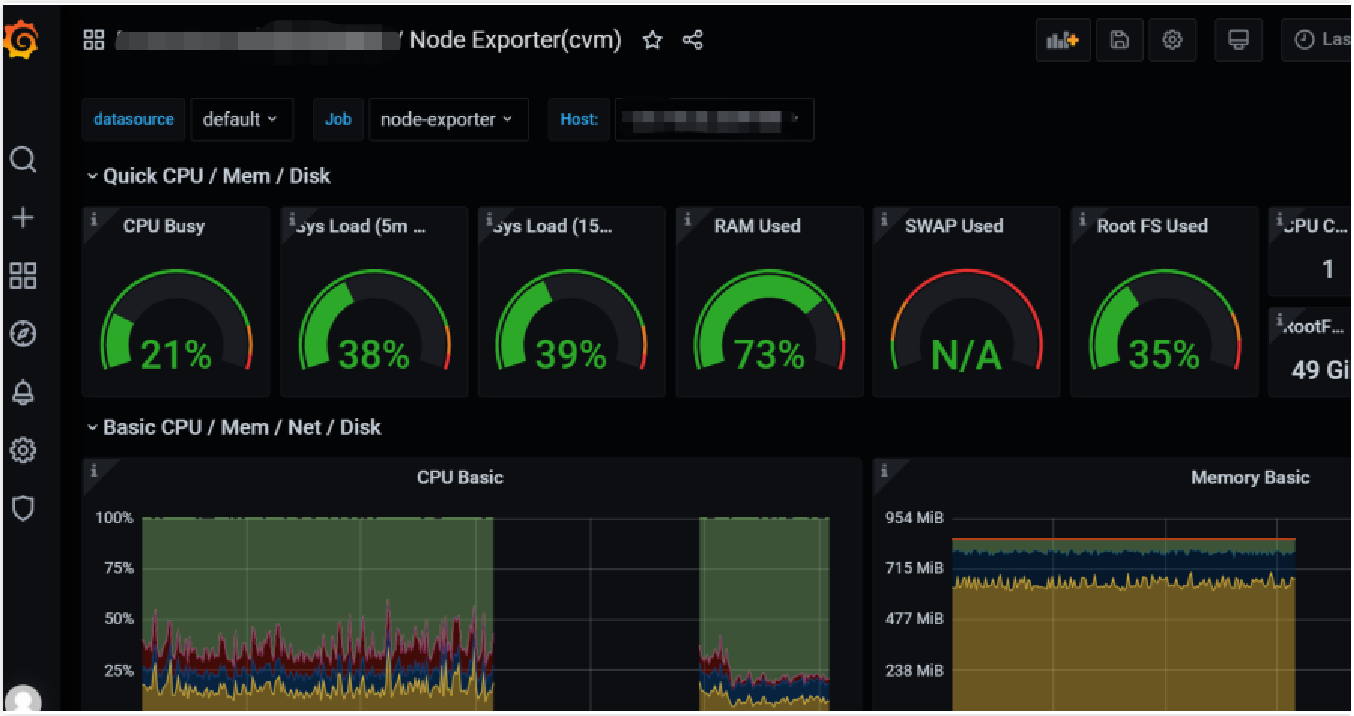
Unique identifier (uid)

The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

rYdddlPWk1

Prometheus: Prometheus

[redacted]



Integrating CVM Metrics at the Business Layer

Prometheus provides four metric types for different monitoring scenarios: Counter, Gauge, Histogram, and Summary. The Prometheus community provides SDKs for multiple programing languages, which are basically similar in usage and mainly differ in the syntax. The following uses Go as an example to describe how to report custom monitoring metrics. For detailed directions of other metric types, see Custom Monitoring.

Counter

A metric in Counter type increases monotonically and will be reset after service restart. You can use counters to monitor the numbers of requests, exceptions, user logins, orders, etc.

1. You can use a counter to monitor the number of orders as follows:

```
package order

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

// Define the counter object to be monitored
var (
 opsProcessed = promauto.NewCounterVec(prometheus.CounterOpts{
     Name: "order_service_processed_orders_total",
     Help: "The total number of processed orders",
 }, []string{"status"}) // Processing status
)

// Process the order
func makeOrder() {
 opsProcessed.WithLabelValues("success").Inc() // Success
 // opsProcessed.WithLabelValues("fail").Inc() // Failure

 // Order placement business logic
}
```

For example, you can use the `rate()` function to get the order increase rate:

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

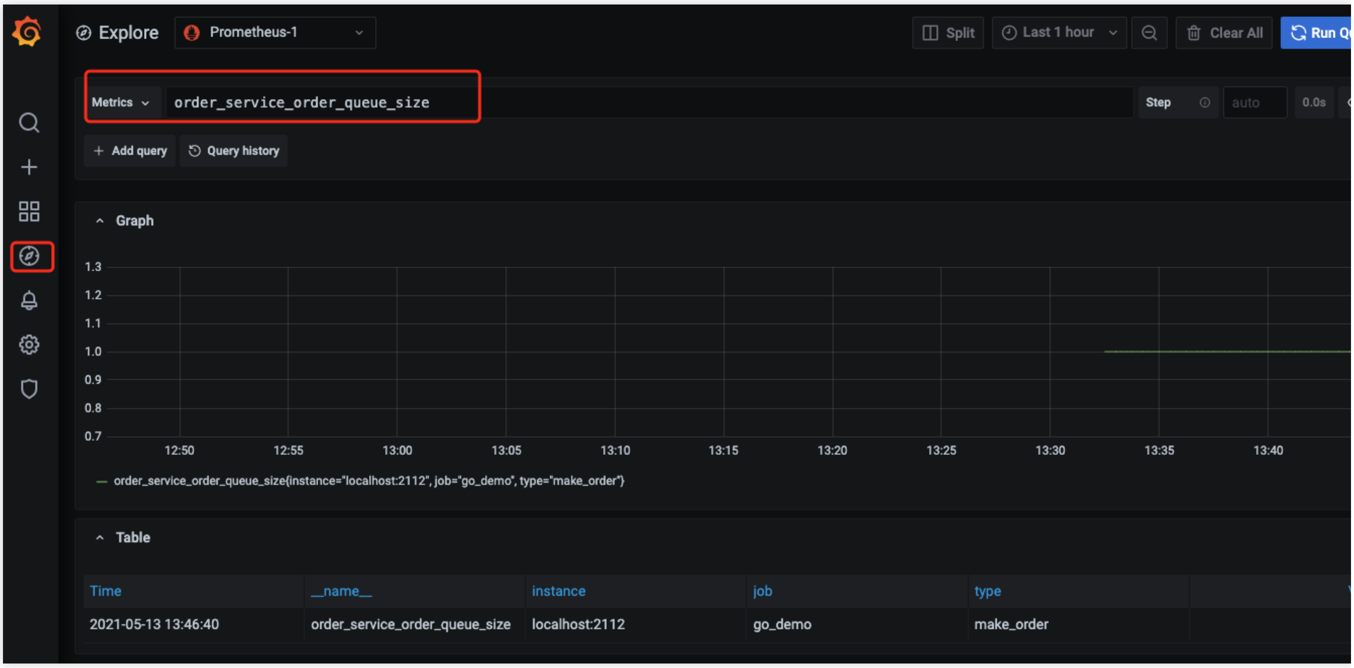
2. Expose Prometheus metrics:

Use `promhttp.Handler()` to expose the metric tracking data to the HTTP service.

```
package main

import (
 "net/http"

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

func main() {
     // Business code

     // Expose Prometheus metrics in the HTTP service
     http.Handle("/metrics", promhttp.Handler())

     // Business code
}
```

3. Collect data:

After the tracking of custom metrics for your business is completed and the application is released, you can use Prometheus to collect the monitoring metric data. After the collection is completed, wait a few minutes and then you can view the business metric monitoring data in Grafana integrated in TMP.



Metrics `order_service_order_queue_size`

— order_service_order_queue_size{instance="localhost:2112", job="go_demo", type="make_order"}

Time	__name__	instance	job	type
2021-05-13 13:46:40	order_service_order_queue_size	localhost:2112	go_demo	make_order

TKE Monitoring

Last updated：2024-01-29 15:55:08

Background

As we all know, Prometheus is the best monitoring tool for container scenarios. However, self-building Prometheus is too expensive for small and medium-sized enterprises with limited Ops manpower, and it is likely to hit performance bottlenecks for large enterprises with rapid business development. Therefore, using Prometheus managed on the cloud has become the first choice for more and more cloud companies. For how to use the managed Prometheus to monitor TKE, see Tencent Kubernetes Engine (TKE).

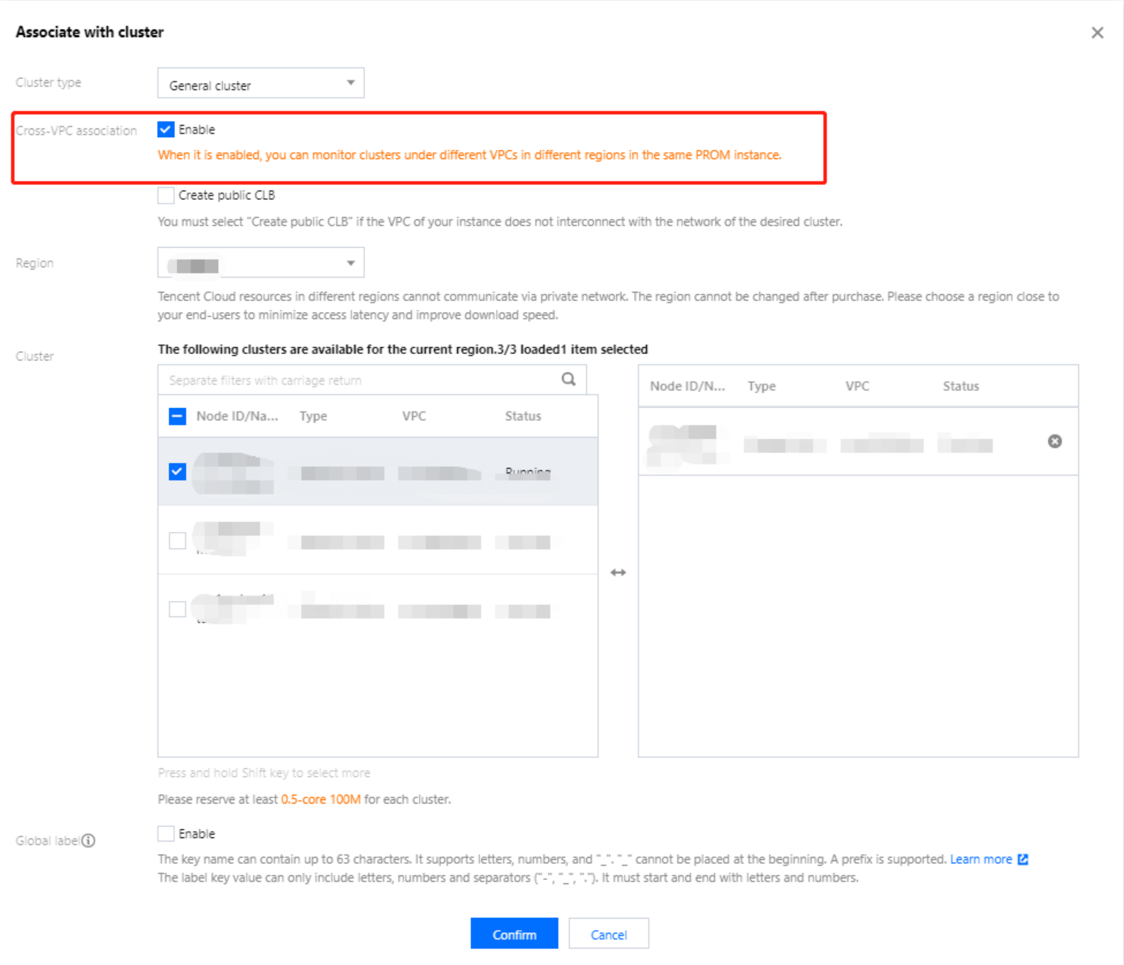
Directions

Step 1. Purchasing an instance

1. Log in to the TMP console.
2. Click **Create**, select the purchase region, storage duration and select the Grafana instance to be associated based on your needs. If there is no Grafana instance, see Creating Instance to create one. You need to create an instance and complete the purchase.
3. After completing the configuration, click **Buy Now**. For more information on billing rules, see Pay-as-You-Go Description.

Step 2. Integrate with TKE

1. After creating the instance, click the **ID**/**Name** of the target instance in the instance list to enter the instance details page.
2. On the left sidebar, click **Integrate with TKE** > **Associate Cluster**.
3. Select the cluster that needs to be associated in the pop-up window. A total of 4 types of clusters are supported: standard cluster, elastic cluster, registered cluster, and edge cluster. The clusters can be across VPCs. If different VPCs are not interconnected, you need to create a public network CLB instance.



