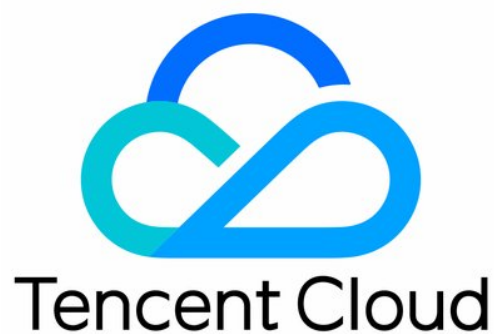


Tencent Cloud Observability Platform Best Practice Product Documentation



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

- Troubleshooting

- TKE Monitoring

- Configuring CVM Metrics and Creating Alarms

- Using Tag and TopN Features to Automatically Monitor Cloud Resources in Batches

- Using API to Pull Tencent Cloud Service Monitoring Data

- Dynamic Alarm Threshold

- Using API to Create Alarm Policy

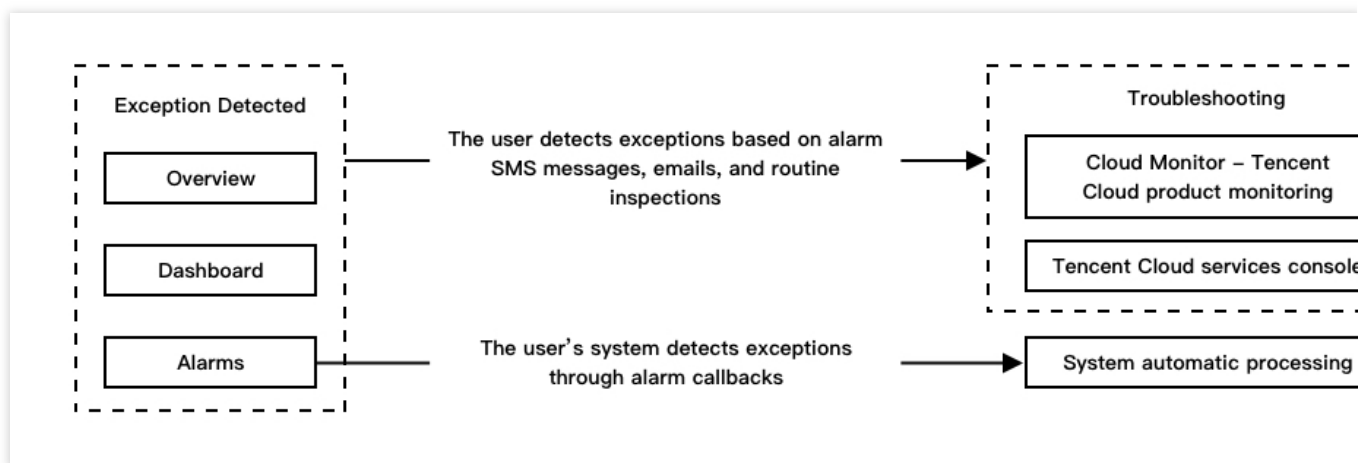
Best Practice

Troubleshooting

Last updated : 2024-01-27 17:45:42

Overview

Tencent Cloud Observability Platform provides multiple ways to help users detect resource exceptions and multiple channels to send the exception information to users as soon as possible.



Locating Exceptions

Detecting exceptions through alarms

Tencent Cloud uses monitoring and alarming to promptly detect exceptions and notify you automatically. This helps keep you informed on exceptions in real time across all scenarios. You can log in to the [Tencent Cloud Observability Platform console](#) and configure alarm policies for resources. For more information, please see [Creating Alarm Policies](#).

If you have configured key performance metrics and events as alarm rules, you will be notified promptly in multiple ways through the alarm channel if an exception occurs.

Alarm policies configured with an alarm recipient group will be sent to you through SMS, email, etc. Features such as repeated alarms and alarm aggregation are also supported to keep you informed while avoiding unnecessary notifications.

