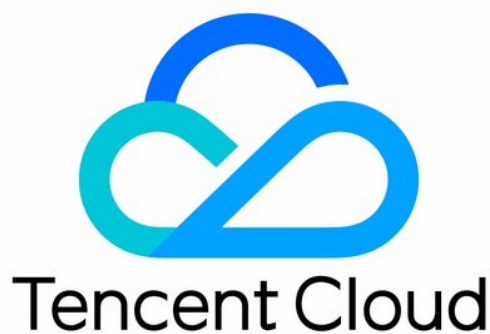


TencentCloud Managed Service for Prometheus Best Practice Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

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

Best Practice

Migration from Self-Built Prometheus

Custom Integration with CVM

TKE Monitoring

Enabling Public Network Access for TKE Serverless Cluster

Connecting TMP to Local Grafana

Best Practice

Migration from Self-Built Prometheus

Last updated : 2024-01-29 15:55:07

Overview




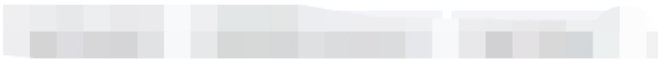

You can quickly migrate from your self-built Prometheus service to TMP.


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

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


IPv4 Address 



Grafana Status ☐ Disabled



Billing Mode Trial Edition


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

Service Address

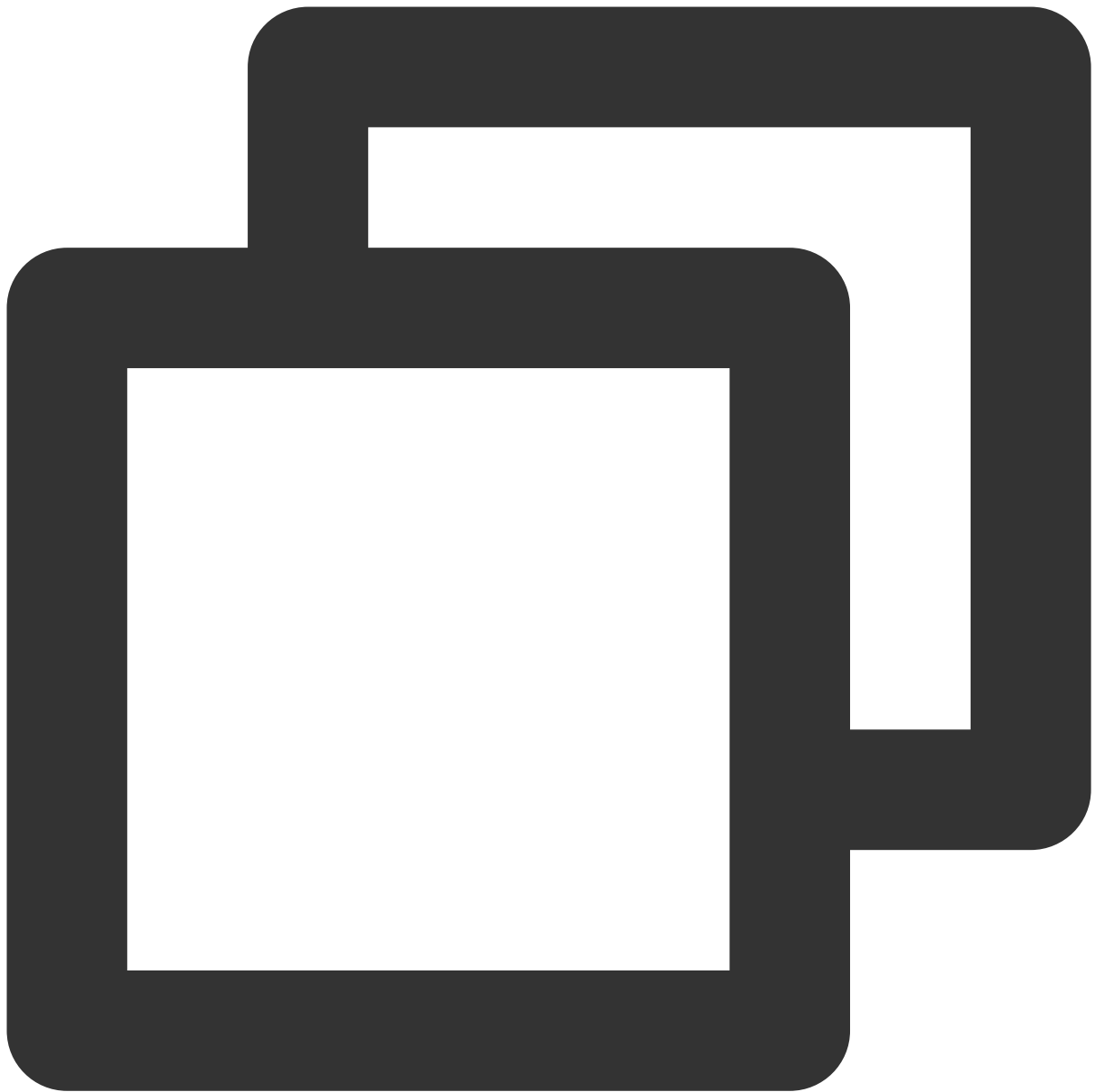
Token ***** 

Remote Write Address http  pi/v1/prom/write 

HTTP API ht  pi/v1 

Pushgateway Address 

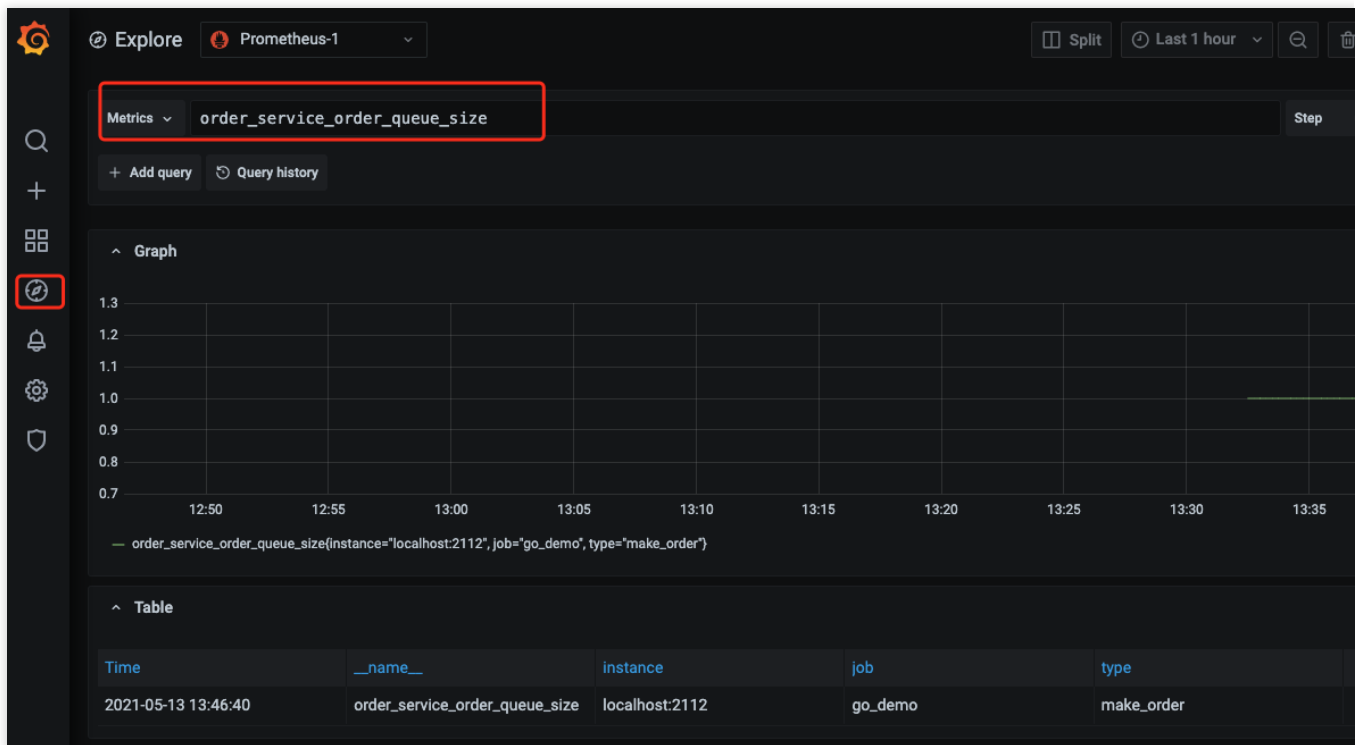
2. Modify `prometheus.yml` and restart Prometheus. The specific configuration is as follows. For more information on the remote write configuration parameters, please see [remote_write](#).



```
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](#).



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

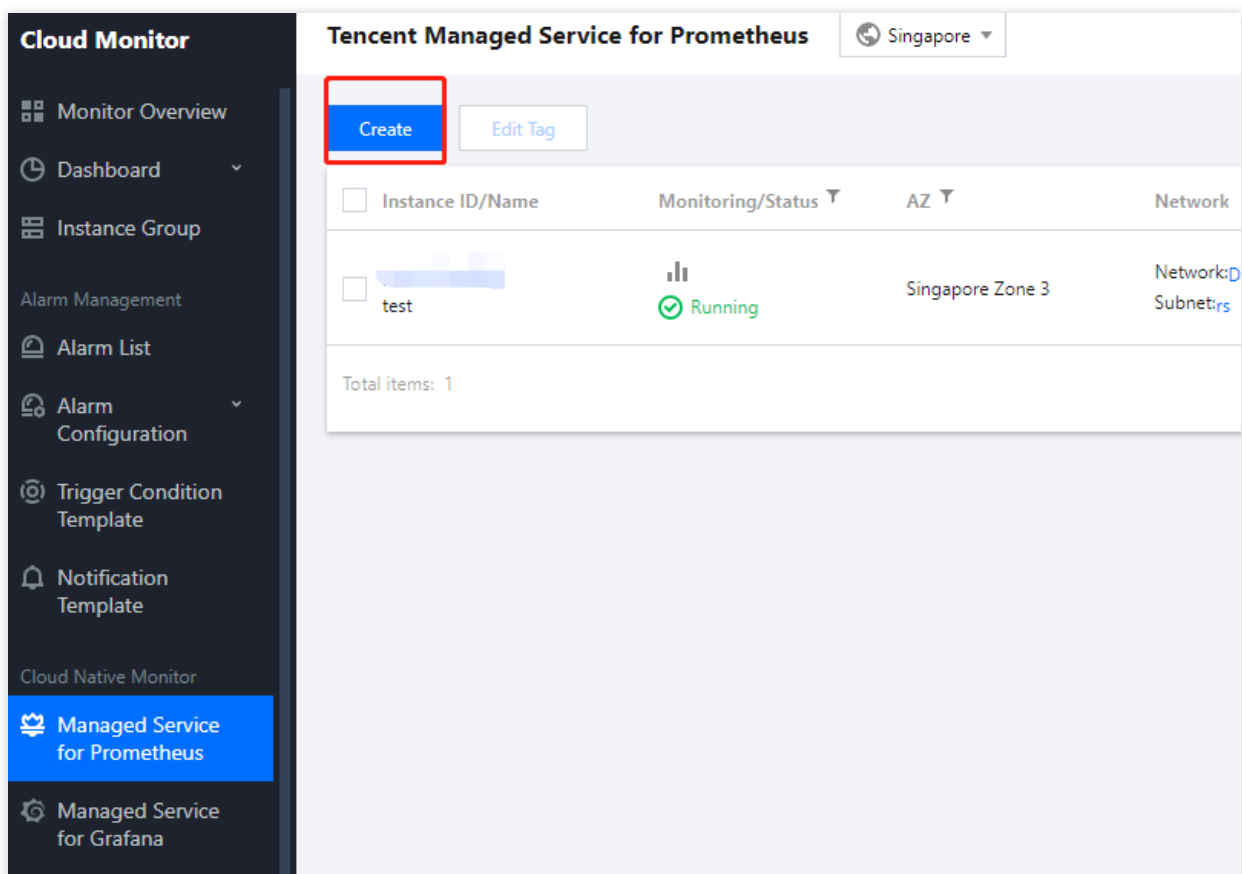
This document describes how to integrate CVM with TMP.

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.