Associate with cluster

Cluster type: General cluster

Cross-VPC association: ☑ Enable
When it is enabled, you can monitor clusters under different VPCs in different regions in the same PROM instance.

□ Create public CLB
You must select "Create public CLB" if the VPC of your instance does not interconnect with the network of the desired cluster.

Region: [redacted]
Tencent Cloud resources in different regions cannot communicate via private network. The region cannot be changed after purchase. Please choose a region close to your end-users to minimize access latency and improve download speed.

Cluster: **The following clusters are available for the current region.3/3 loaded1 item selected**

	Node ID/Na...	Type	VPC	Status
☑	[redacted]	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]	[redacted]
□	[redacted]	[redacted]	[redacted]	[redacted]

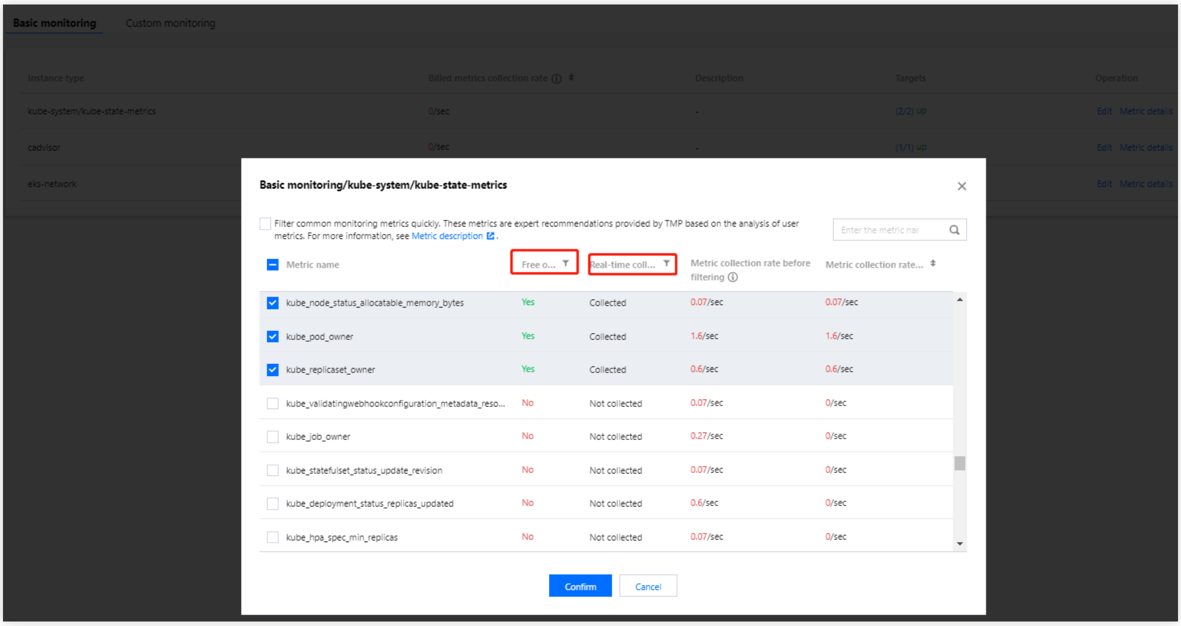
Node ID/N...	Type	VPC	Status
[redacted]	[redacted]	[redacted]	[redacted]

Press and hold Shift key to select more

Please reserve at least 0.5-core 100M for each cluster.

Global label: □ Enable
The key name can contain up to 63 characters. It supports letters, numbers, and "_". "_" cannot be placed at the beginning. A prefix is supported. Learn more
The label key value can only include letters, numbers and separators ("-", "_", "."). It must start and end with letters and numbers.

4. After associating the cluster, you can manually configure metrics for collection on **Cluster Monitoring** > **Data Collection Configuration**, view the default free basic collection metrics, and add or reduce the metrics as needed.



Instance type	Billed metrics collection rate	Description	Targets	Operation
kube-system/kube-state-metrics	0/sec	-	(2/2) up	Edit Metric details
cadvisor	0/sec	-	(1/1) up	Edit Metric details
eks-network				Edit Metric details

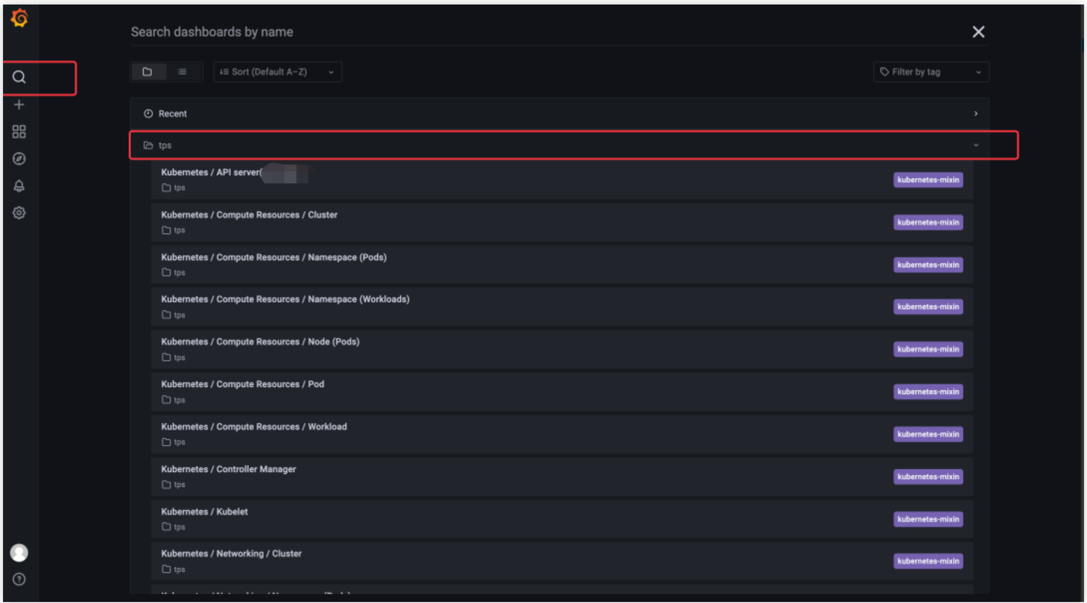
Basic monitoring/kube-system/kube-state-metrics

□ Filter common monitoring metrics quickly. These metrics are expert recommendations provided by TMP based on the analysis of user metrics. For more information, see Metric description.

	Metric name	Free o...	Real-time coll...	Metric collection rate before filtering	Metric collection rate...
☑	kube_node_status_allocatable_memory_bytes	Yes	Collected	0.07/sec	0.07/sec
☑	kube_pod_owner	Yes	Collected	1.6/sec	1.6/sec
☑	kube_replicaset_owner	Yes	Collected	0.6/sec	0.6/sec
□	kube_validatingwebhookconfiguration_metadata_reso...	No	Not collected	0.07/sec	0/sec
□	kube_job_owner	No	Not collected	0.27/sec	0/sec
□	kube_statefulset_status_update_revision	No	Not collected	0.07/sec	0/sec
□	kube_deployment_status_replicas_updated	No	Not collected	0.6/sec	0/sec
□	kube_hpa_spec_min_replicas	No	Not collected	0.07/sec	0/sec

Step 3. View monitoring data in Grafana

1. Click the Grafana icon to the right of the instance in the instance list to enter the Grafana service platform.
2. In the dashboard search list, TKE-related monitoring panels are preset by default, and click a panel name.



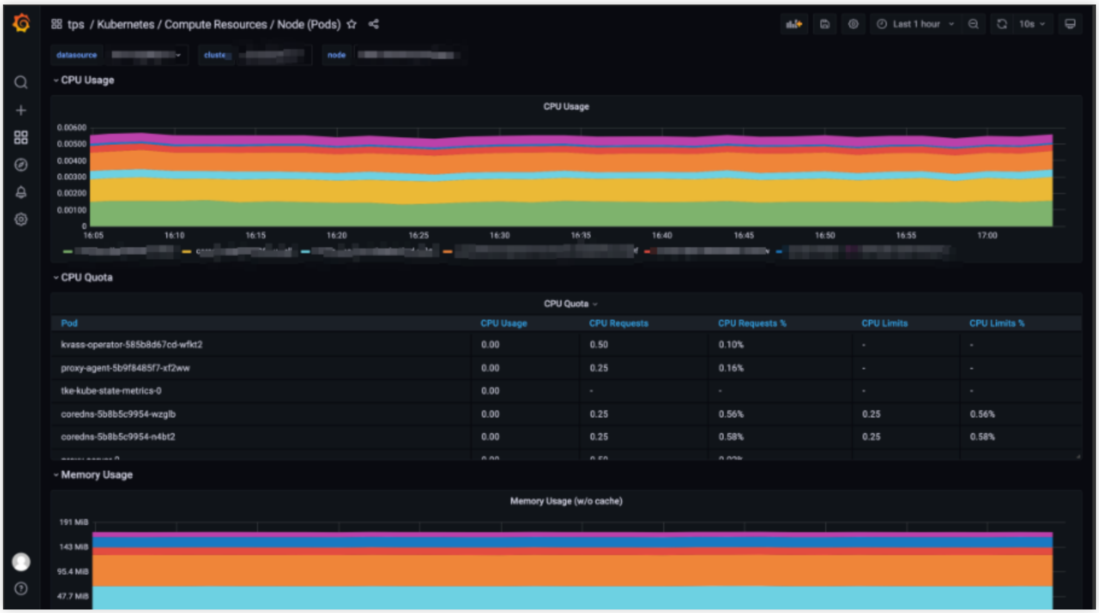
Search dashboards by name

Recent

tps

- Kubernetes / API server([redacted] — tps — kubernetes-mixin
- Kubernetes / Compute Resources / Cluster — tps — kubernetes-mixin
- Kubernetes / Compute Resources / Namespace (Pods) — tps — kubernetes-mixin
- Kubernetes / Compute Resources / Namespace (Workloads) — tps — kubernetes-mixin
- Kubernetes / Compute Resources / Node (Pods) — tps — kubernetes-mixin
- Kubernetes / Compute Resources / Pod — tps — kubernetes-mixin
- Kubernetes / Compute Resources / Workload — tps — kubernetes-mixin
- Kubernetes / Controller Manager — tps — kubernetes-mixin
- Kubernetes / Kubelet — tps — kubernetes-mixin
- Kubernetes / Networking / Cluster — tps — kubernetes-mixin

Enter the panel page, and you can view the preset monitoring data charts.



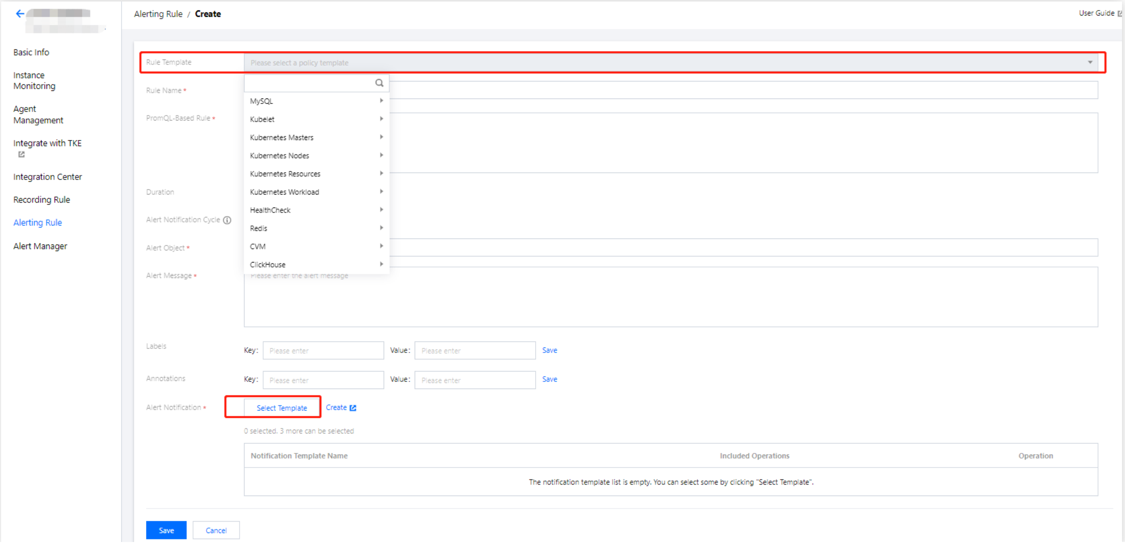
tps / Kubernetes / Compute Resources / Node (Pods)

CPU Quota

Pod	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
kvass-operator-585b8d67cd-wfkt2	0.00	0.50	0.10%	-	-
proxy-agent-5b9f8485f7-xf2ww	0.00	0.25	0.16%	-	-
tke-kube-state-metrics-0	0.00	-	-	-	-
coredns-5b8b5c9954-wzglb	0.00	0.25	0.56%	0.25	0.56%
coredns-5b8b5c9954-n4bt2	0.00	0.25	0.58%	0.25	0.58%

Step 4. Configure the alert policy

On the Prometheus instance details page, click **Alerting Rule**, and you can select a preset template type without manual configuration. For alert notifications, you can select existing notification templates on the TCOP to quickly configure alerts.