You can also configure the callback API feature in alarm channel to receive alarms promptly and process the alarm information.

Detecting exceptions through monitoring charts

You need to actively analyze the historical data and average trends of performance metrics to locate exceptions through monitoring charts. If an exception is difficult to locate by using alarm rules or has not been configured with alarms, you can use monitoring charts to locate it during daily health check. Compared to alarms, monitoring charts allow you to query the global impact of resource exceptions. You can subscribe key resources to the dashboard and configure monitoring charts to highlight exceptions in different scenarios.

For some instances, you can subscribe to details views to compare the trends of instance performance data on the dashboard.

For resource clusters, you can subscribe to the aggregated data of a cluster to see the overall monitoring chart of the cluster on the dashboard and compare it with the chart of a single instance in this cluster.

For exceptions detected through monitoring charts, you can use the sorting feature to locate specific resources related to an exception for further troubleshooting.

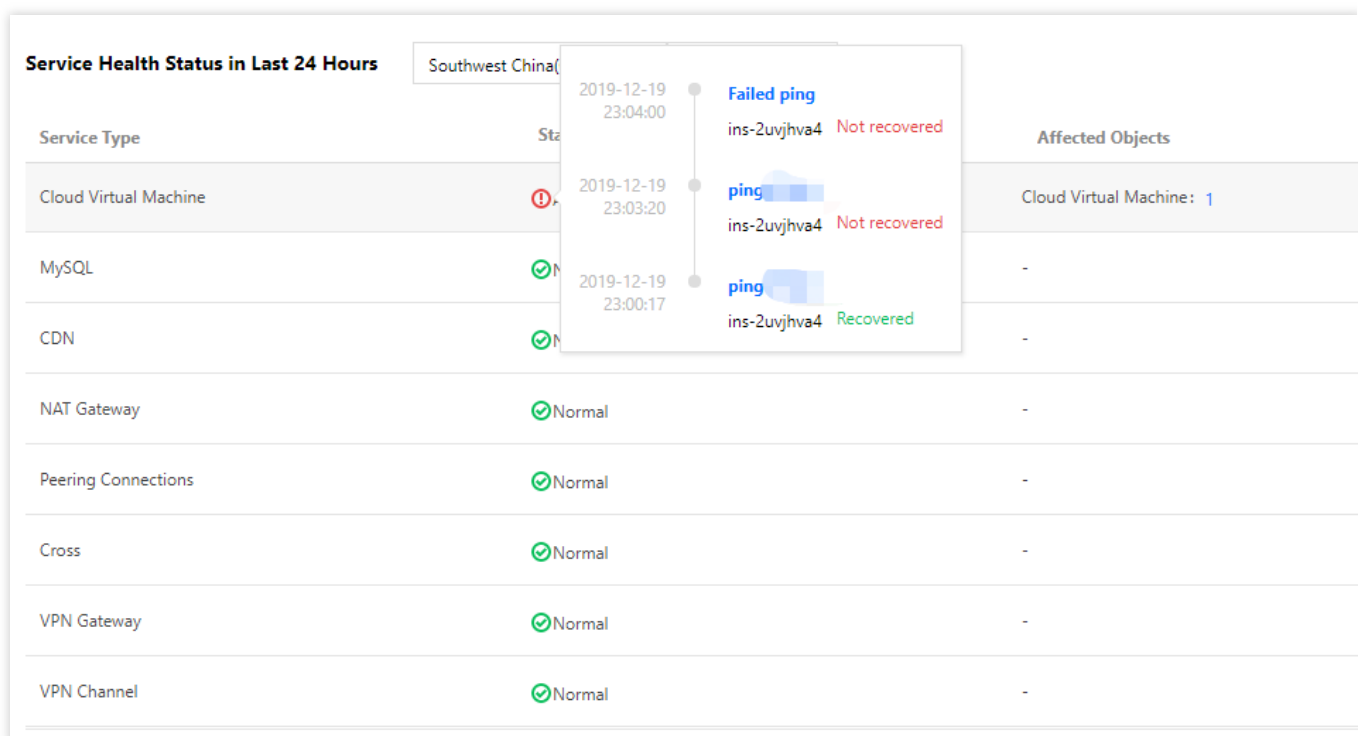
Troubleshooting

Locating exception objects on monitoring overview page

If you receive an alarm during daily health check, you can go to [Monitoring Overview](#) in the Tencent Cloud Observability Platform console.