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 Doc

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. Please check the region after purchasing the instance.

AZ

Singapore Zone 1

Singapore Zone 2

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

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.

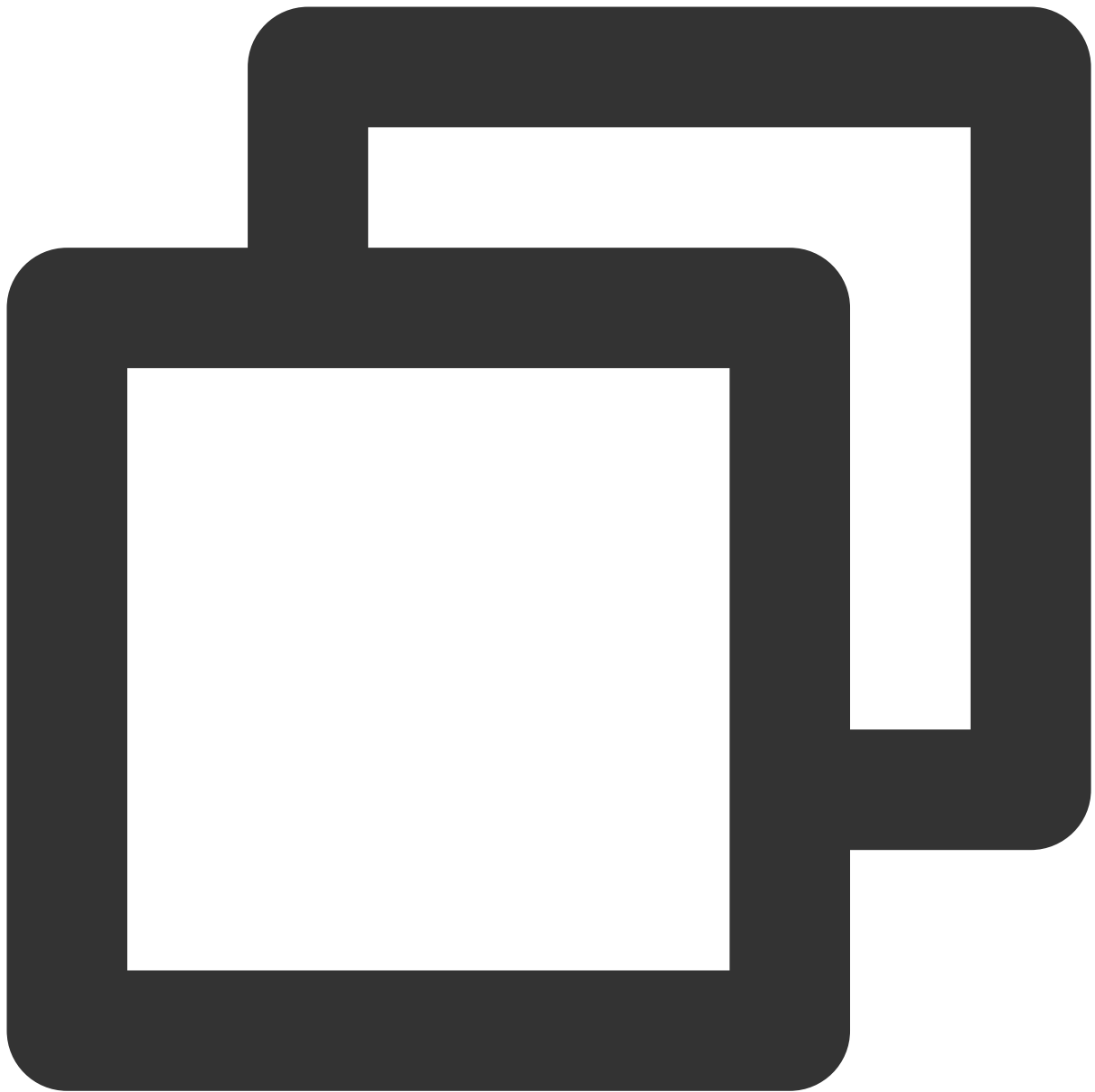
Integrating CVM Basic Metrics

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:

©2013-2022 Tencent Cloud. All rights reserved.

Page 9 of 33



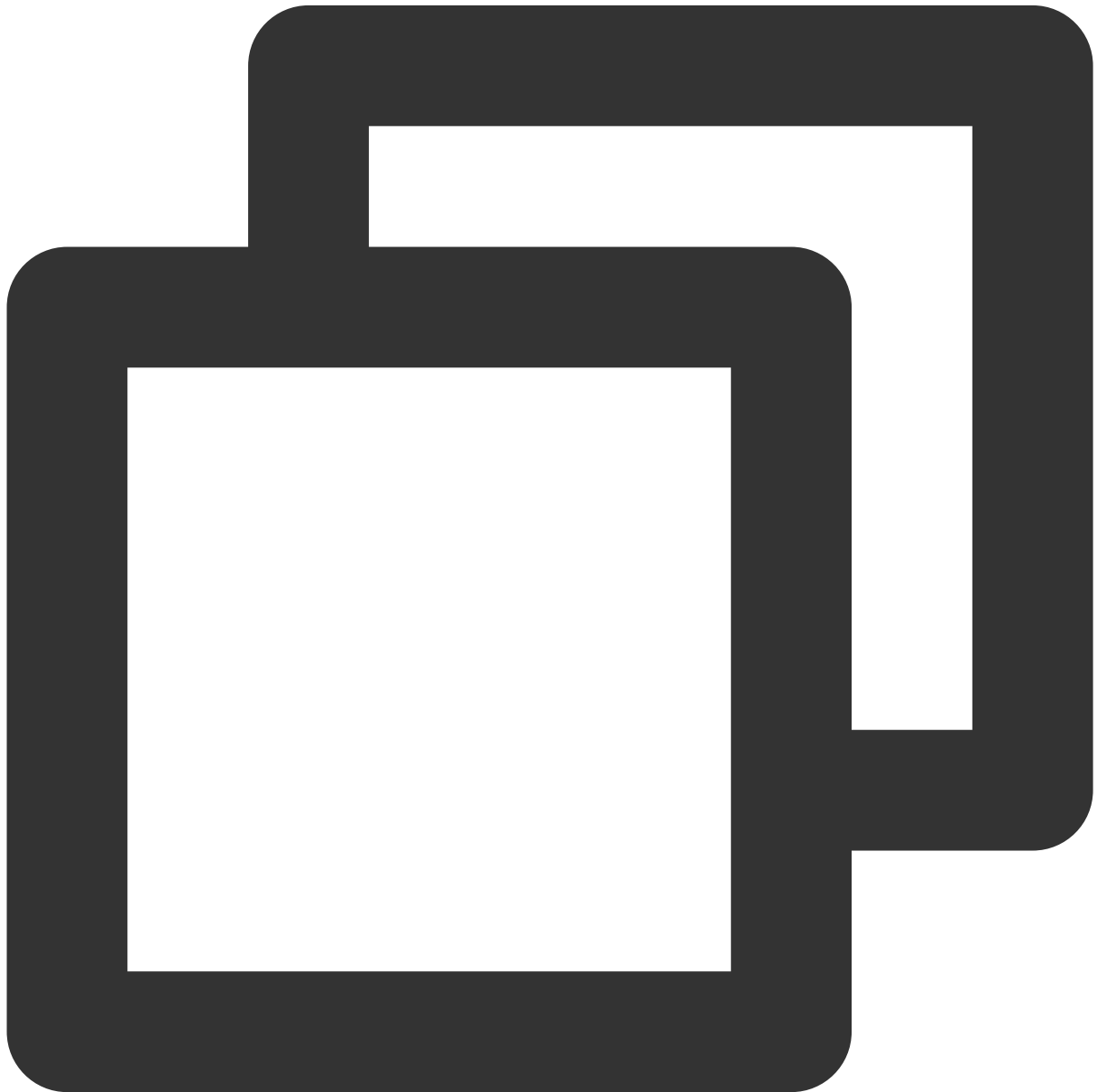
```
wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_expo
```

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.

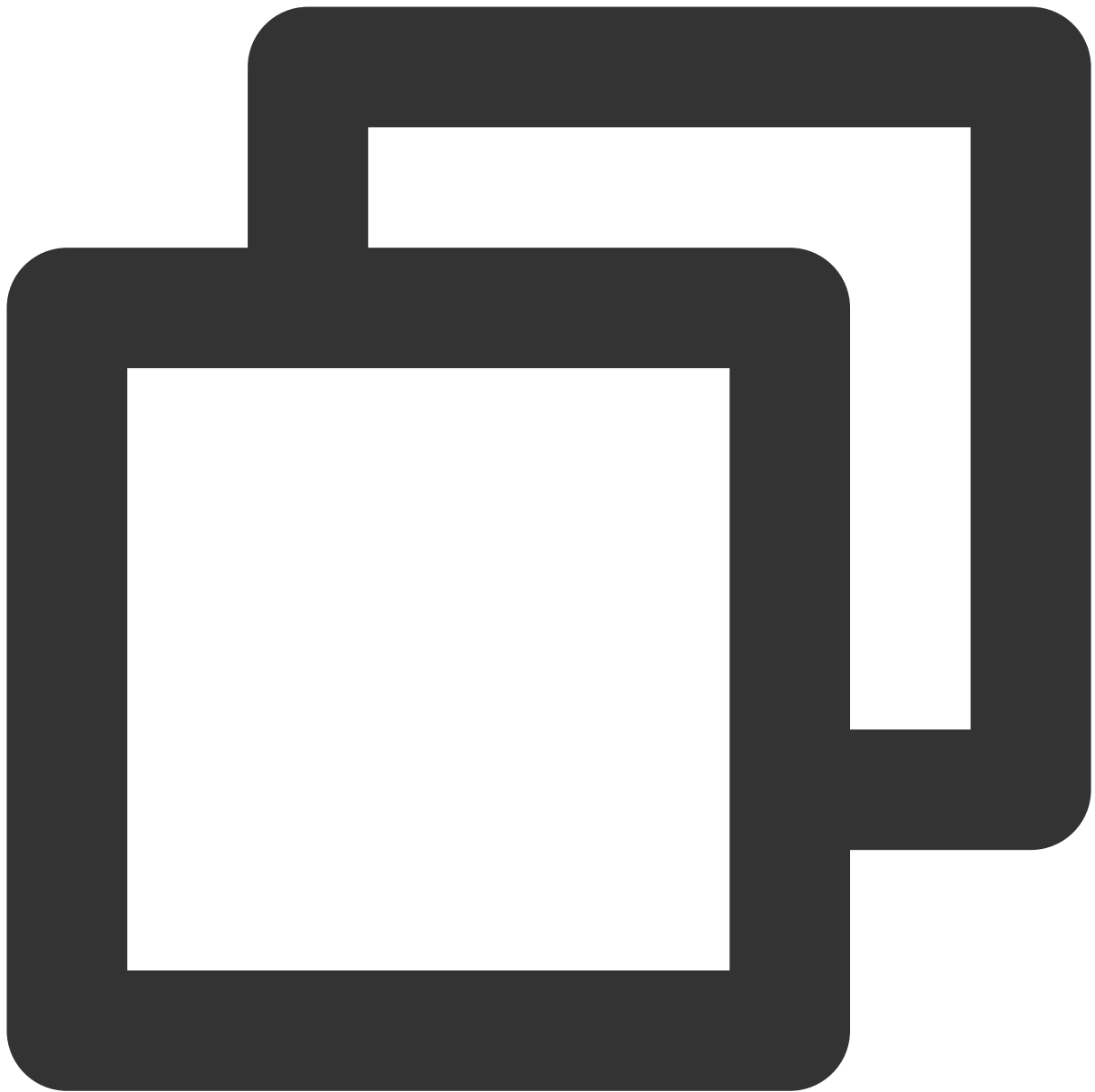