Alerting Rule / **Create**

Rule Template: Please select a policy template

Rule Name *

PromQL-Based Rule *

Duration

Alert Notification Cycle

Alert Object *

Alert Message *: Please enter the alert message

Labels — Key: Please enter Value: Please enter Save

Annotations — Key: Please enter Value: Please enter Save

Alert Notification *

0 selected. 3 more can be selected

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

Enabling Public Network Access for TKE Serverless Cluster

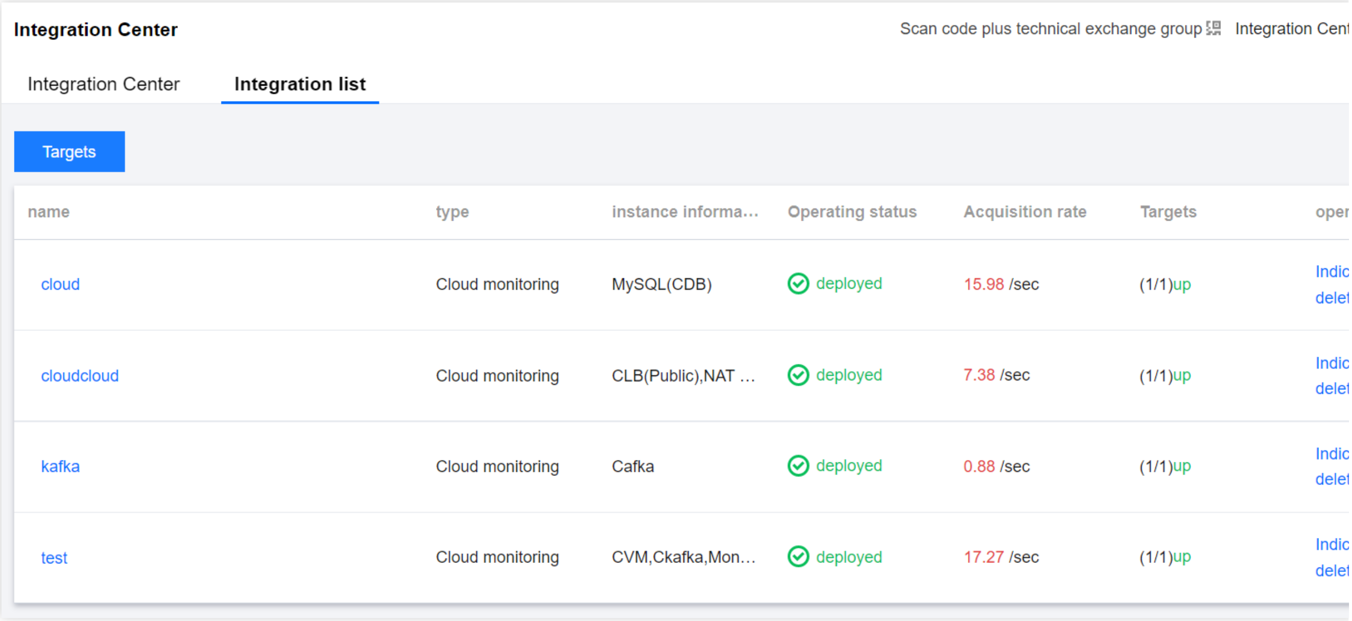
Last updated：2024-01-29 15:55:07

Overview

TMP is integrated with CM. When creating an integration, if you select COS, you need to enable public network access for the TKE Serverless cluster of the target CM exporter, as COS doesn't support private network access.

Directions

1. Log in to the TMP console.
2. Click the target instance to enter the instance management page. Then, click **Integration Center** > **Integration List**.
3. Click **Log** in the **Operation** column of the line where **Type** is **CM**.

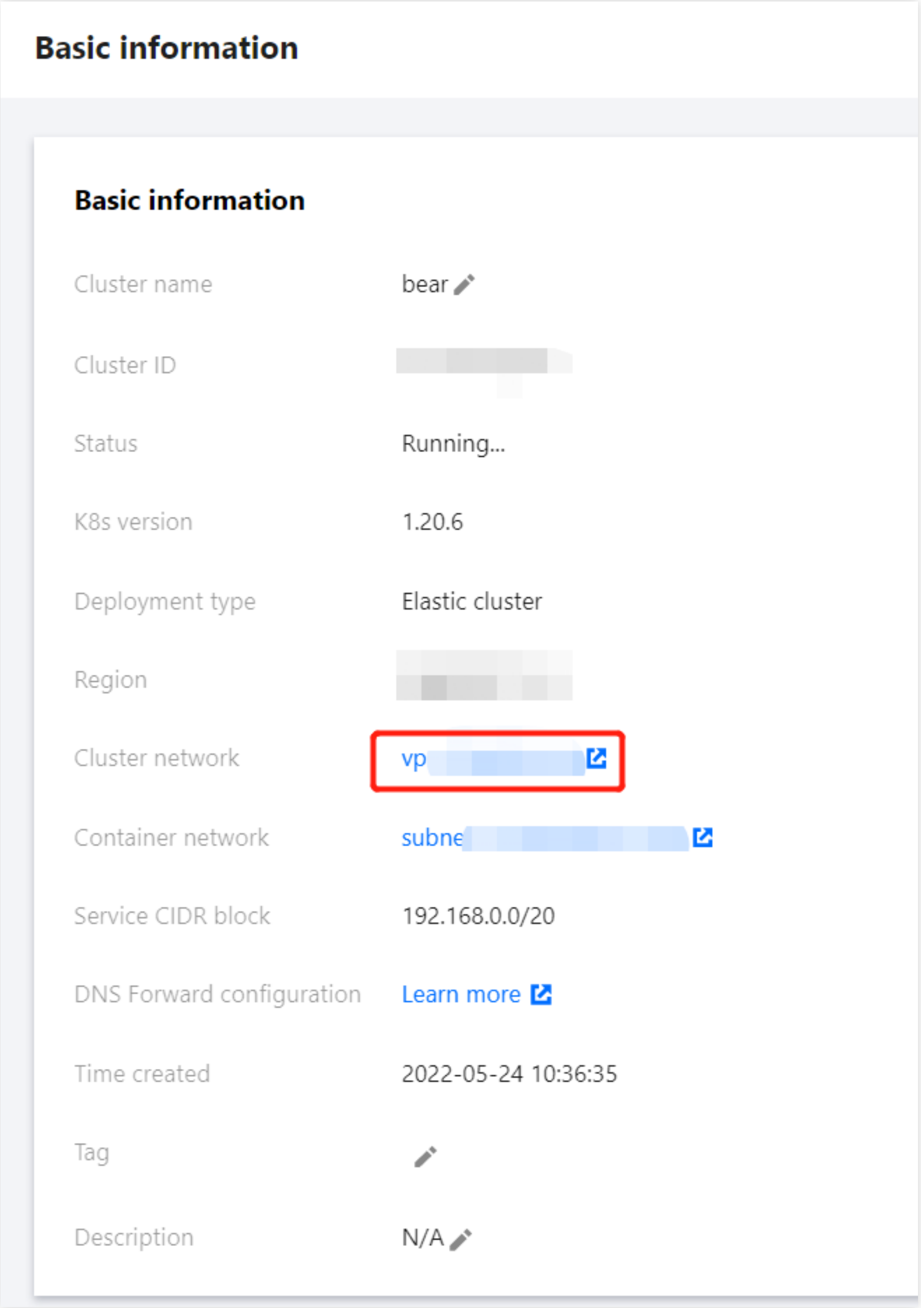


Integration Center

Scan code plus technical exchange group Integration Cent

name	type	instance informa...	Operating status	Acquisition rate	Targets	oper
cloud	Cloud monitoring	MySQL(CDB)	✓ deployed	15.98 /sec	(1/1)up	Indic delet
cloudcloud	Cloud monitoring	CLB(Public),NAT ...	✓ deployed	7.38 /sec	(1/1)up	Indic delet
kafka	Cloud monitoring	Cafka	✓ deployed	0.88 /sec	(1/1)up	Indic delet
test	Cloud monitoring	CVM,Ckafka,Mon...	✓ deployed	17.27 /sec	(1/1)up	Indic delet

4. On the topbar, switch to the Pod management page. Click the instance name to enter the cluster page.
5. On the **Basic Info** page, click **Container Network**.

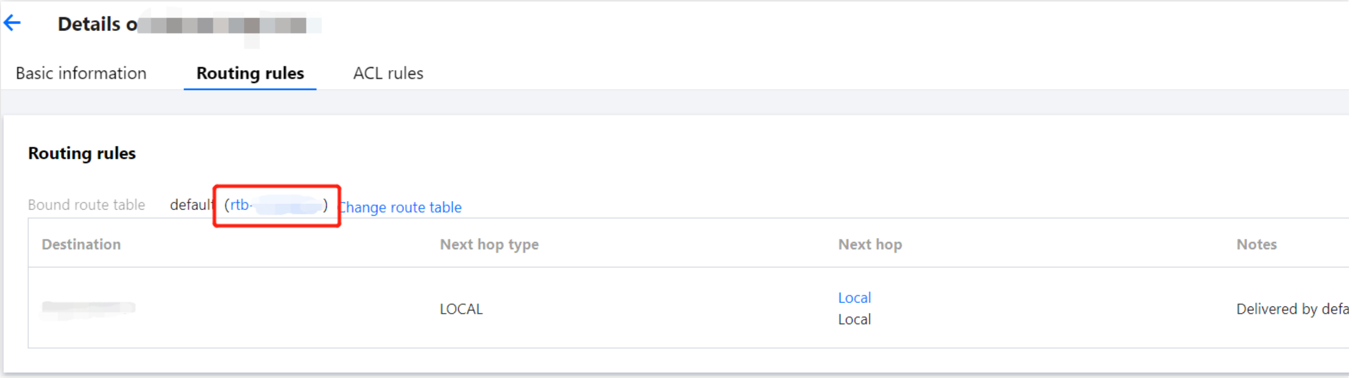


Basic information

Basic information

Cluster name	bear
Cluster ID	
Status	Running...
K8s version	1.20.6
Deployment type	Elastic cluster
Region	
Cluster network	vp
Container network	subne
Service CIDR block	192.168.0.0/20
DNS Forward configuration	Learn more
Time created	2022-05-24 10:36:35
Tag	
Description	N/A

6. On the topbar of the container network page, switch to the **Routing Policy** tab. Click the route table link (`rtb-xxx` in the list) to enter the route table page.



Details o[redacted]

Basic information | Routing rules | ACL rules

Routing rules

Bound route table default (rtb-[redacted]) Change route table

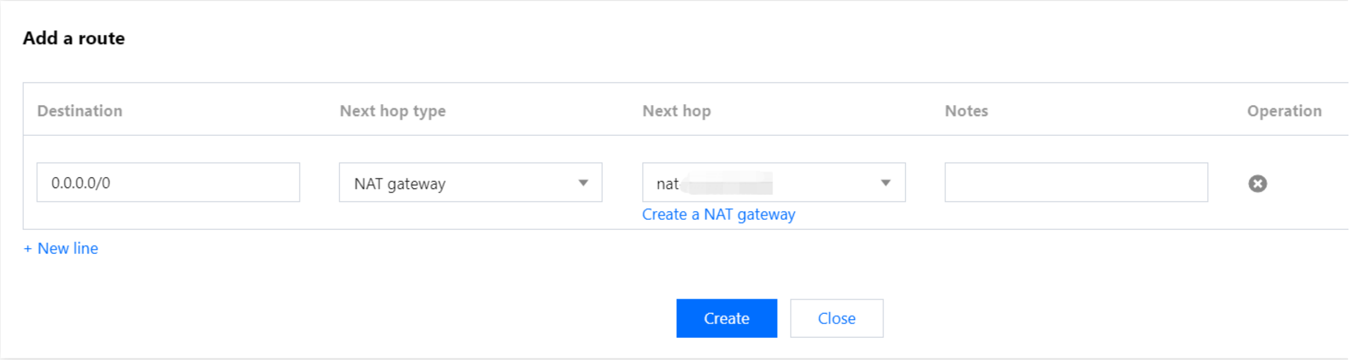
Destination	Next hop type	Next hop	Notes
[redacted]	LOCAL	Local Local	Delivered by defa

7. On the route table page, click **Create Routing Policy**.

Destination: Enter `0.0.0.0/0` .

Next Hop Type: Select **NAT Gateway**.

Next Hop: Select the target gateway. If there is no gateway, create one as instructed in Getting Started.