1. Go to the overview page > service health status module to view exceptions in each region and project.

You can browse recent exceptions by clicking on the status of each service.



2. Click the number of exception objects to access the service monitoring page.

Service Health Status in Last 24 Hours

Southwest China(Chong...

▼

Default Project

▼

Service Type	Status ⓘ	Affected Objects
Cloud Virtual Machine	⚠ Abnormal	Cloud Virtual Machine: 1

Affected resource objects are automatically filtered out on the service monitoring page.

3. Click the ID of a specific object to go to the monitoring details page, where detailed information about its historical exceptions is provided.

The exception timeline allows you to view the current and historical information of the affected object. This helps you troubleshoot current exceptions based on historical alarms and status changes.

The monitoring data for resource performance allows you to compare the current and historical data of the same metric over a specified period of time or compare data changes of different metrics within the same period for troubleshooting.

Real Time

Last 24 hours

Last 7 days

Select Date

Data Comparison

Period: 10 second(s)

① Note: Max, Min, and Avg are the maximum, minimum, and average values of all points in the current line chart respectively.

CPU	CPUUtilization%	No data yet. Please check the status of the monitoring component. Click to fix	Max: -	Min: -	Avg: -
	CPUAvgLoad	No data yet. Please check the status of the monitoring component. Click to fix	Max: -	Min: -	Avg: -
	Basic CPU Utilization%	No data	Max: -	Min: -	Avg: -
MEM	MemoryUsageMB	No data yet. Please check the status of the monitoring component. Click to fix	Max: -	Min: -	Avg: -
	MemoryUtilization%	No data yet. Please check the status of the monitoring component. Click to fix	Max: -	Min: -	Avg: -

Locating exception objects through dashboards

Log in to the [Tencent Cloud Observability Platform console](#). On the left sidebar, click **Dashboard** to access the dashboard management page.

1. When you find an exceptional trend in the monitoring chart, click the time period when the exception occurs. A sorting list of corresponding instances is displayed below the chart. You can locate the specific exception objects based on the sorting list.

2. Click the name of an object in the sorting list to access its monitoring details page, where detailed information about its historical exceptions is provided.

The exception timeline allows you to view the current and historical information of the affected object. This helps you troubleshoot current exceptions based on historical alarms and status changes.

The monitoring data for resource performance allows you to compare the current and historical data of the same metric over a specified period of time or compare data changes of different metrics within the

TKE Monitoring

Last updated : 2024-01-27 17:45:42

New TKE Monitoring Features

- Monitored objects can be updated automatically.
- Workload/Component/Node monitoring scenarios are added.
- More monitoring metrics are added. The total number of new TKE metrics reaches 140.
- You can block a special object (such as a frequently alarming pod) in a specific monitoring dimension.

Directions

This document describes how to [automatically update a monitored object in the dashboard](#), [automatically update an alarm object](#), and [block a frequently triggered alarm object](#) by taking the "TKE monitoring - pod" dimension as an example.

Automatically updating monitored object in dashboard

1. Log in to the [Tencent Cloud Observability Platform console](#).
2. Select **Dashboard > Dashboard List > Create Dashboard > Create Chart**.
3. Configure the monitoring chart as detailed below:

Monitoring Type: select **Cloud Product Monitoring** here.

Metric: select "TKE (New) - pod" as the service and "CPU utilization (%)" as the metric here.

Filter: you can filter the objects to be bound to the chart by dimension (such as region, cluster, namespace, and workload).