```
cd node_exporter-1.3.1.linux-amd64
./node_exporter
```

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, build=26645363b486e12be40af7ce4fc91e731a33104e)"
level=info ts=2022-02-11T07:15:26.555Z caller=node_exporter.go:183 msg="Build context" build_context="(go=go1.16.7, user=root@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 export
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
nts-exclude" flag=~/(dev|proc|sys|var/lib/docker/.+)($/|)
level=info ts=2022-02-11T07:15:26.555Z caller=filesystem_common.go:112 collector=filesystem msg="Parsed flag --collecto
-exclude" flag=^(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue|nsfs
store|rpc_pipefs|securityfs|selinuxfs|squashfs|sysfs|tracefs)$
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
```

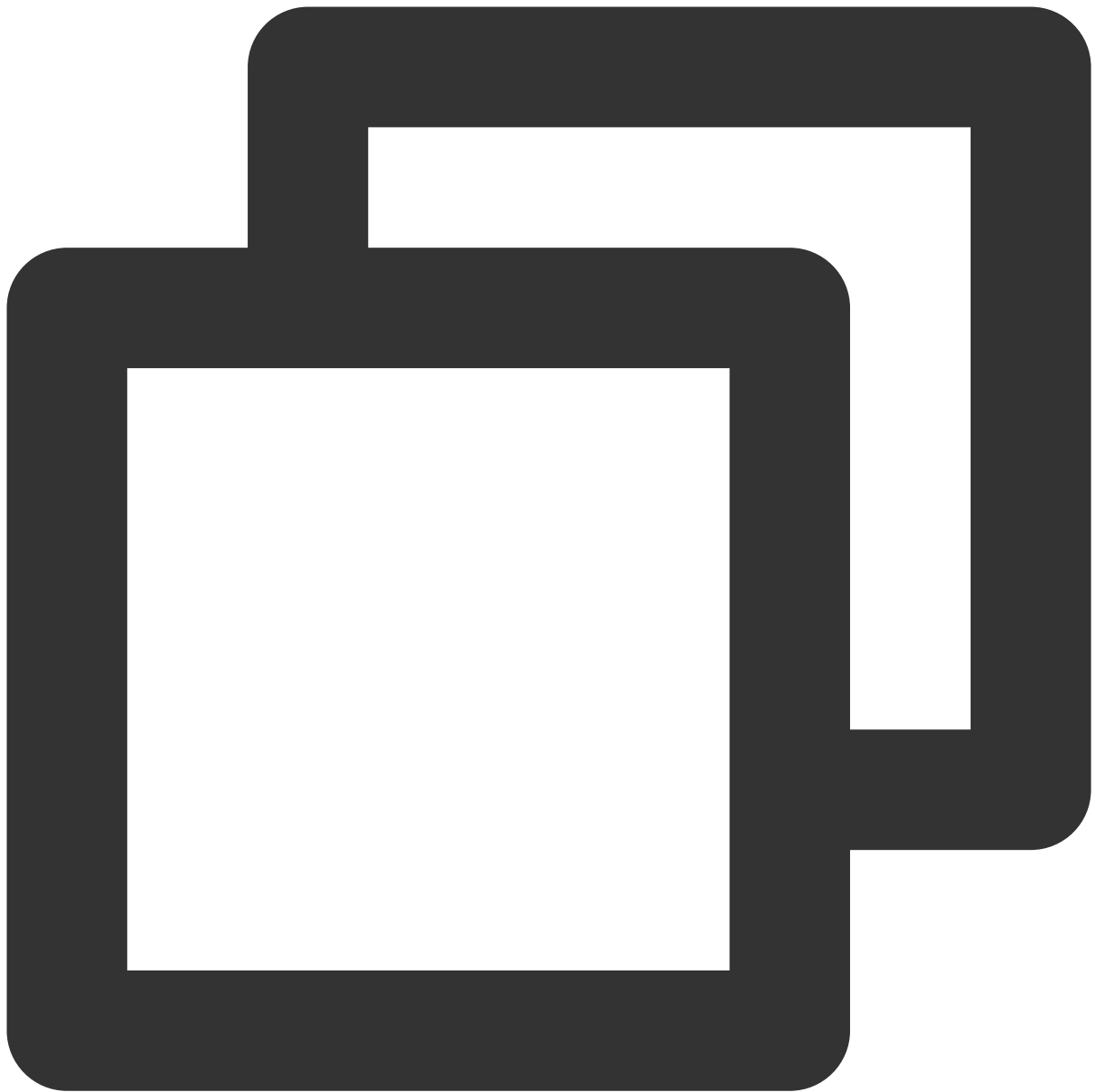
You can see the following metric monitoring data that is exposed after the command is executed.

```
[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 since
# 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:



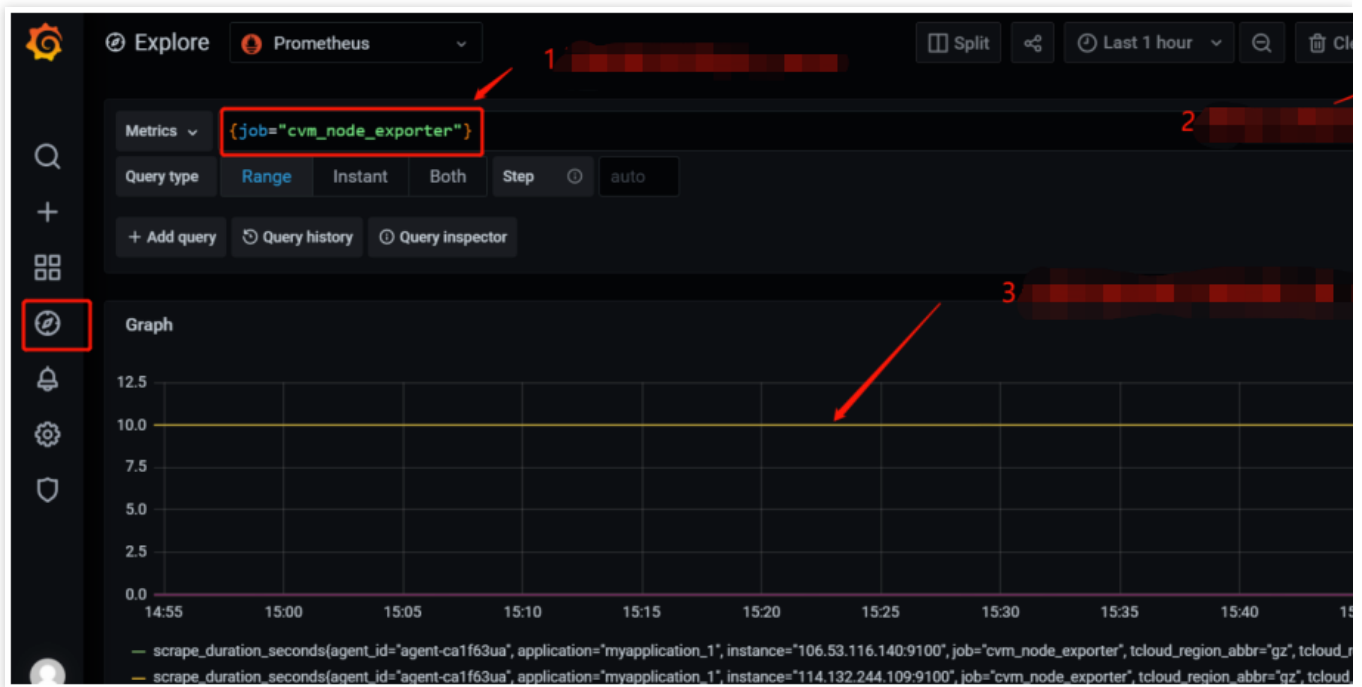
```
job_name: example-job-name
metrics_path: /metrics
cvm_sd_configs:
- region: ap-guangzhou
  ports:
  - 9100
  filters:
  - name: tag:Sample tag key
    values:
    - Sample tag value
relabel_configs:
```

```
- 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.

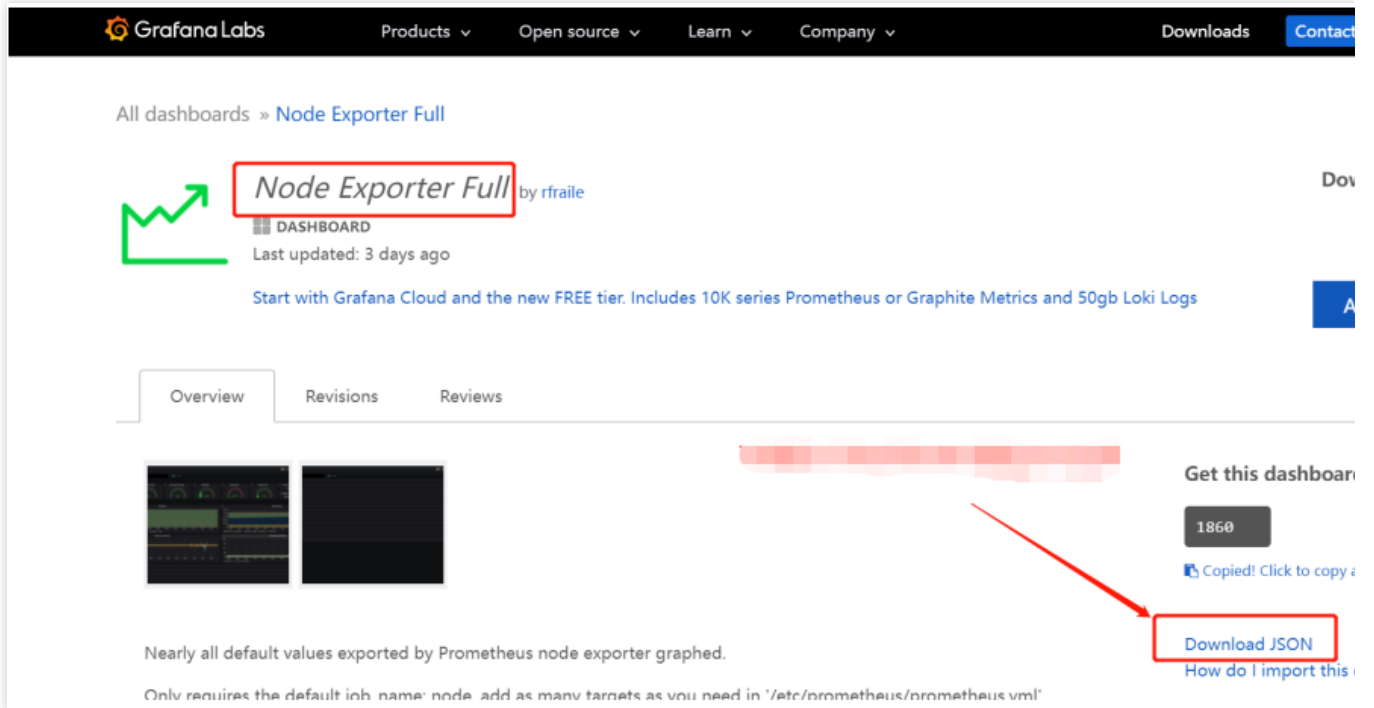
Log in to the [TMP console](#) and click the Grafana icon to enter Grafana.