Add a route

Destination	Next hop type	Next hop	Notes	Operation
0.0.0.0/0	NAT gateway	nat[redacted] Create a NAT gateway		⊗

8. Click **Create**. After the creation is successful, public network access is enabled for TKE Serverless.

Connecting TMP to Local Grafana

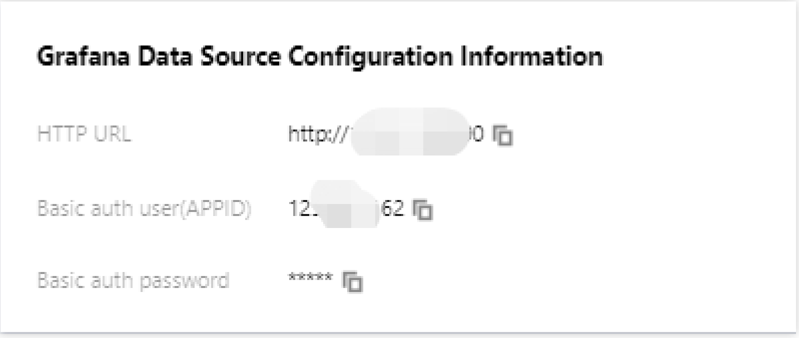
Last updated：2024-01-29 15:55:08

If you need to view the relevant data of TMP in the local Grafana system, you can use the HTTP API provided by TMP to do so. This document describes how to connect TMP data to local Grafana.

Directions

Step 1: Obtain the HTTP API provided by TMP

1. Log in to the TMP console.
2. Click the corresponding pay-as-you-go instance to enter the basic information page of TMP.
3. Get the HTTP API address in the service address module. If you need to improve the security of Grafana data read, you can obtain the authentication token of the TMP instance and fill it in as instructed in step 2.



Grafana Data Source Configuration Information

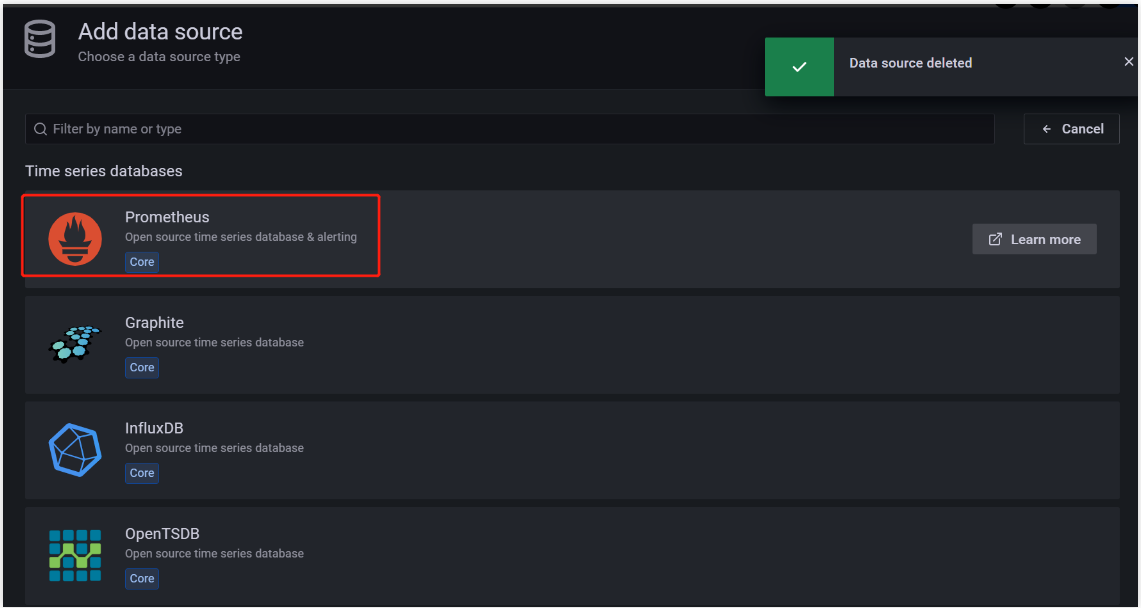
HTTP URL http://[redacted]0

Basic auth user(APPID) 12[redacted]62

Basic auth password *****

Step 2. Add data source to local Grafana

1. Log in to the local Grafana system with **admin account**.
2. Select **Configuration** > **Data Sources** on the left sidebar (non-admins cannot view this menu).
3. On the **Data Sources** page, click **Add data source**.
4. On the **Add data source** page, select **Prometheus**.



Add data source

Choose a data source type

Data source deleted

Filter by name or type

Cancel

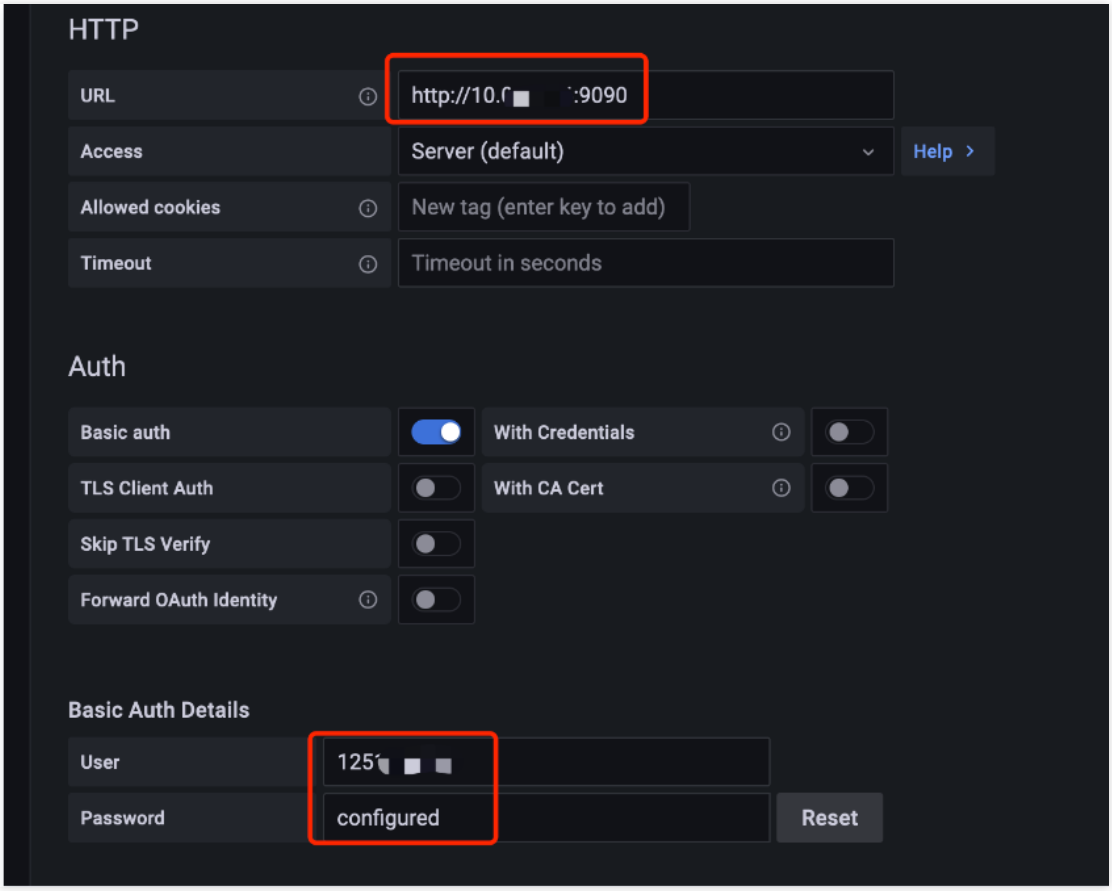
Time series databases

Data source	Description	Tag
Prometheus	Open source time series database & alerting	Core
Graphite	Open source time series database	Core
InfluxDB	Open source time series database	Core
OpenTSDB	Open source time series database	Core

5. On the **Settings** tab, enter a custom name in the **Name** field, and paste the **HTTP API** obtained in step 1 in the URL field.

6. Toggle on **Basic Auth** in the **Auth** module. In the **Basic Auth Details** module, set **User** same as **Basic auth user** and **Password** as **Basic auth password** obtained in the step 1.

7. Click **Save & test** to complete the settings.



HTTP

URL	http://10.[redacted]:9090
Access	Server (default)
Allowed cookies	New tag (enter key to add)
Timeout	Timeout in seconds

Auth

Basic auth	on	With Credentials	off
TLS Client Auth	off	With CA Cert	off
Skip TLS Verify	off		
Forward OAuth Identity	off		

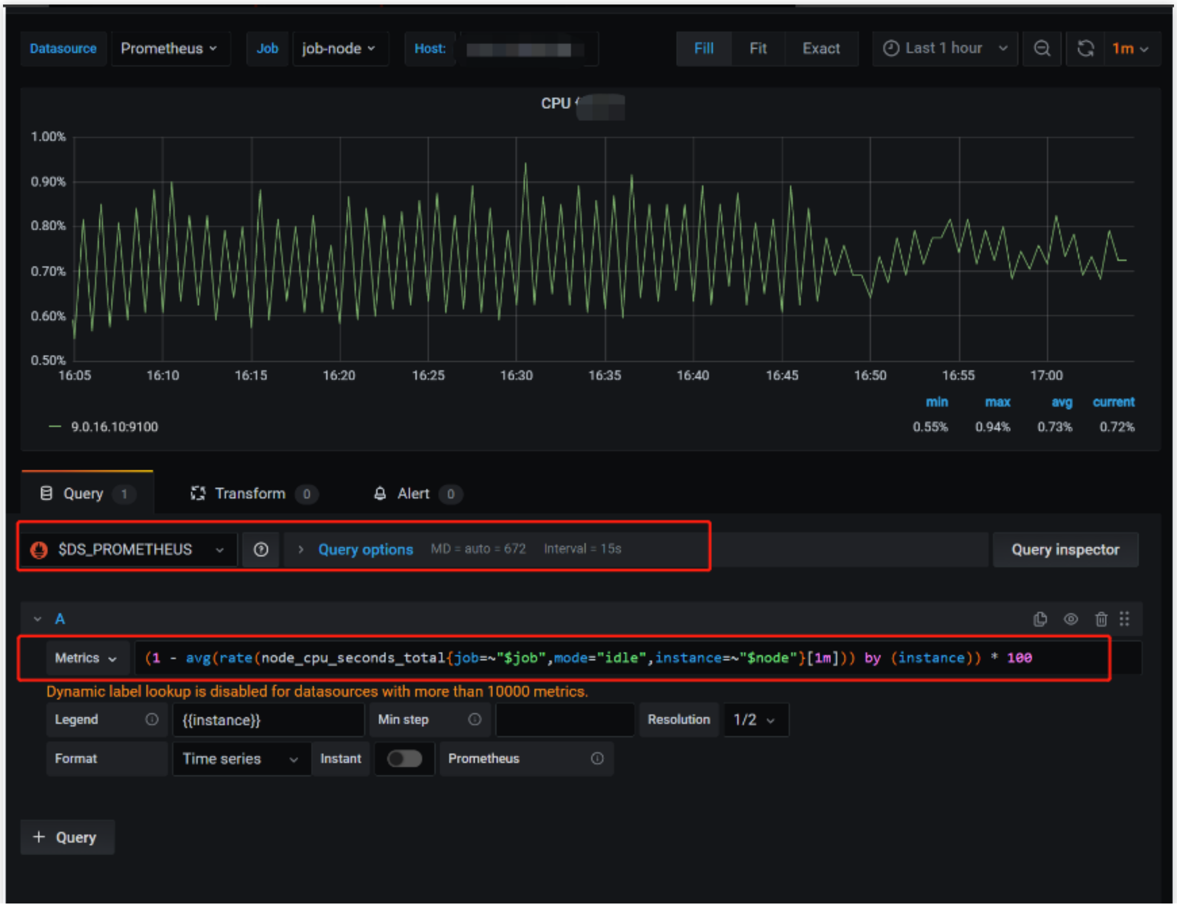
Basic Auth Details

User	125[redacted]
Password	configured

Step 3. Verify whether the connection is successful

Follow the steps below to verify whether TMP is successfully connected to the local Grafana:

1. Log in to your local Grafana system.
2. On the left sidebar, select **+ > Create**.
3. On the **New dashboard** page, click **Add a new panel**.
4. On the **Edit Panel** page, select the data source added in step 2 in the drop-down box on the **Query** page, enter the metric name in the **Metrics** field in the A module and press Enter.
5. If the chart of the corresponding metric can be displayed, the operation is successful. Otherwise, check whether the API address or token entered is correct, and whether the data source has TMP data.



Datasource: Prometheus; Job: job-node; Host: [redacted]

CPU [redacted]

$DS_PROMETHEUS

Query options MD = auto = 672 Interval = 15s

Metrics

```
(1 - avg(rate(node_cpu_seconds_total{job=~"$job",mode="idle",instance=~"$node"}[1m])) by (instance)) * 100
```

Dynamic label lookup is disabled for datasources with more than 10000 metrics.