Region: select the region of the monitored object.

Cluster: select the cluster of the monitored object.

Filter: you need to configure two filters, namely, the namespace and workload balance type, to monitor all pods under the specified workload and to automatically update the monitored object in the dashboard when pods are created/updated frequently, as shown below:

Cloud Product Monitoring

Metric ^① docker(new)-pod Others / CPU Usage(Core)

Filter ^①

Region Beijing

tke_cluster_instance_id cls-2

pod_name I7-lb-1 +

namespace kube-system +

workload_name I7-lt +

VS

☐ Day on day (compare to the same period yesterday) ☐ Week on week (compare to the same period last week) ☐ Select time periods and compare

4. After completing the configuration, click **Save** in the top-right corner of the page to save the chart.

Automatically updating alarm object

1. Log in to the [Tencent Cloud Observability Platform console](#).
2. Select **Alarm Configuration > Alarm Policy > Create** to enter the alarm policy creation page.
3. Select "TKE (New) - pod" as the policy type and configure the alarm object as detailed below:

Region: select the region of the monitored object.

Cluster: select the cluster of the monitored object.

Filter: you need to configure two filters, namely, the namespace and workload balance type, to monitor all pods under the specified workload and to automatically update the alarm object when pods are created/updated frequently, as shown below:

Basic Info

Policy Name example

Remarks Up to 100 characters. Only Chinese and English characters, numbers, underscores, and hyphens are allowed.

Monitor Type ☒ Cloud Product Monitoring ☐ Custom Cloud Monitor

Policy Type docker(new)-pod 0 exist. You can create 300 more static threshold policies. The current account has dynamic threshold policies, and 20 more can

Configure Alarm Rule

Filters (AND) ^①

Region Beijing

tke_cluster_instance_id cls-a

namespace kube-system +

workload_kind Deployment +

Note:

For more information on how to configure an alarm, please see [Creating Alarm Policy](#).

Blocking frequently triggered alarm object

When a pod frequently triggers an alarm, you can block some or all alarm objects under the node as instructed below.

As shown below, you can block some pod alarms by configuring the "!=" operator for the pod name.

Configure Alarm Rule

Filters (AND) ⓘ

Region

Beijing

tke_cluster_instance_id

cls-

namespace

=

kube-system

+

pod_name

!=

2 selec

+

🗑️

Configuring CVM Metrics and Creating Alarms

Last updated : 2024-01-27 17:45:42

Overview

This example shows you how to configure an alarm. Assume you want to send an alarm via SMS to the number

12345678888 when the CPU utilization of CVM instance ins-12345678 (in Guangzhou region) exceeds 80% for two consecutive 5-minute periods.

Directions

1. Log in to the [Tencent Cloud Observability Platform Console](#).
2. In the left sidebar, click **Alarm Configuration** -> **Alarm Policy**.
3. Click **Add** and configure the following items.
4. Configure the policy name and other items.

Policy Name: CPU alarm

Policy Type: Cloud Virtual Machine

5. Configure the alarm object. In the “Alarm Object” module, choose “Select some objects” and select the CVM instance.

Alarm Object

☐ All Objects

☒ Select some objects(0 selected)

☐ Select instance group [Create instance group](#)

Region: Mumbai

ID/Name	Network Type	IP
---------	--------------	----

6. Configure the trigger condition. In the “Trigger Conditions” module, configure the following conditions.

Select “Configure trigger conditions”

Select “Indicator alarm”: CPU Utilization -> 80% 5 minutes 2 periods

Alarm repetition period: 15 minutes

The screenshot shows the 'Trigger Condition' configuration window. At the top, there is a tab labeled 'Trigger Condition' with a sub-tab 'Trigger Condition Template' and a link 'Add Trigger Condition Template'. Below this, the 'Configure trigger conditions' option is selected with a radio button. Under 'Configure trigger conditions', the 'Indicator alarm' checkbox is checked. Below it, the 'Meet' dropdown is set to 'Any', followed by the text 'conditions, the alarm will be triggered'. The 'if' section shows a dropdown for 'CPUUtilization', a 'Measurement Pe' dropdown, a comparison operator '>', a value '0', a unit '%', a 'Continuous2' dropdown, and a 'then' clause followed by 'Alarm oc'. There are 'Add' links below the 'if' and 'then' clauses. The 'Event Alarm' checkbox is also checked, with a dropdown menu showing 'DiskReadonly' and an 'Add' link below it.