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

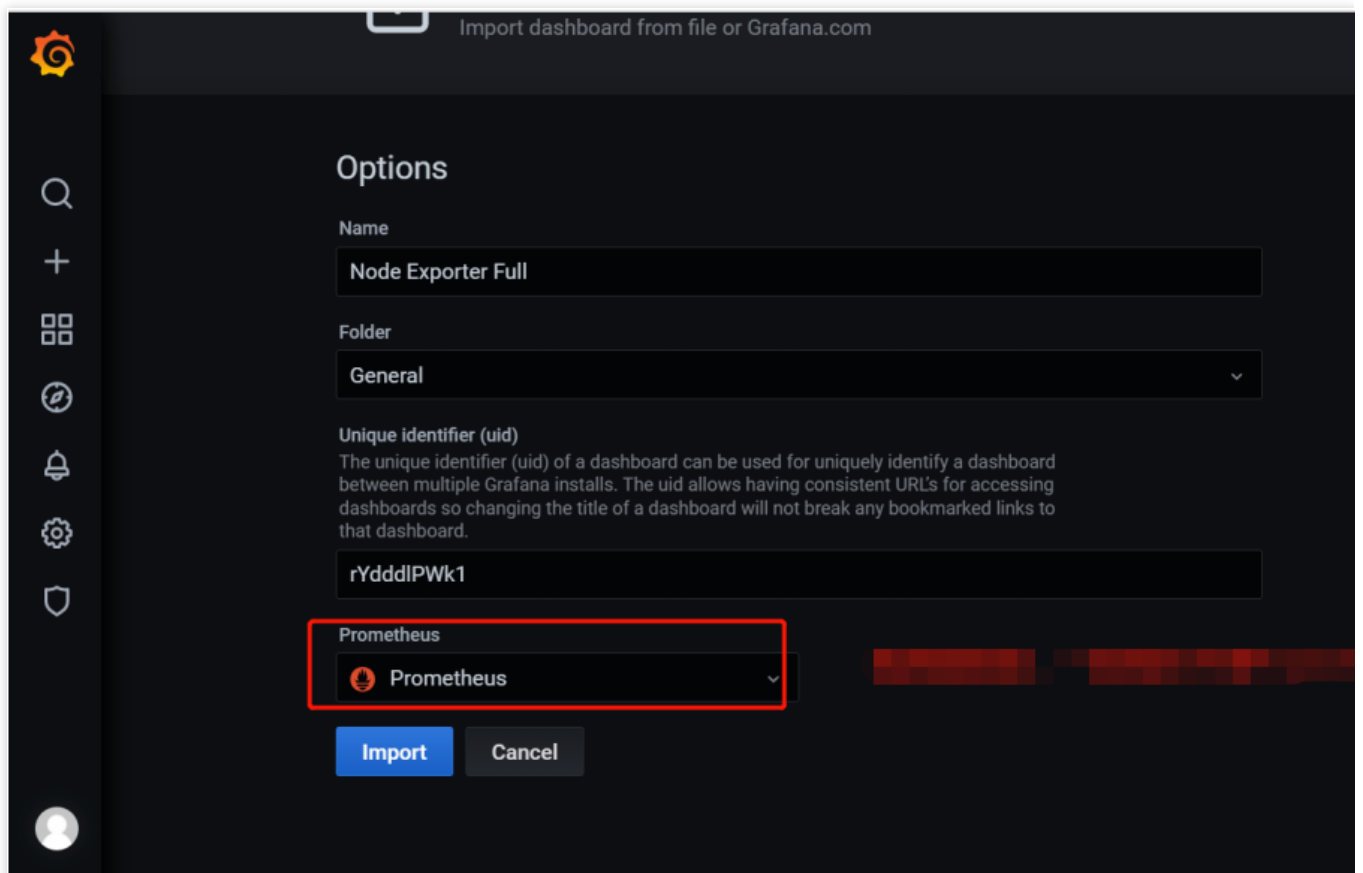


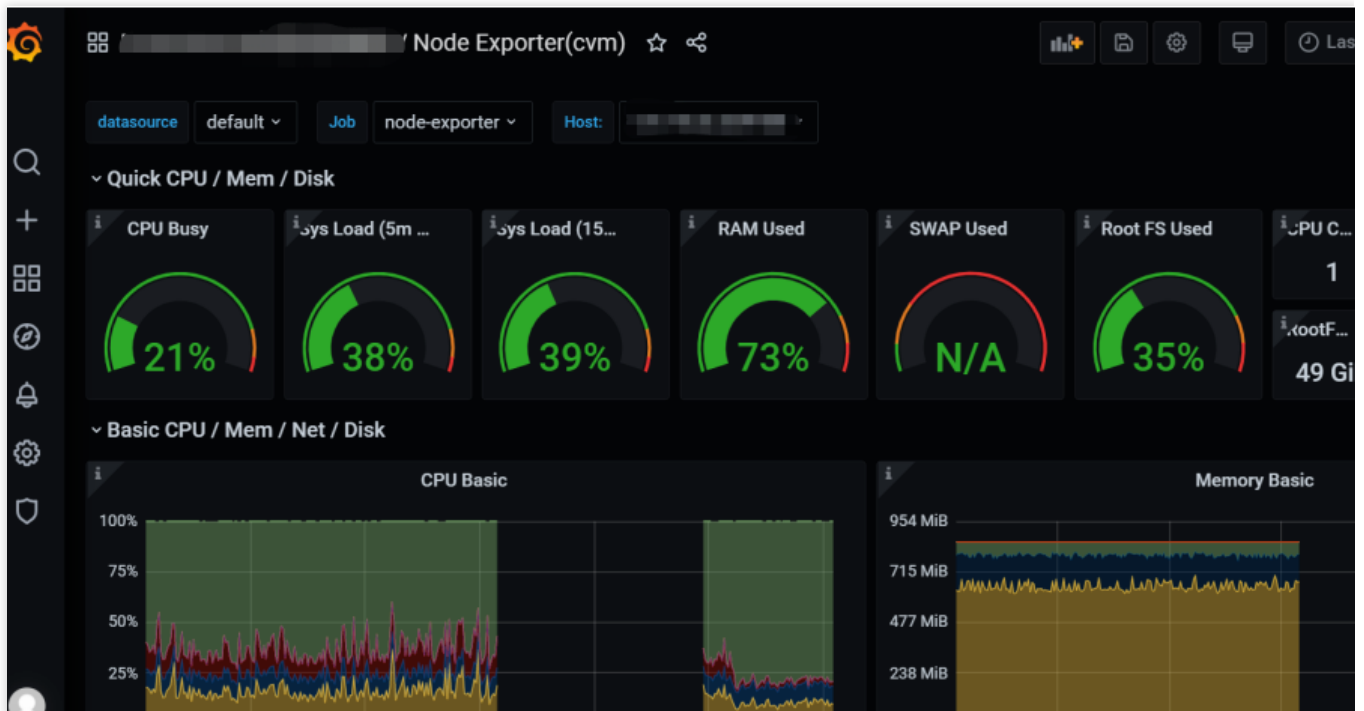
5. Configure the dashboard page: Every product has some existing JSON files that can be directly imported into the dashboard.

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



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**.





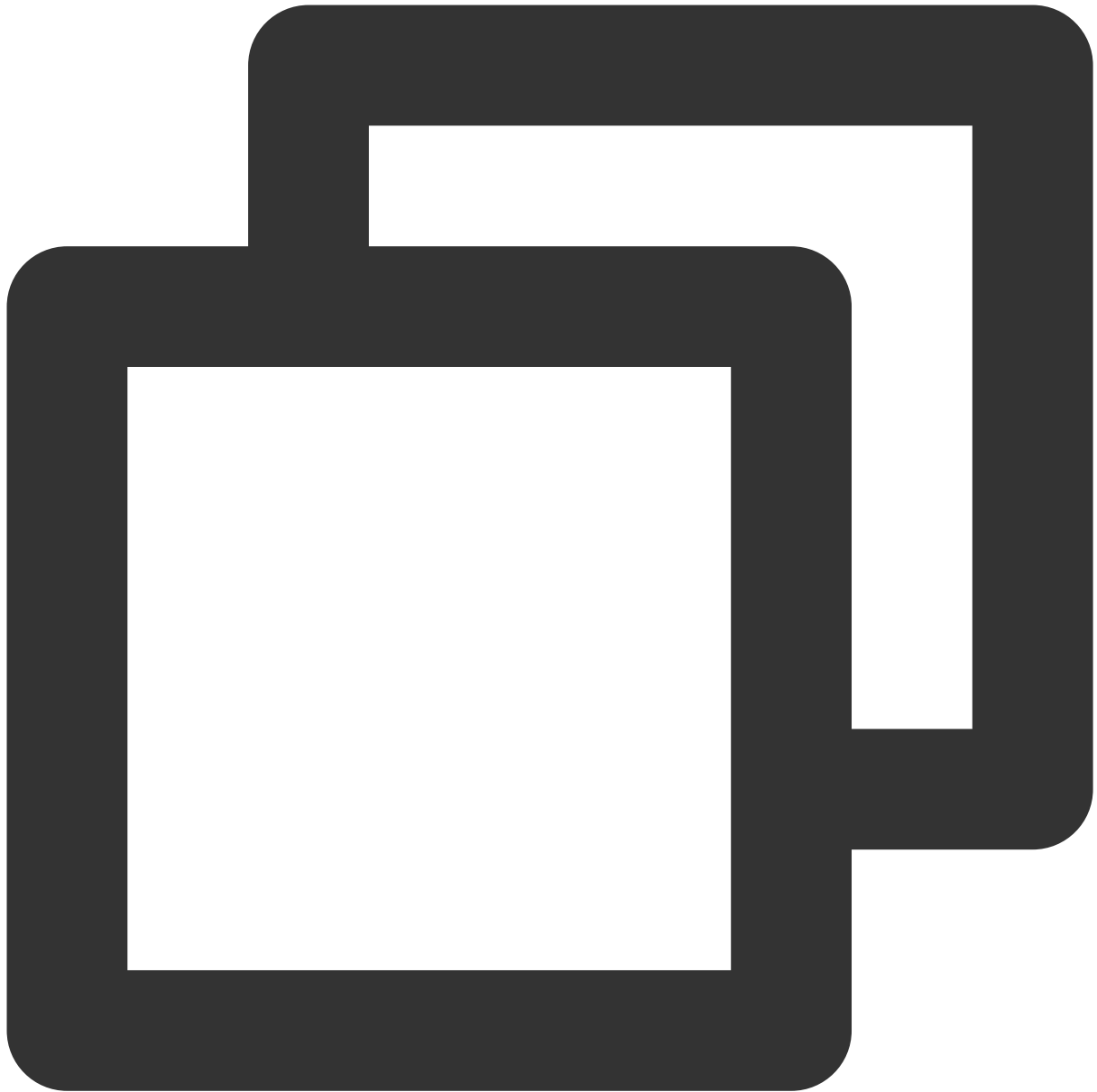
Integrating CVM Metrics at the Business Layer