Legend {{instance}} Min step Resolution 1/2

Format Time series Instant Prometheus

Enabling Public Network Access for Prometheus Instances

Last updated：2024-11-22 18:16:04

Operation Background

In some scenarios, you may need your Prometheus instance to have the ability to access the public network, for example:

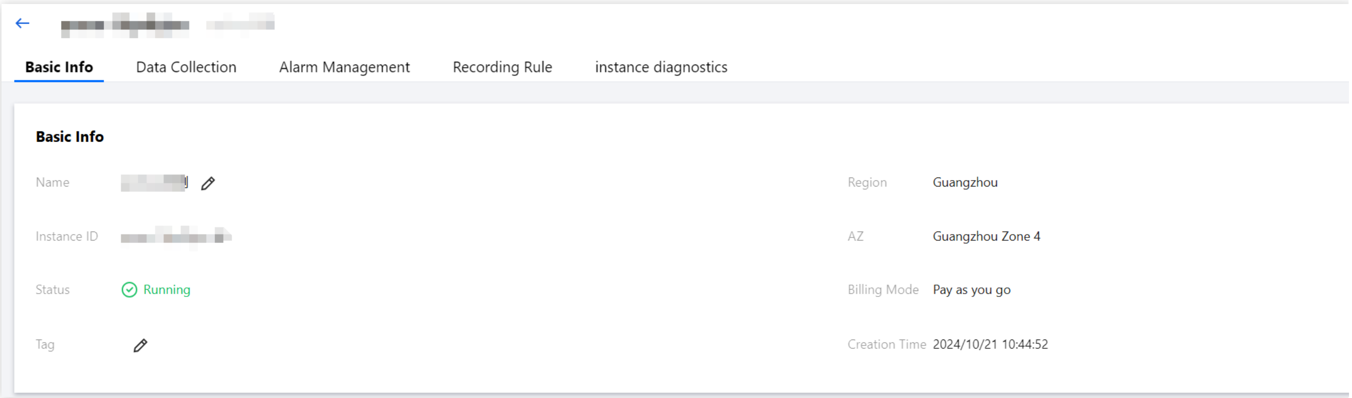
When Prometheus is integrated to monitor COS products on the cloud, since COS products do not support private network access, the Prometheus instance needs to have the ability to access the public network.

The WebHook in the Prometheus alarm policy is used, and this WebHook URL is a public network address.

In such cases, you need to enable public network access for Prometheus (which essentially means enabling public network access for the TKE-managed cluster where the Prometheus instance components are located).

Operations

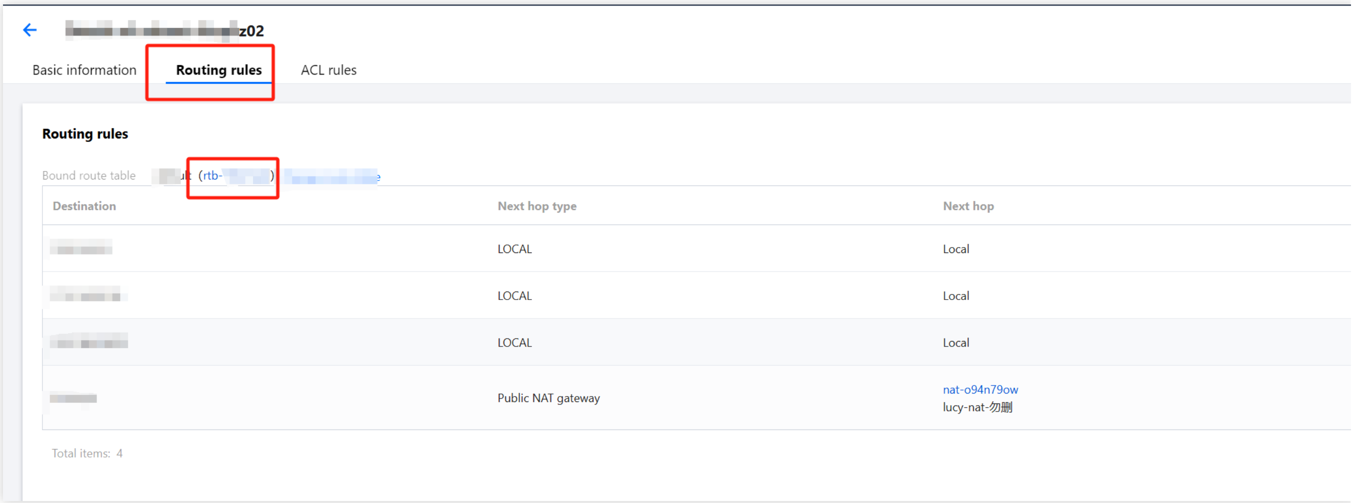
1. Log in to TMP Console.
2. Click the corresponding instance to enter the Instance Management page. Then, click **Subnet** on the Basic info tab page.



Basic Info

Name	[redacted]	Region	Guangzhou
Instance ID	[redacted]	AZ	Guangzhou Zone 4
Status	Running	Billing Mode	Pay as you go
Tag		Creation Time	2024/10/21 10:44:52

3. In the top menu of the subnet, switch to the **Routing rules** section. Click the route table link (rtb-xxx on the list) to enter the route table interface.



[redacted]z02

Routing rules

Bound route table [redacted] (rtb-[redacted]) [redacted]

Destination	Next hop type	Next hop
[redacted]	LOCAL	Local
[redacted]	LOCAL	Local
[redacted]	LOCAL	Local
[redacted]	Public NAT gateway	nat-o94n79ow lucy-nat-勿删

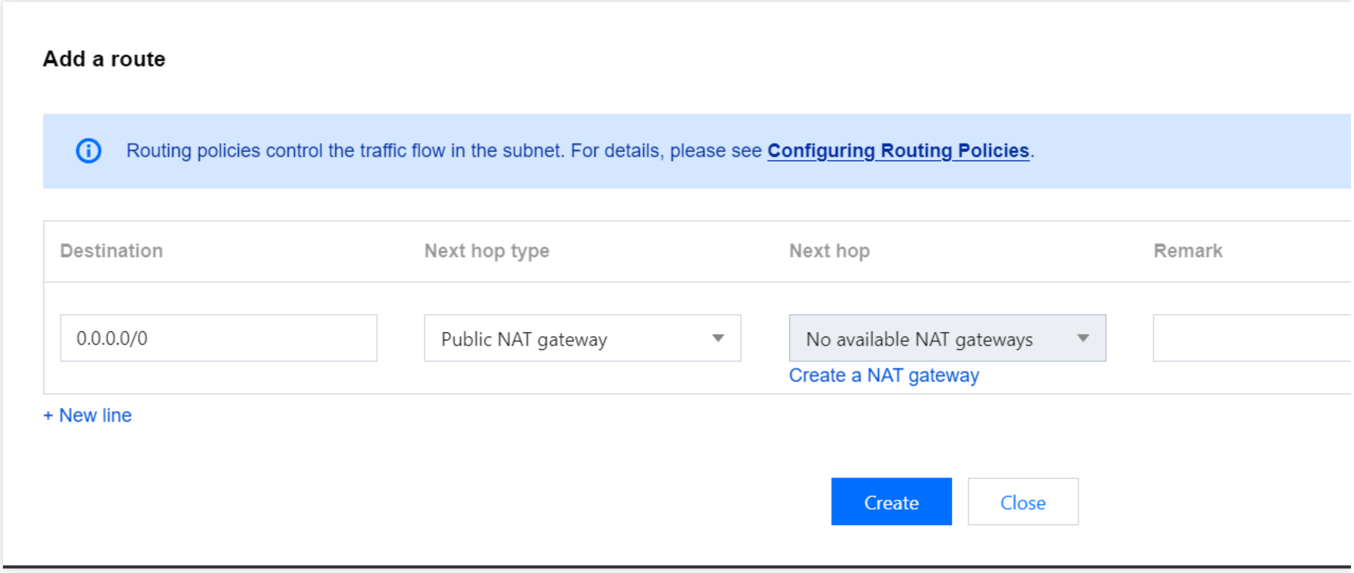
Total items: 4

4. On the route table page, click **Add routing policy**.

Destination: Enter 0.0.0.0/0.

Next hop type: select NAT Gateway.

Next hop: select the target gateway. If there is no gateway, please refer to the NAT Gateway guide to create one.



Add a route

Routing policies control the traffic flow in the subnet. For details, please see Configuring Routing Policies.

Destination	Next hop type	Next hop	Remark
0.0.0.0/0	Public NAT gateway	No available NAT gateways Create a NAT gateway	

5. After completing the settings, click **Create**. After the creation is successful, public network access is enabled for TKE Serverless.

Configuring a Public Network Address for a Prometheus Instance

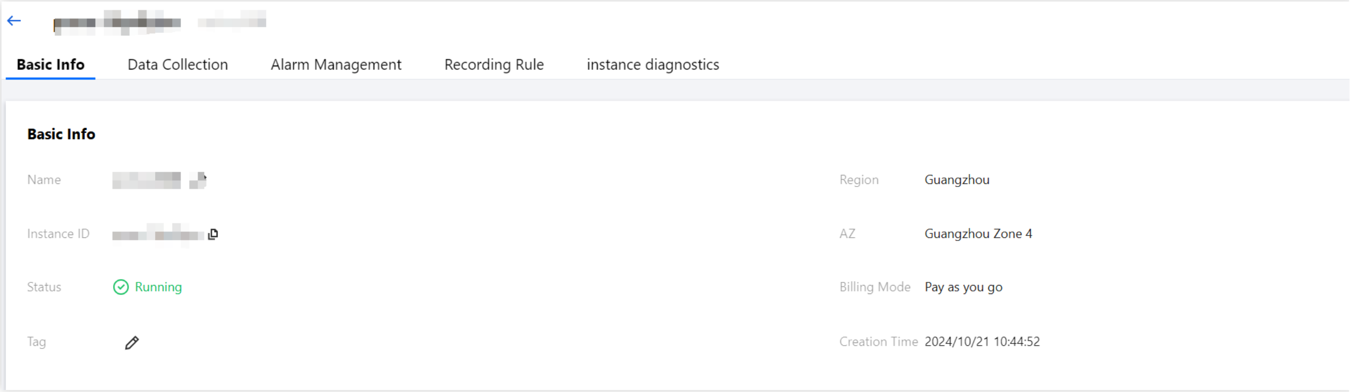
Last updated：2024-11-22 18:15:33

This document introduces how to configure a public network address for Prometheus monitoring.

Practice Steps

Step 1: Purchasing a Prometheus Instance

1. Log in to TMP Console.
2. Click **Create** in the upper left corner to enter the Prometheus purchase page. You can purchase the corresponding instance based on your actual needs. For details, see Creating Instance.
3. After successfully purchasing, click the ID/Name of the created instance to enter the **Basic Info** of the instance details page to obtain the Prometheus IPV4 address.



Basic Info

Name	[redacted]	Region	Guangzhou
Instance ID	[redacted]	AZ	Guangzhou Zone 4
Status	Running	Billing Mode	Pay as you go
Tag		Creation Time	2024/10/21 10:44:52

Step 2: Applying for the Cross-VPC Access Feature of CLB (Beta Feature)

1. Go to the CLB Cross-Region Binding 2.0 and Hybrid Cloud Deployment Beta Application page.
2. Fill in required information and submit the application.
3. You can use the cross-VPC access feature of CLB after the application is approved.

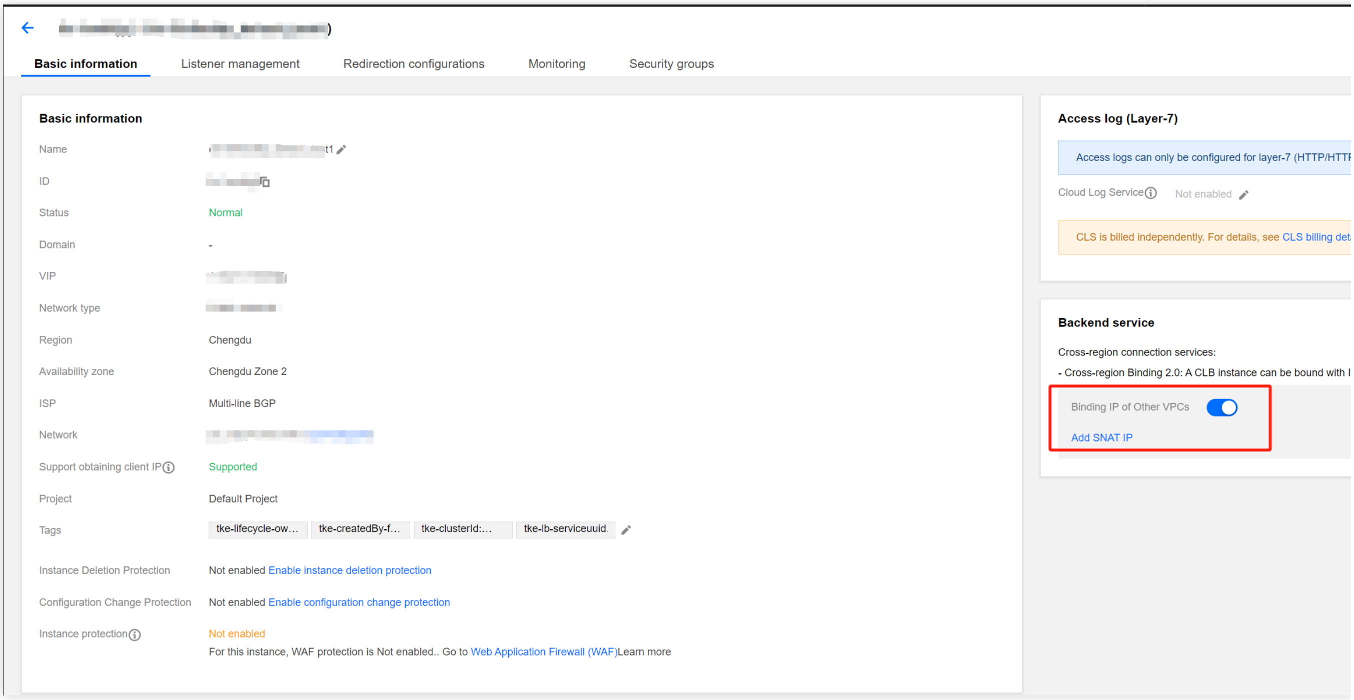
Step 3: Creating a CLB Instance for the Public Network

1. Go to the CLB Console and create a CLB instance.
2. Fill in the information as prompted. For detailed instructions, see Creating a CLB Instance.