7. Configure the alarm channel. Add an alarm recipient group (click **Add Recipient Group** to create one if you have not already done so).

The screenshot shows the 'Alarm Channel' configuration window. On the left, there is a tab labeled 'Alarm Channel'. The main area is titled 'Recipient Object' and contains a 'Recipient Group' dropdown, a search bar, and a link 'Add Recipient Group'. Below this is a table with two columns: 'User Group Name' and 'User Name'. The table has two rows: the first row has a checked checkbox, 'User Group Name', and 'User Name'; the second row has a checked checkbox, 'cm', and a greyed-out 'User Name' field. Below the table, there is a 'Valid Period' section with two time pickers: '00:00:00' and '23:59:59', separated by 'to'. At the bottom, the 'Receiving Channel' section has two checked checkboxes: 'Email' and 'SMS'.

8. Click **Complete** to complete the alarm configuration.

9. If the CPU utilization of the instance exceeds 80% for two consecutive 5-minute periods, the number `12345678888` will receive an alarm via SMS from Tencent Cloud.

Using Tag and TopN Features to Automatically Monitor Cloud Resources in Batches

Last updated : 2024-01-27 17:45:42

Feature Overview

Tencent Cloud Tag: tag is a resource management tool provided by Tencent Cloud. You can use tags to categorize, search for, and aggregate Tencent Cloud resources. A tag has two parts: tag key and tag value. You can create a tag by defining its tag key and tag value based on conditions such as the resource usage and resource owner.

Use tags in the dashboard: the dashboard allows you to bind tags to the data sources of a chart, and the monitoring curves will be updated automatically as the number of the associated instances changes. This enables you to bind tags to instances in batches quickly and dynamically, thus greatly reducing the costs of chart creation and modification.

TopN feature: it updates the monitoring curves automatically when you increase or decrease the number of instances and migrate instances, thus automatically monitoring the loads of machines.

Use Limits

The dashboard tag feature currently is only supported for CVM - basic monitoring and will be supported for more Tencent Cloud services in the future.

Each resource can be associated with up to 50 different tag keys.

Each user can create up to 1,000 tag keys.

Each tag key can be associated with up to 1,000 tag values.