Prometheus provides four metric types for different monitoring scenarios: Counter, Gauge, Histogram, and Summary. The Prometheus community provides SDKs for multiple programming 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"
)

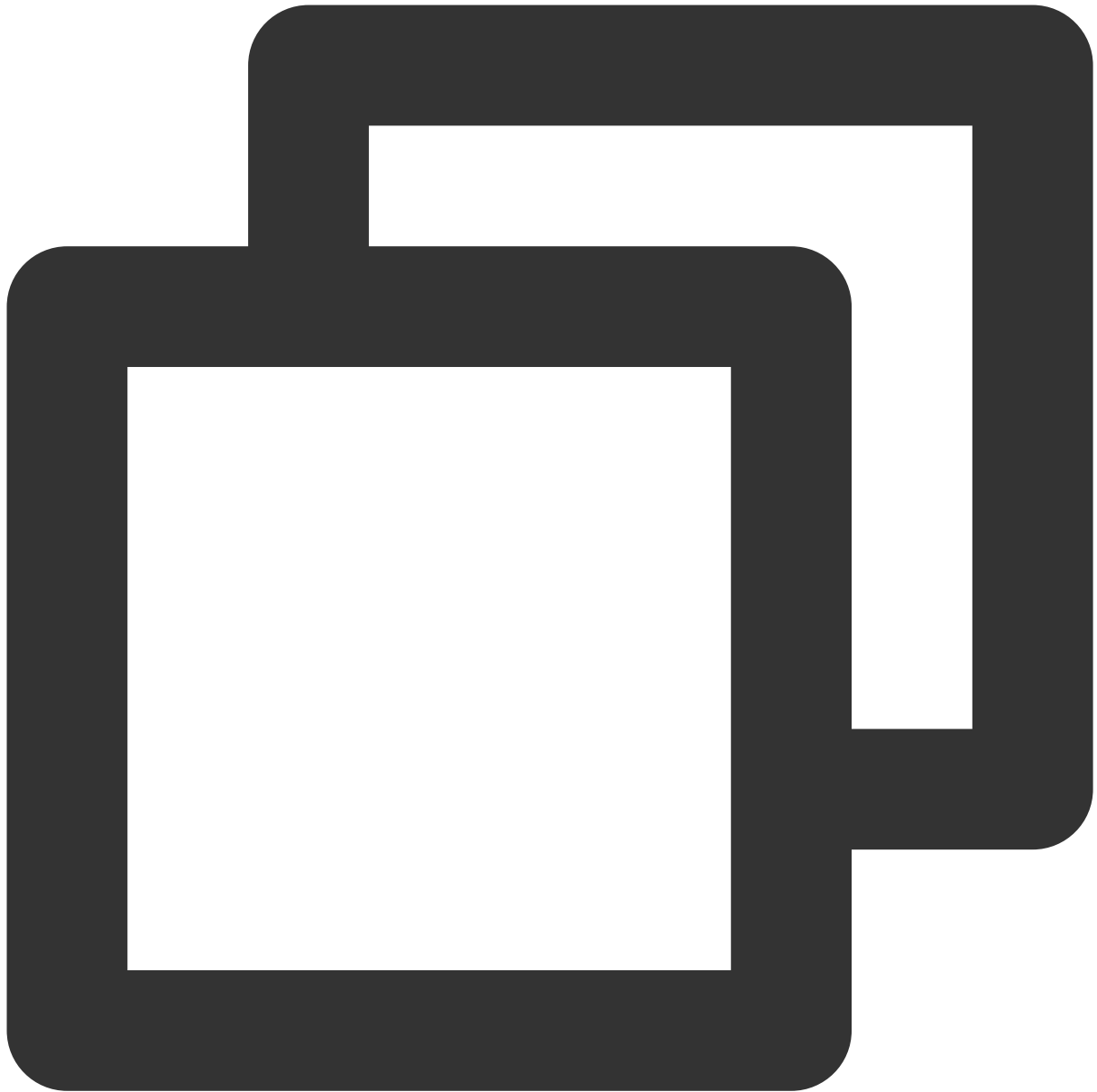
// Define the counter object to be monitored
var (
    opsProcessed = promauto.NewCounterVec(prometheus.CounterOpts{
        Name: "order_service_processed_orders_total",
```

```
    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
}
```

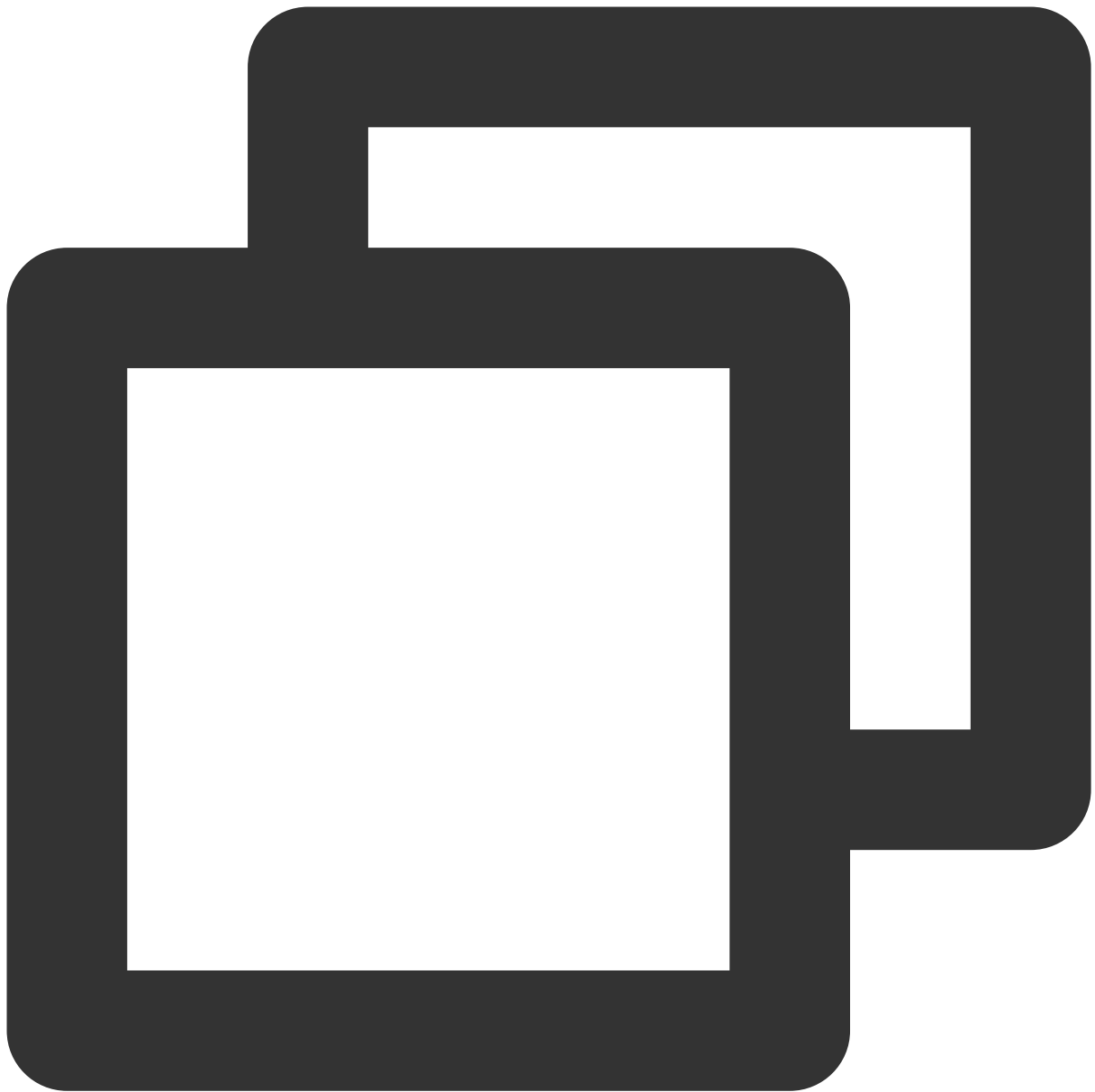
For example, you can use the `rate()` function to get the order increase rate:



```
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"
)

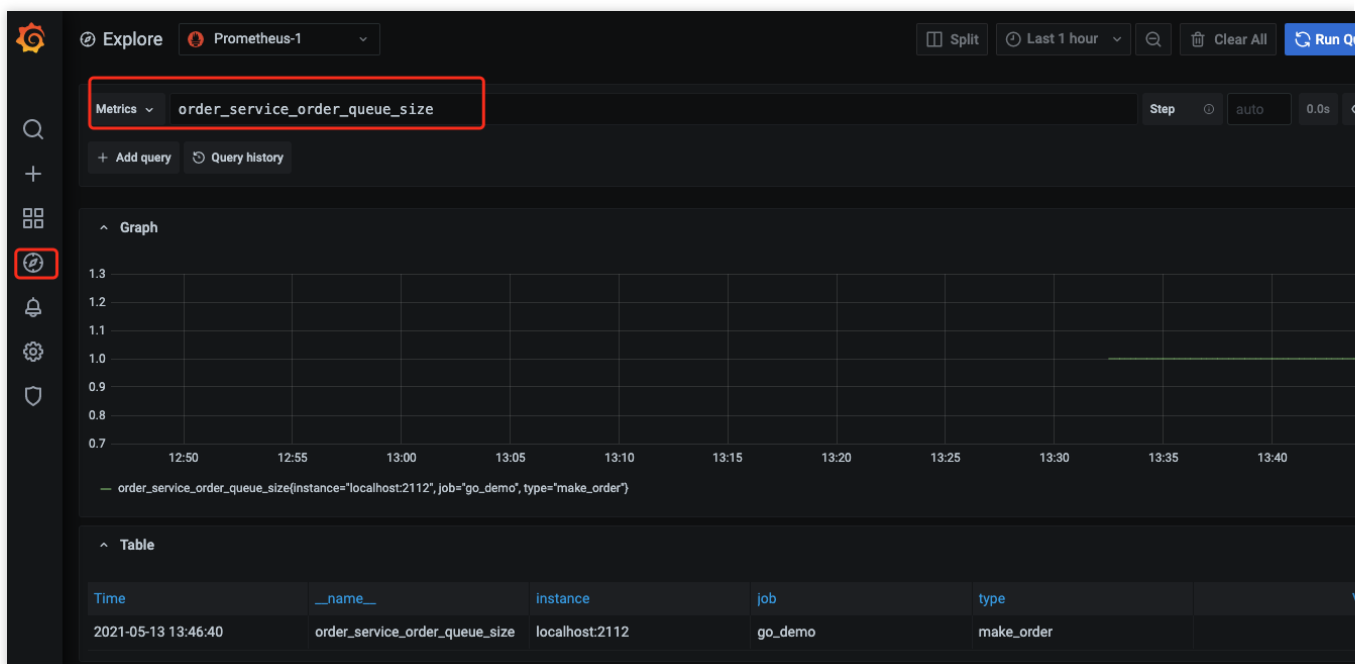
func main() {
    // Business code
}
```

```
// 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.



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
[Region]

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 loaded 1 item selected

Separate filters with carriage return

Node ID/Na...	Type	VPC	Status
<input checked="" type="checkbox"/>			Running
<input type="checkbox"/>			
<input type="checkbox"/>			

Node ID/N...	Type	VPC	Status

Press and hold Shift key to select more

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

Global label ①
☐ 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.

Confirm Cancel

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.

Basic monitoring Custom monitoring

Instance type	Filtered 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](#).

Enter the metric name

Metric name

Free 0...

Real-time coll...

Metric collection rate before filtering

Metric collection rate...

<input checked="" type="checkbox"/>	kube_node_status_allocatable_memory_bytes	Yes	Collected	0.07/sec	0.07/sec
<input checked="" type="checkbox"/>	kube_pod_owner	Yes	Collected	1.6/sec	1.6/sec
<input checked="" type="checkbox"/>	kube_replicaset_owner	Yes	Collected	0.6/sec	0.6/sec
<input type="checkbox"/>	kube_validatingwebhookconfiguration_metadata_reso...	No	Not collected	0.07/sec	0/sec
<input type="checkbox"/>	kube_job_owner	No	Not collected	0.27/sec	0/sec
<input type="checkbox"/>	kube_statefulset_status_update_revision	No	Not collected	0.07/sec	0/sec
<input type="checkbox"/>	kube_deployment_status_replicas_updated	No	Not collected	0.6/sec	0/sec
<input type="checkbox"/>	kube_hpa_spec_min_replicas	No	Not collected	0.07/sec	0/sec

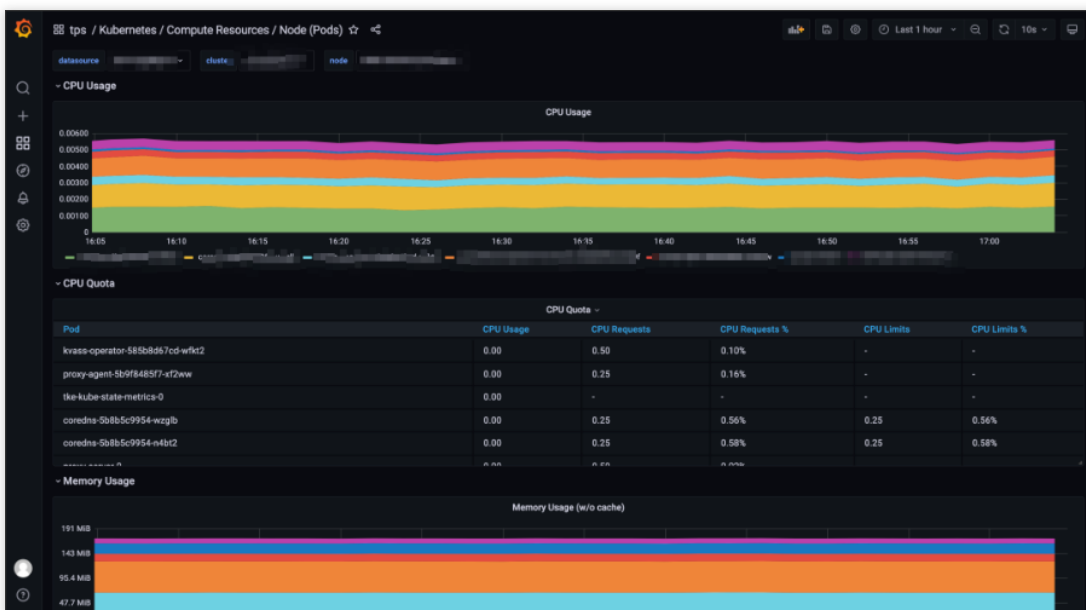
Confirm Cancel

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.



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



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.

Basic Info

Instance

Monitoring

Agent Management

Integrate with TKE ID

Integration Center

Recording Rule

Alerting Rule

Alert Manager

Alerting Rule / Create

User Guide

Rule Template

Please select a policy template

Rule Name

PromQL-Based Rule

Duration

Alert Notification Cycle

Alert Object

Alert Message

Labels

Annotations

Alert Notification

Save

Cancel

MySQL

Kubernetes

Kubernetes Masters

Kubernetes Nodes

Kubernetes Resources

Kubernetes Workload

HealthCheck

Redis

CVM

ClickHouse

Key: Please enter

Value: Please enter

Save

Key: Please enter

Value: Please enter

Save

Select Template

Create ID

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 扫码加技术交流群

Integration Cent

Integration Center

Integration list






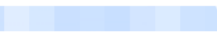




Targets

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

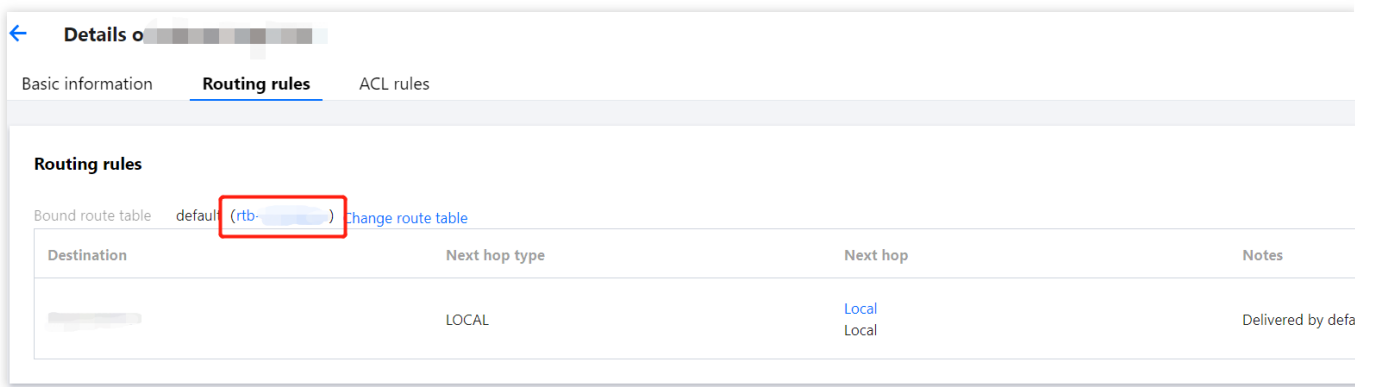
4. On the topbar, switch to the Pod management page. Click the instance name to enter the cluster page.
5. On the **Basic Info** page, click **Container Network**.