Note:

To bind the private IP address of a Prometheus instance, you need to create a CLB instance under the same VPC as this Prometheus instance. If a CLB instance already exists on the public network, no creation is required.

3. After creation, enter the basic information page of the instance and enable the cross-VPC access feature.



Basic information

Name [redacted]t1

ID [redacted]

Status Normal

Domain -

VIP [redacted]

Network type [redacted]

Region Chengdu

Availability zone Chengdu Zone 2

ISP Multi-line BGP

Network [redacted]

Support obtaining client IP Supported

Project Default Project

Tags tke-lifecycle-ow... tke-createdBy-f... tke-clusterId:... tke-lb-serviceuuid

Instance Deletion Protection Not enabled Enable instance deletion protection

Configuration Change Protection Not enabled Enable configuration change protection

Instance protection Not enabled
For this instance, WAF protection is Not enabled.. Go to Web Application Firewall (WAF)Learn more

Access log (Layer-7)

Access logs can only be configured for layer-7 (HTTP/HTTI

Cloud Log Service Not enabled

CLS is billed independently. For details, see CLS billing det

Backend service

Cross-region connection services:

- Cross-region Binding 2.0: A CLB instance can be bound with I

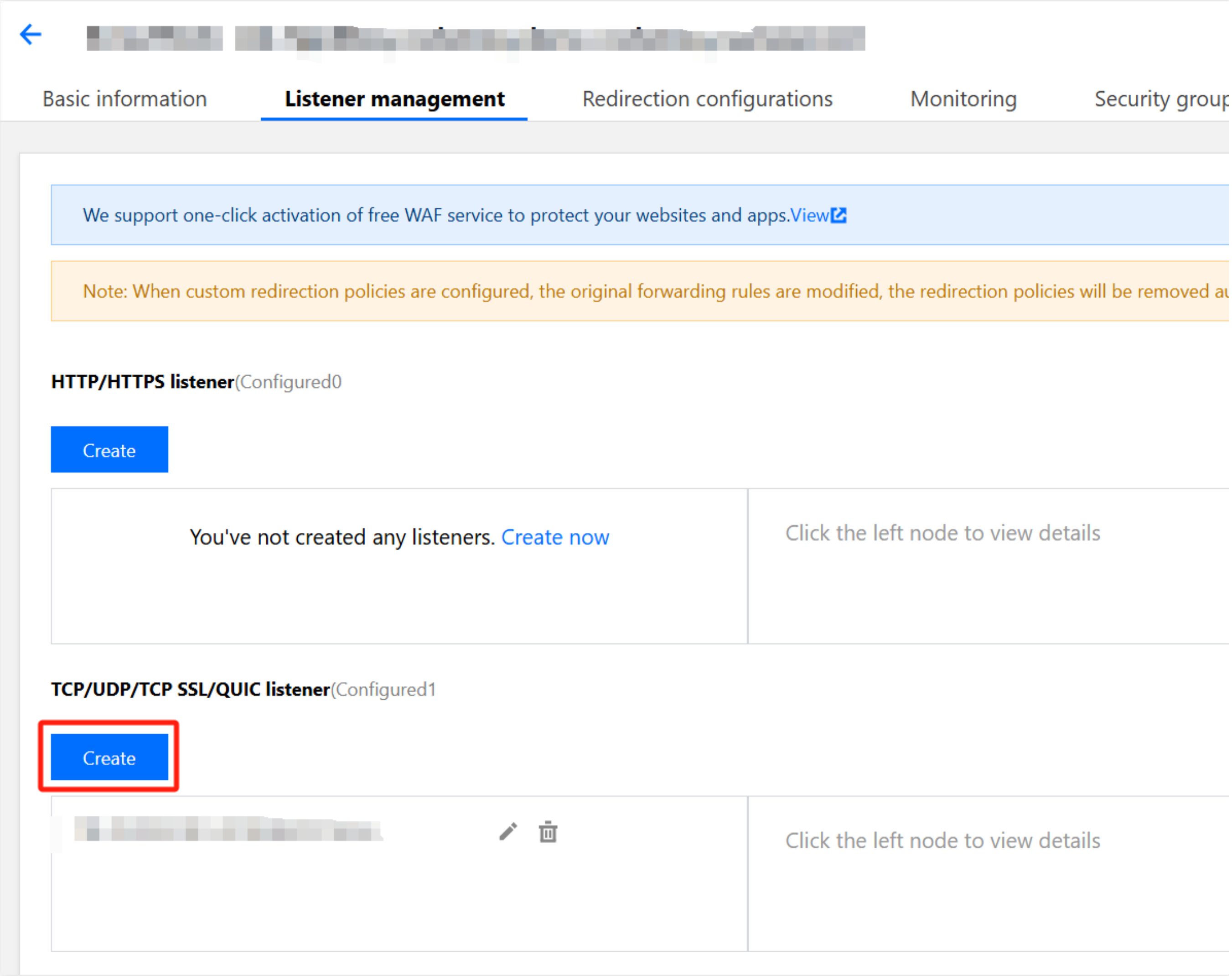
Binding IP of Other VPCs

Add SNAT IP

Step 4: Binding the Backend Service

1. Enter the **Listener management** page.

2. Click **Create** under the TCP/UDP/TCP SSL/QUIC listener.



Basic information | **Listener management** | Redirection configurations | Monitoring | Security group

We support one-click activation of free WAF service to protect your websites and apps.View

Note: When custom redirection policies are configured, the original forwarding rules are modified, the redirection policies will be removed au

HTTP/HTTPS listener(Configured0

Create

You've not created any listeners. Create now

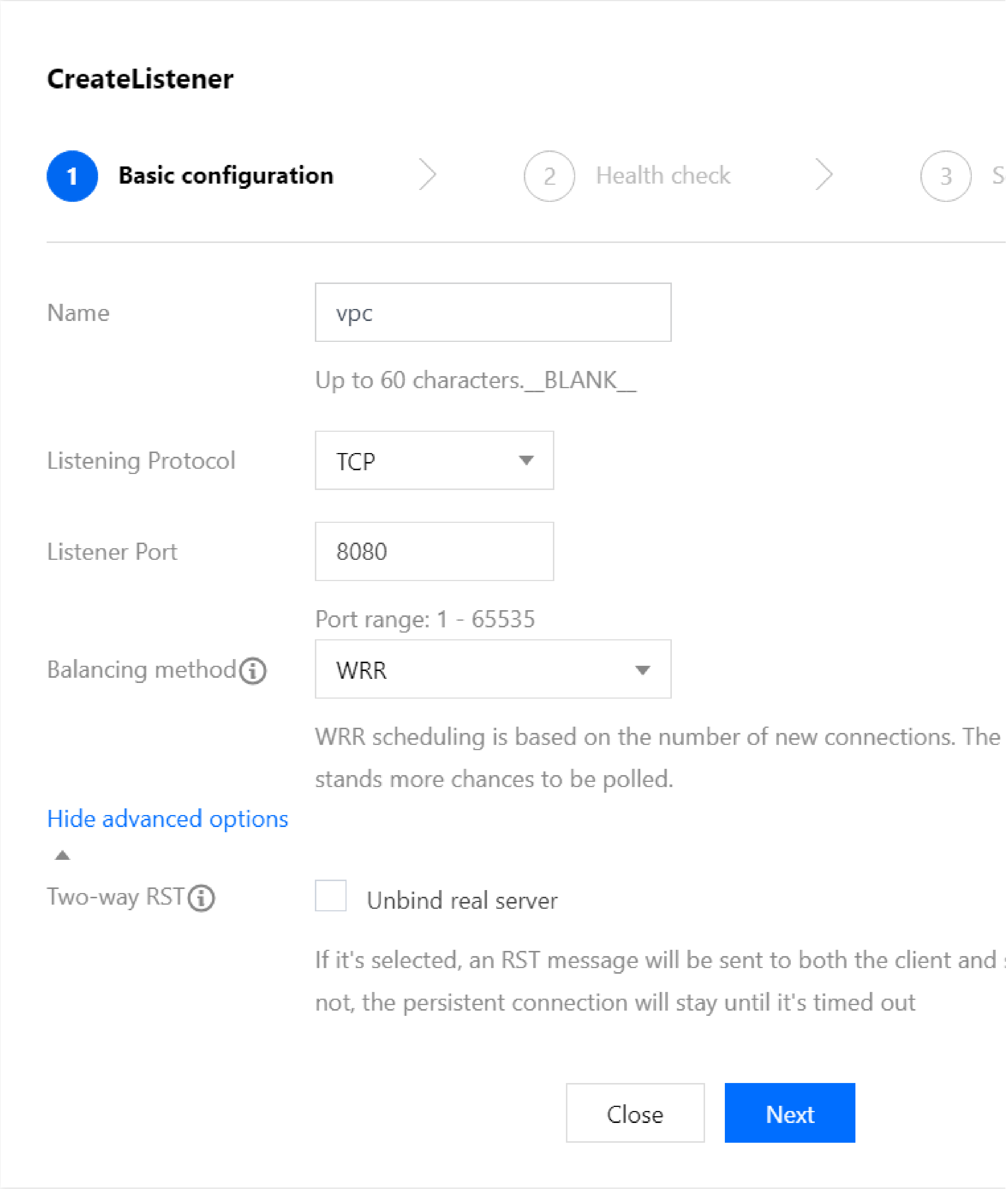
Click the left node to view details

TCP/UDP/TCP SSL/QUIC listener(Configured1

Create

Click the left node to view details

For the operation guide, see CLB Listener.



CreateListener

Name: vpc

Up to 60 characters.__BLANK__

Listening Protocol: TCP

Listener Port: 8080

Port range: 1 - 65535

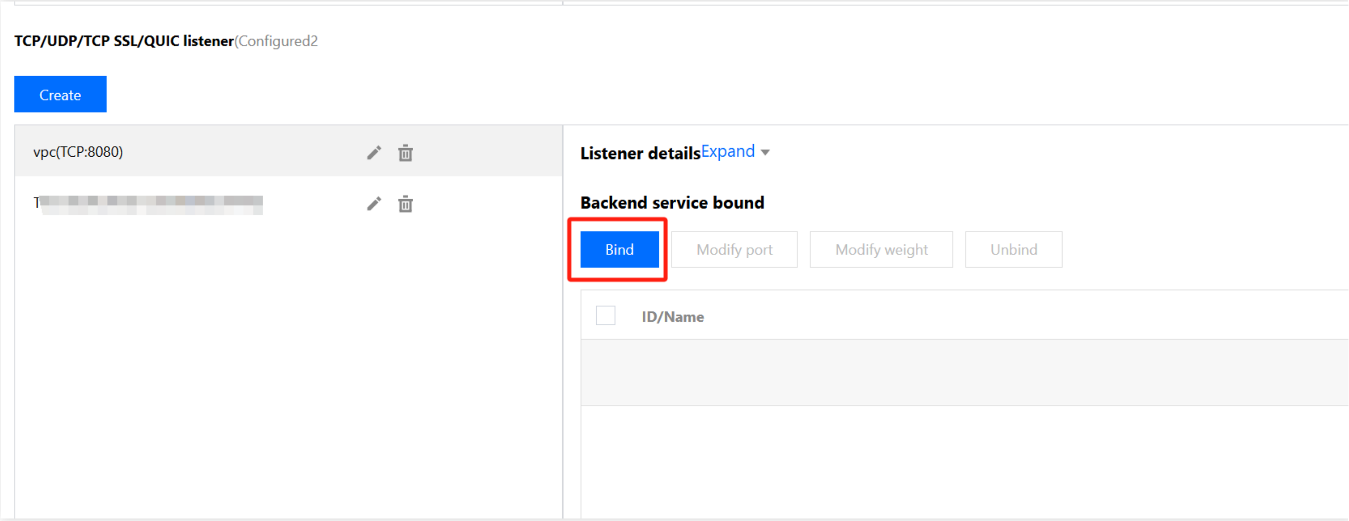
Balancing method: WRR

WRR scheduling is based on the number of new connections. The stands more chances to be polled.