Purpose

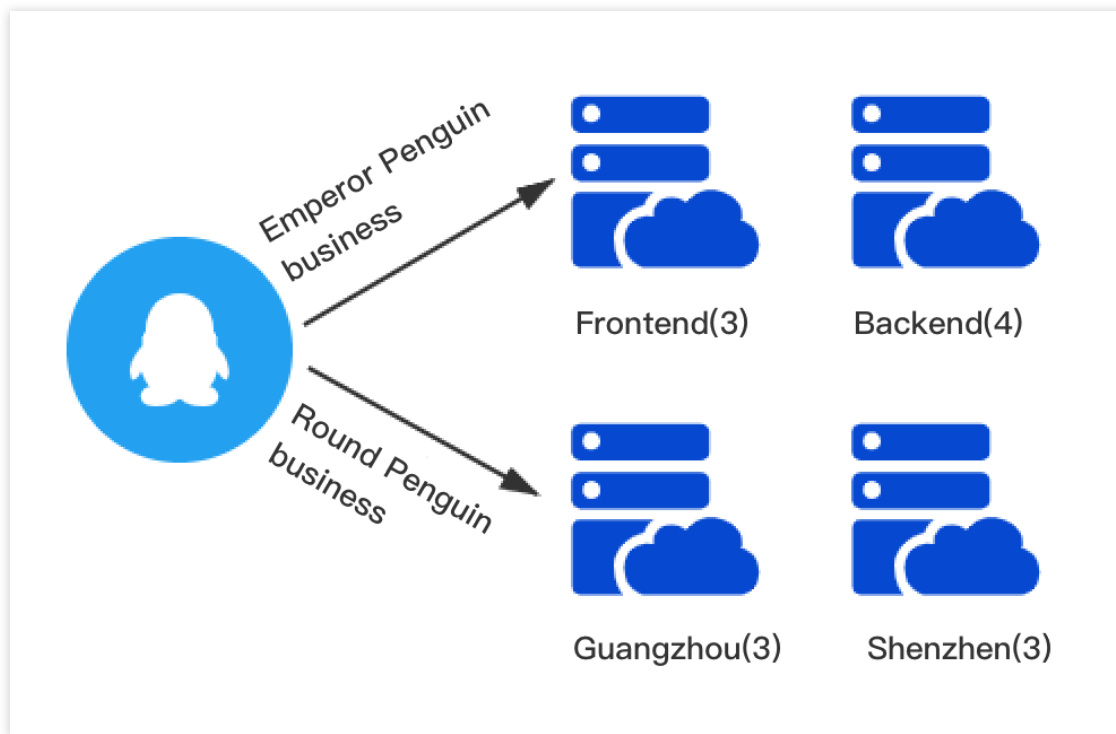
This document describes how to use the tag, chart grouping, and TopN features of the dashboard to achieve the automated OPS of resources and the automated monitoring of the loads of machines in batches by taking "CVM – basic monitoring – CPU utilization" as an example.

Background

As shown below, two businesses run under the Penguin project: the Emperor Penguin business and the Round Penguin business.

The Emperor Penguin business involves 7 servers, with 3 on the frontend and 4 on the backend.

The Round Penguin business involves 6 servers, with 3 in Guangzhou and 3 in Shenzhen.