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.

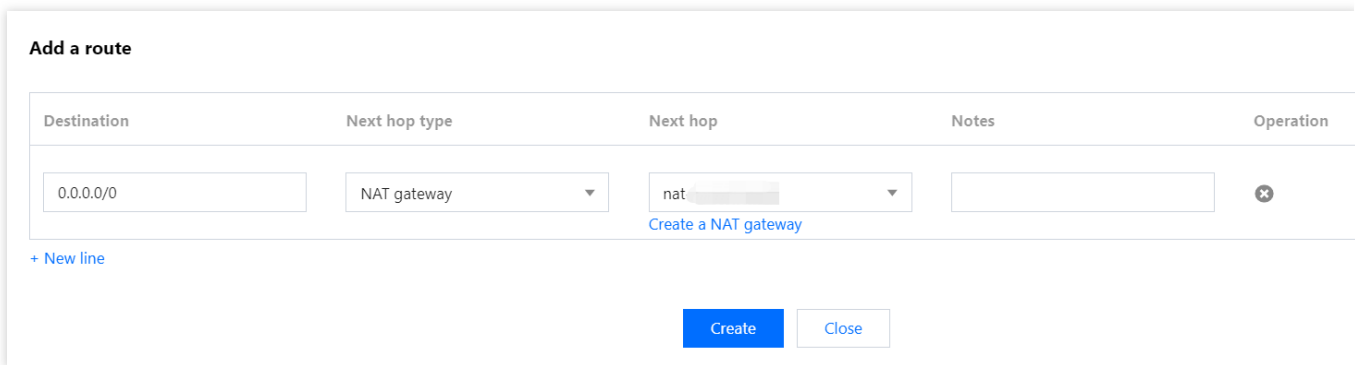


7. On the route table page, click **Create Routing Policy**.

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](#).



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

Connecting TMP to Local Grafana

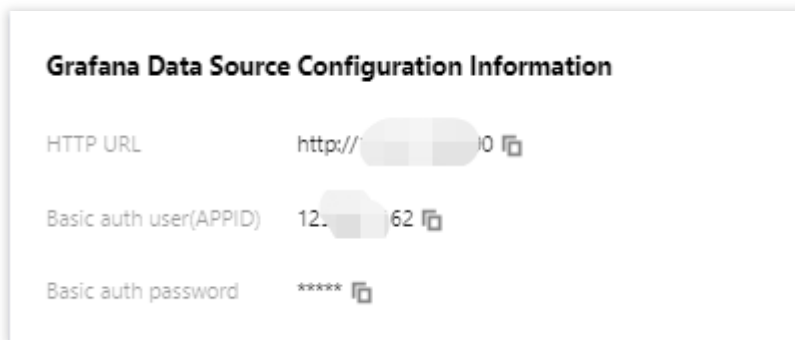
Last updated : 2024-01-29 15:55:08

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.

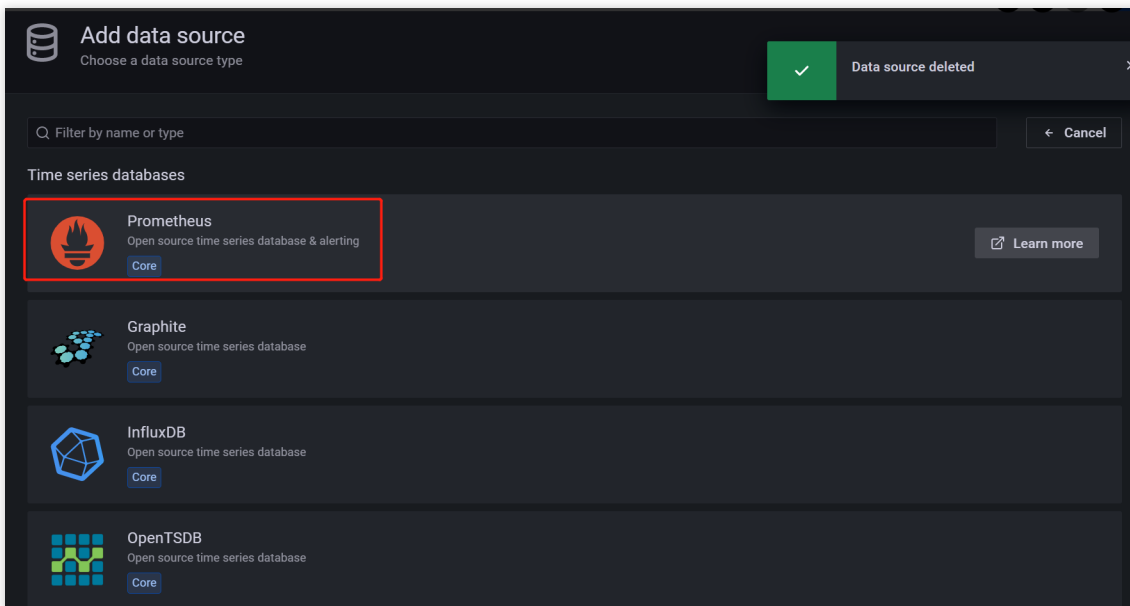


The screenshot shows a configuration form titled "Grafana Data Source Configuration Information". It contains three fields: "HTTP URL" with a value starting with "http://" and ending with ".io", "Basic auth user(APPID)" with a value "12..." followed by "62", and "Basic auth password" with a masked value "*****". Each field has a copy icon to its right.

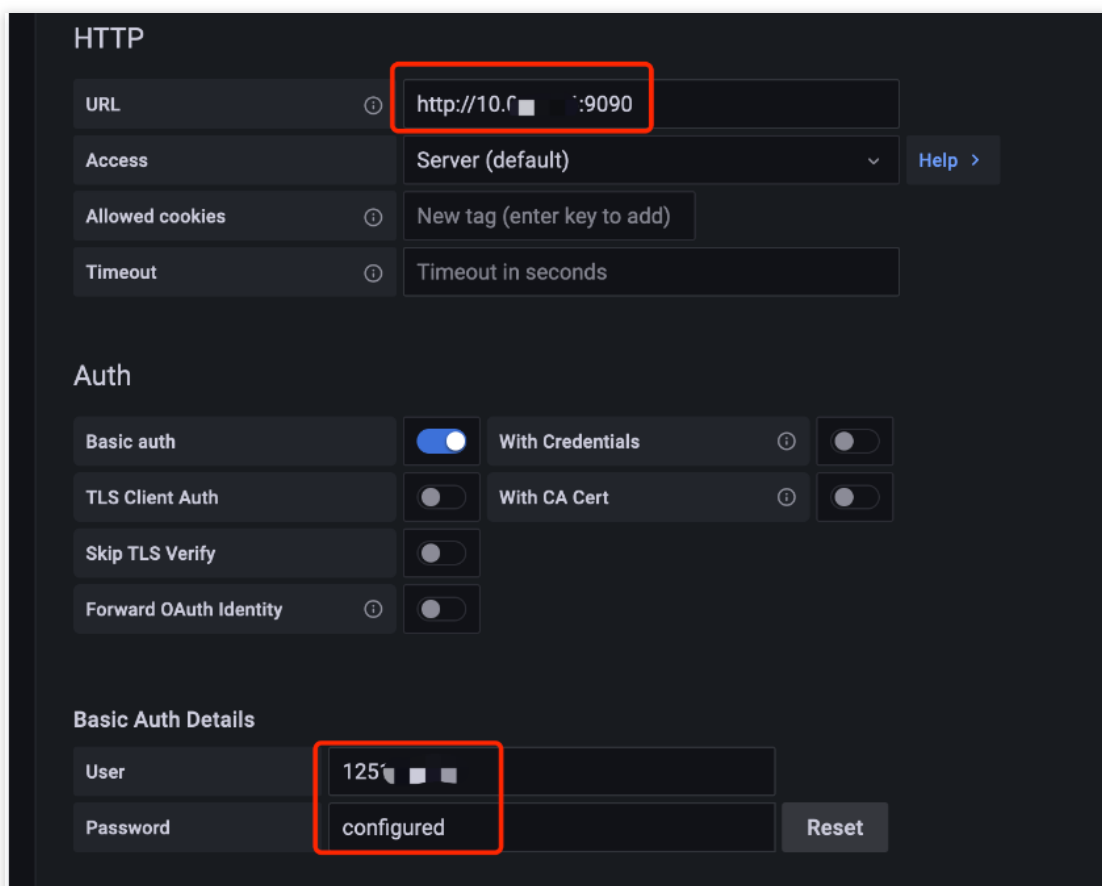
Grafana Data Source Configuration Information	
HTTP URL	http://...io
Basic auth user(APPID)	12...62
Basic auth password	*****

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**.



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.



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.