Two-way RST: □ Unbind real server

If it's selected, an RST message will be sent to both the client and not, the persistent connection will stay until it's timed out

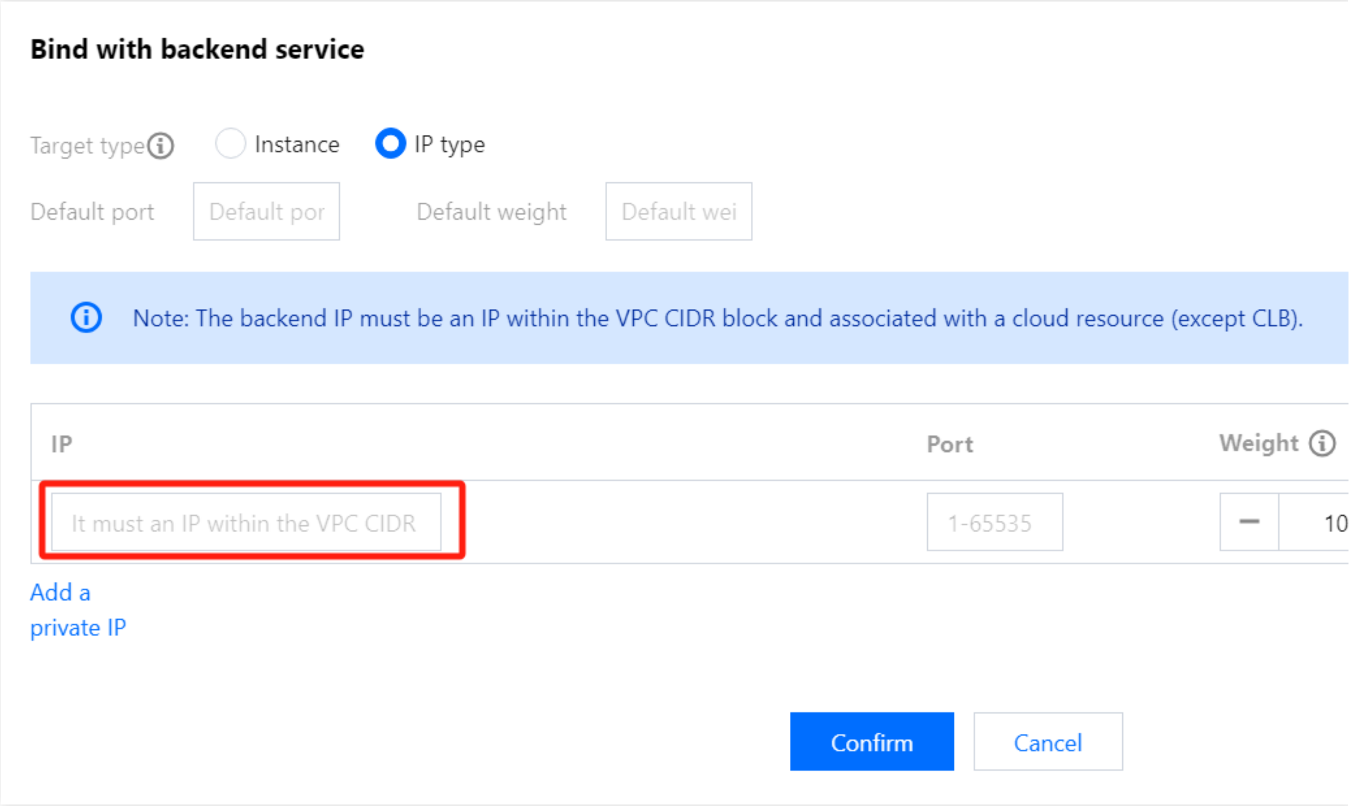
3. After creating the listener, click the listener name. In the sub-window, click **Bind** to bind the backend service.



Target Type: Select IP Type.

Default Port: Enter 9090.

IP Address: Enter the IPv4 address of the Prometheus instance obtained in step 1.



Bind with backend service

Target type ○ Instance ● IP type

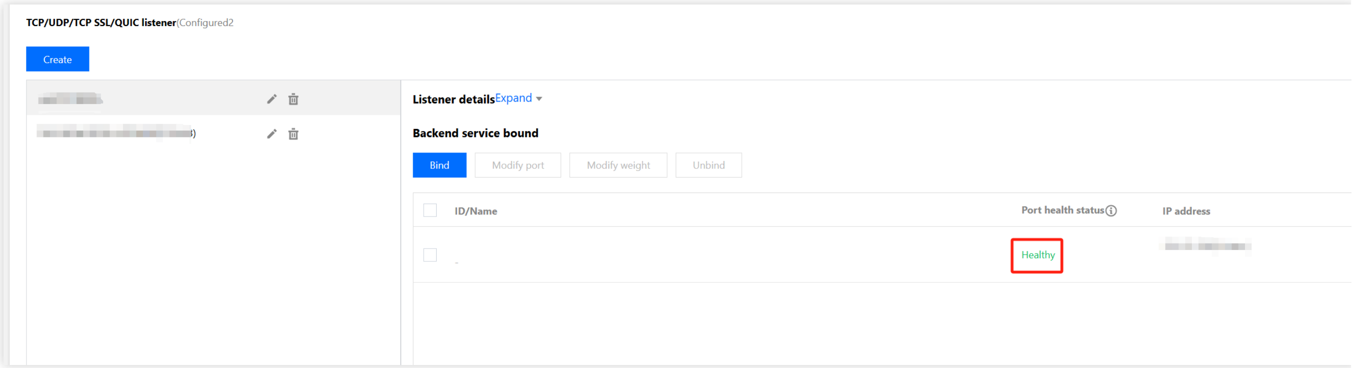
Default port: Default por

Default weight: Default wei

Note: The backend IP must be an IP within the VPC CIDR block and associated with a cloud resource (except CLB).

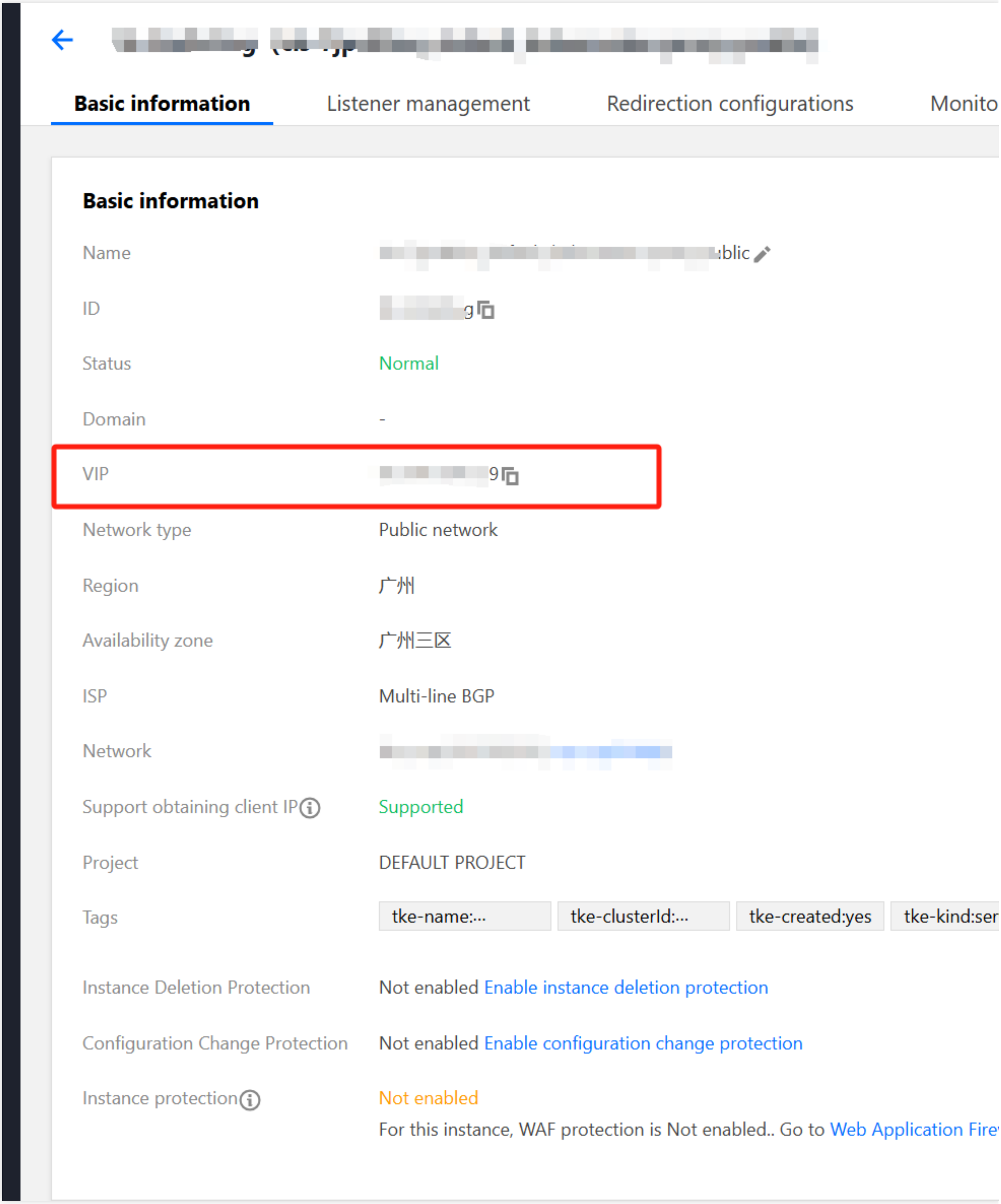
IP	Port	Weight
It must an IP within the VPC CIDR	1-65535	− 10

4. Click **Listener Name** to check whether the listening is normal.



Step 5: Testing Whether the Configuration is Successful

1. Check the public network address of CLB. Assume that the address is 192.168.1.1.



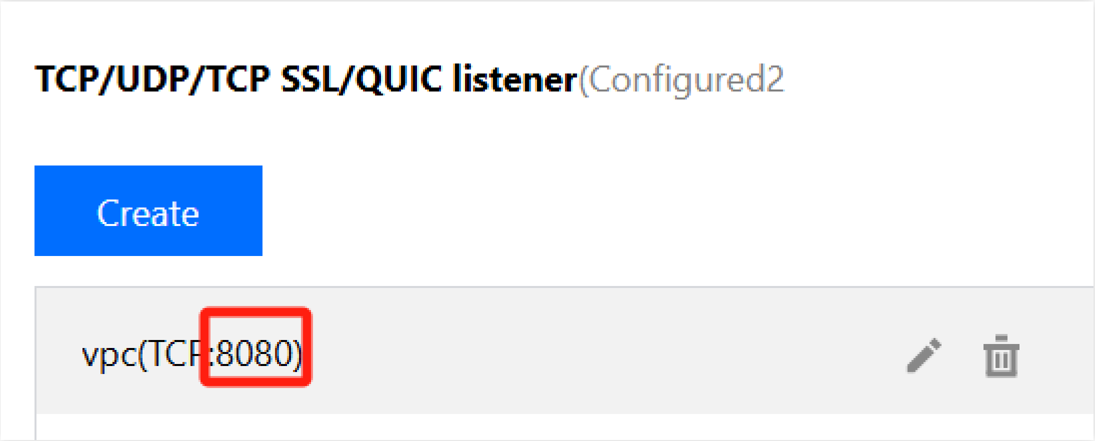
← [redacted]

Basic information | Listener management | Redirection configurations | Monito

Basic information

Name	[redacted]blic
ID	[redacted]g
Status	Normal
Domain	-
VIP	[redacted]9
Network type	Public network
Region	广州
Availability zone	广州三区
ISP	Multi-line BGP
Network	[redacted]
Support obtaining client IP ⓘ	Supported
Project	DEFAULT PROJECT
Tags	tke-name:... tke-clusterId:... tke-created:yes tke-kind:ser
Instance Deletion Protection	Not enabled Enable instance deletion protection
Configuration Change Protection	Not enabled Enable configuration change protection
Instance protection ⓘ	Not enabled For this instance, WAF protection is Not enabled.. Go to Web Application Fire

2. Check the port configured for the listener, for example, port 8080.



Based on the above information, it is determined that the public network address for Prometheus forwarding is IP:PORT in the following new address, that is, `192.168.1.1:8080` .