Directions

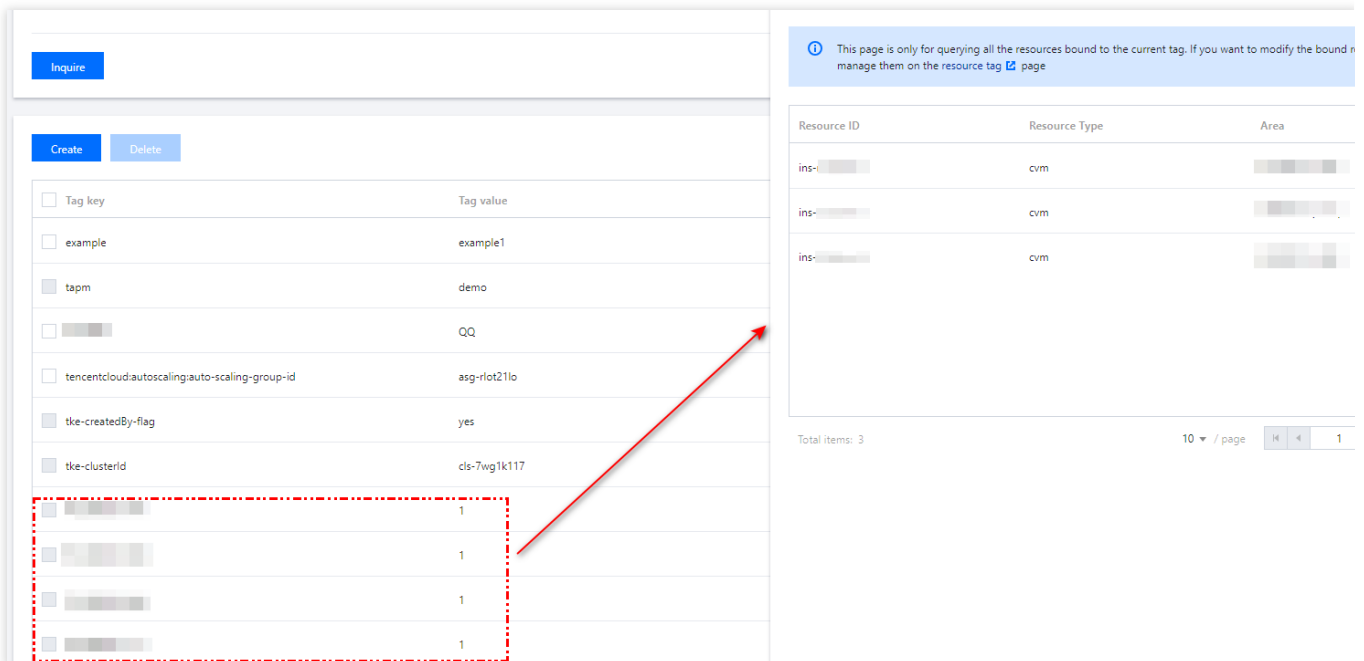
Step 1. Create a tag

1. Go to the tag list page in the [Tag console](#).
2. On the tag list page, click **Create** and enter the tag key and tag value (which can be left empty).
3. After entering the information, click **OK**.
4. Create four tags ("Emperor Penguin - Frontend", "Emperor Penguin - Backend", "Round Penguin - Guangzhou", and "Round Penguin - Shenzhen") as instructed in steps 2 and 3.

Step 2. Associate instances

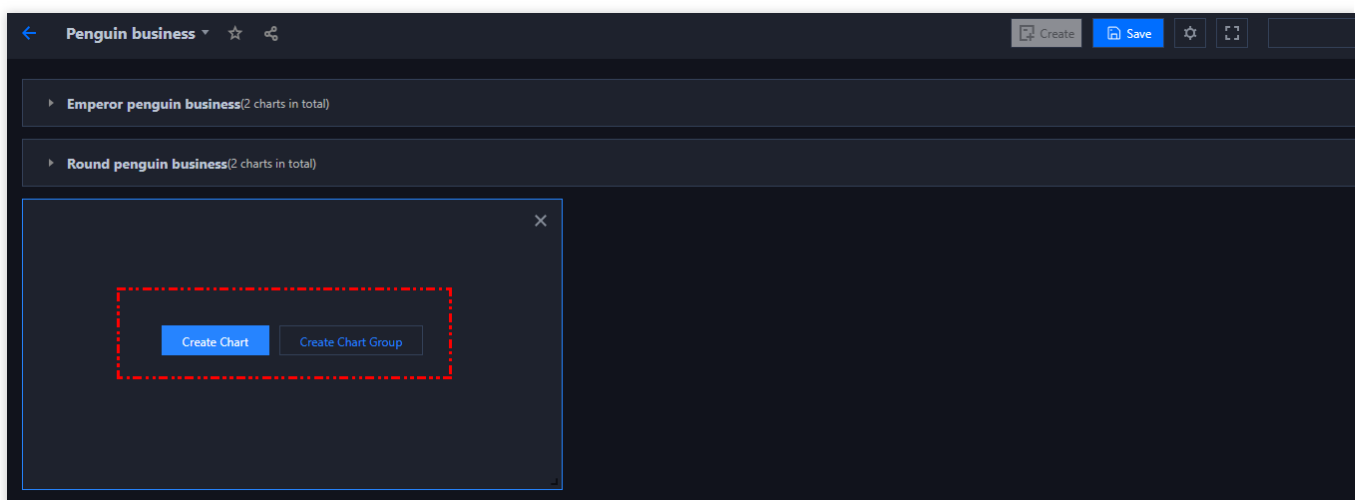
1. Go to the instance list page in the [CVM console](#).
2. On the instance list page, find the instances of the Emperor Penguin and Round Penguin businesses and select **More > Instance Settings > Edit Tag** in the **Operation** column.
3. In the tag editing window, associate the instance with the corresponding tag key and value and click **OK**.