3. Check in the browser or on the machine to see whether UP data can be obtained by using this IP address.

4. HTTP API address:

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

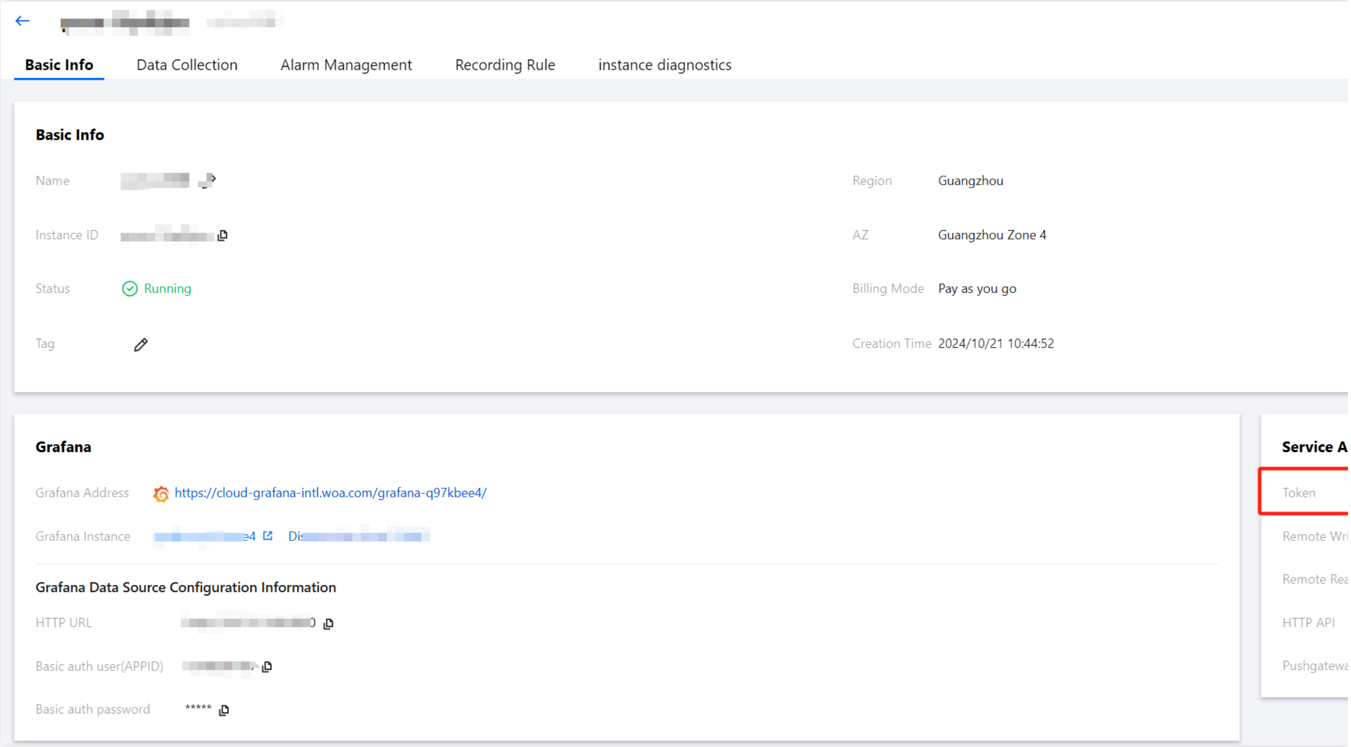
Replace IP:PORT with **Public IP and Port** of CLB as follows:

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

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

username: Enter your root account ID (app ID).

password: Obtain the token from the basic information page of the instance.



[redacted]

Basic Info | Data Collection | Alarm Management | Recording Rule | instance diagnostics

Basic Info

Name [redacted]

Region Guangzhou

Instance ID [redacted]

AZ Guangzhou Zone 4

Status Running

Billing Mode Pay as you go

Tag

Creation Time 2024/10/21 10:44:52

Grafana

Grafana Address https://cloud-grafana-intl.woa.com/grafana-q97kbee4/

Grafana Instance [redacted] [redacted]

Grafana Data Source Configuration Information

HTTP URL [redacted]

Basic auth user(APPID) [redacted]

Basic auth password *****

Service A

Token

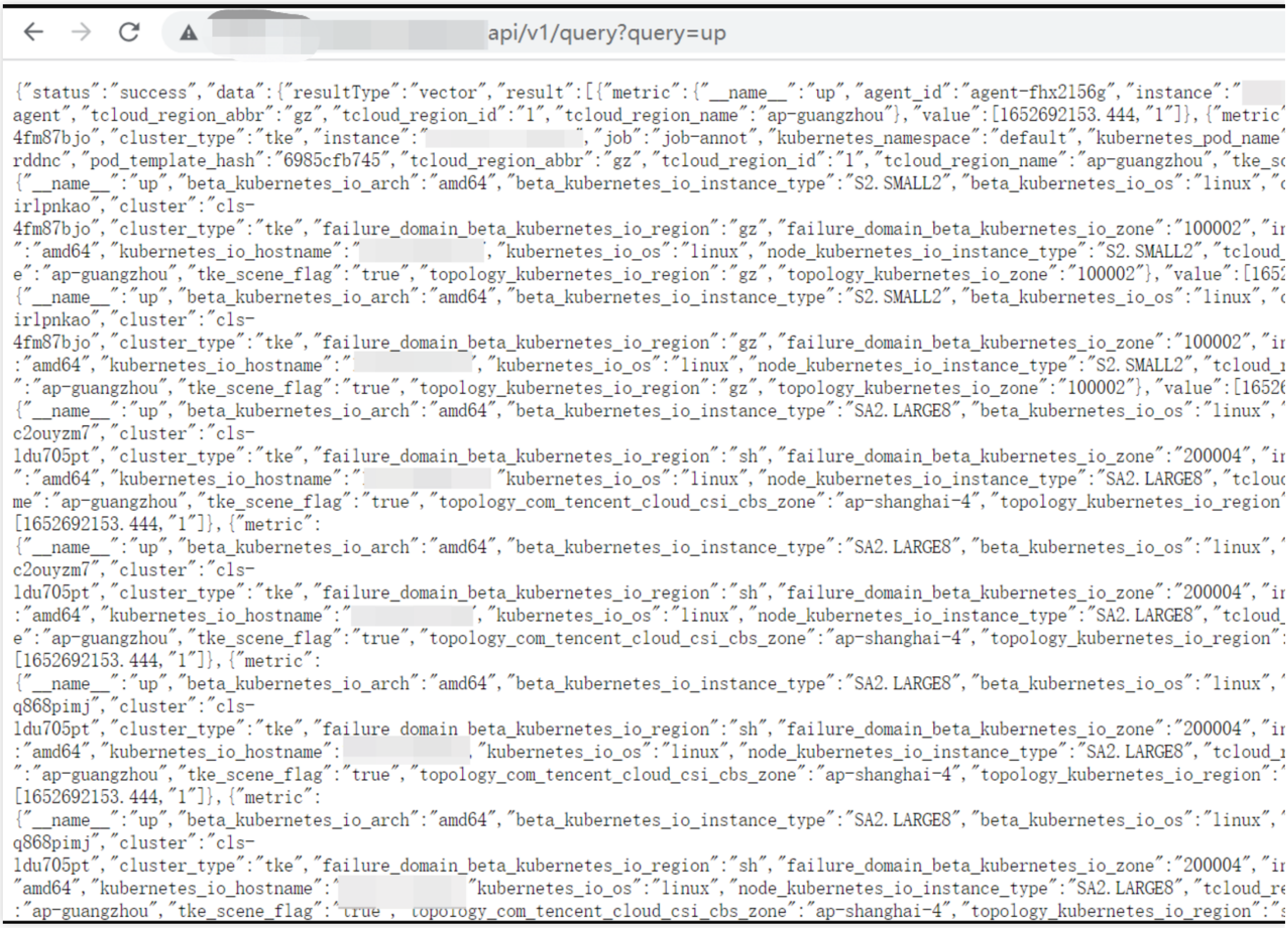
Remote Wri

Remote Rea

HTTP API

Pushgatewa

As shown below, the public network address is successfully configured for the Prometheus instance.



[redacted]api/v1/query?query=up

```
{"status":"success","data":{"resultType":"vector","result":[{"metric":{"__name__":"up","agent_id":"agent-fhx2156g","instance":"[redacted]
agent","tcloud_region_abbr":"gz","tcloud_region_id":"1","tcloud_region_name":"ap-guangzhou"},"value":[1652692153.444,"1"]},{"metric"
4fm87bjo","cluster_type":"tke","instance":"[redacted]","job":"job-annot","kubernetes_namespace":"default","kubernetes_pod_name"
rddnc","pod_template_hash":"6985cfb745","tcloud_region_abbr":"gz","tcloud_region_id":"1","tcloud_region_name":"ap-guangzhou","tke_sc
{"__name__":"up","beta_kubernetes_io_arch":"amd64","beta_kubernetes_io_instance_type":"S2.SMALL2","beta_kubernetes_io_os":"linux","c
irlpnkao","cluster":"cls-
4fm87bjo","cluster_type":"tke","failure_domain_beta_kubernetes_io_region":"gz","failure_domain_beta_kubernetes_io_zone":"100002","ir
":"amd64","kubernetes_io_hostname":"[redacted]","kubernetes_io_os":"linux","node_kubernetes_io_instance_type":"S2.SMALL2","tcloud_
e":"ap-guangzhou","tke_scene_flag":"true","topology_kubernetes_io_region":"gz","topology_kubernetes_io_zone":"100002"},"value":[1652
{"__name__":"up","beta_kubernetes_io_arch":"amd64","beta_kubernetes_io_instance_type":"S2.SMALL2","beta_kubernetes_io_os":"linux","c
irlpnkao","cluster":"cls-
4fm87bjo","cluster_type":"tke","failure_domain_beta_kubernetes_io_region":"gz","failure_domain_beta_kubernetes_io_zone":"100002","ir
:"amd64","kubernetes_io_hostname":"[redacted]","kubernetes_io_os":"linux","node_kubernetes_io_instance_type":"S2.SMALL2","tcloud_r
":"ap-guangzhou","tke_scene_flag":"true","topology_kubernetes_io_region":"gz","topology_kubernetes_io_zone":"100002"},"value":[16526
{"__name__":"up","beta_kubernetes_io_arch":"amd64","beta_kubernetes_io_instance_type":"SA2.LARGE8","beta_kubernetes_io_os":"linux","
c2ouyzm7","cluster":"cls-
ldu705pt","cluster_type":"tke","failure_domain_beta_kubernetes_io_region":"sh","failure_domain_beta_kubernetes_io_zone":"200004","ir
":"amd64","kubernetes_io_hostname":"[redacted] "kubernetes_io_os":"linux","node_kubernetes_io_instance_type":"SA2.LARGE8","tclouc
me":"ap-guangzhou","tke_scene_flag":"true","topology_com_tencent_cloud_csi_cbs_zone":"ap-shanghai-4","topology_kubernetes_io_region"
[1652692153.444,"1"]},{"metric":
{"__name__":"up","beta_kubernetes_io_arch":"amd64","beta_kubernetes_io_instance_type":"SA2.LARGE8","beta_kubernetes_io_os":"linux","
c2ouyzm7","cluster":"cls-
ldu705pt","cluster_type":"tke","failure_domain_beta_kubernetes_io_region":"sh","failure_domain_beta_kubernetes_io_zone":"200004","ir
:"amd64","kubernetes_io_hostname":"[redacted]","kubernetes_io_os":"linux","node_kubernetes_io_instance_type":"SA2.LARGE8","tcloud_
e":"ap-guangzhou","tke_scene_flag":"true","topology_com_tencent_cloud_csi_cbs_zone":"ap-shanghai-4","topology_kubernetes_io_region":
[1652692153.444,"1"]},{"metric":
{"__name__":"up","beta_kubernetes_io_arch":"amd64","beta_kubernetes_io_instance_type":"SA2.LARGE8","beta_kubernetes_io_os":"linux","
q868pimj","cluster":"cls-
ldu705pt","cluster_type":"tke","failure domain beta_kubernetes_io_region":"sh","failure_domain_beta_kubernetes_io_zone":"200004","ir
:"amd64","kubernetes_io_hostname":[redacted],"kubernetes_io_os":"linux","node_kubernetes_io_instance_type":"SA2.LARGE8","tcloud_r
":"ap-guangzhou","tke_scene_flag":"true","topology_com_tencent_cloud_csi_cbs_zone":"ap-shanghai-4","topology_kubernetes_io_region":"
[1652692153.444,"1"]},{"metric":
{"__name__":"up","beta_kubernetes_io_arch":"amd64","beta_kubernetes_io_instance_type":"SA2.LARGE8","beta_kubernetes_io_os":"linux","
q868pimj","cluster":"cls-
ldu705pt","cluster_type":"tke","failure_domain_beta_kubernetes_io_region":"sh","failure_domain_beta_kubernetes_io_zone":"200004","ir
"amd64","kubernetes_io_hostname":"[redacted]"kubernetes_io_os":"linux","node_kubernetes_io_instance_type":"SA2.LARGE8","tcloud_re
:"ap-guangzhou","tke_scene_flag":"true","topology_com_tencent_cloud_csi_cbs_zone":"ap-shanghai-4","topology_kubernetes_io_region":"s
```