4. Associate all instances of the Emperor Penguin and Round Penguin businesses with the corresponding tags as instructed in steps 2 and 3 as shown below:



Step 3. Create a dashboard and chart group

1. Create a dashboard named `Penguin Project` as instructed in [Creating a Dashboard](#).
2. Create a chart group. As shown below, click the creation icon in the top-right corner of the dashboard, then click **Create Chart Group** and the settings icon next to the chart group name. Enter the chart group name and click **OK**. In this document, create two chart groups for the two businesses, respectively.



Step 4. Create a monitoring chart and bind tags

1. Click **Create Chart** and configure the chart as follows:

Chart Name: you can enter the **Chart Name** in the **Basic Information** section of the **Chart Configuration**.

Monitoring Type: select **Cloud Product Monitoring** here.

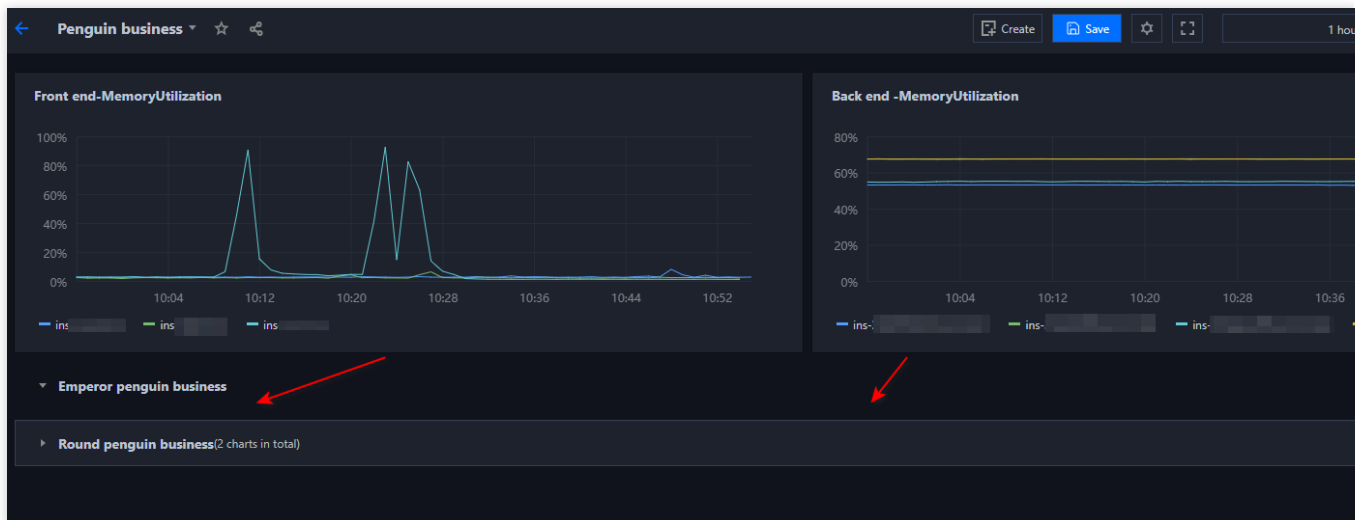
Metric: select the Tencent Cloud service type and the metric to be monitored. Here, **CVM - basic monitoring** and the **CPU utilization** metric are used as an example.

Filter: you can filter the data sources to be monitored. Here, **tag** is selected as the filter type, and different tag keys and values are bound to different charts.



2. After the chart is created successfully, drag it to the desired chart group and adjust its size as needed.

3. You can create monitoring charts for the 3 frontend CVM instances and 4 backend CVM instances of the Emperor Penguin business and for the 3 CVM instances of the Round Penguin business in Guangzhou and 3 CVM instances of the Round Penguin business in Shenzhen as instructed in steps 2 and 3. Then, drag and drop them to the corresponding group as shown below:

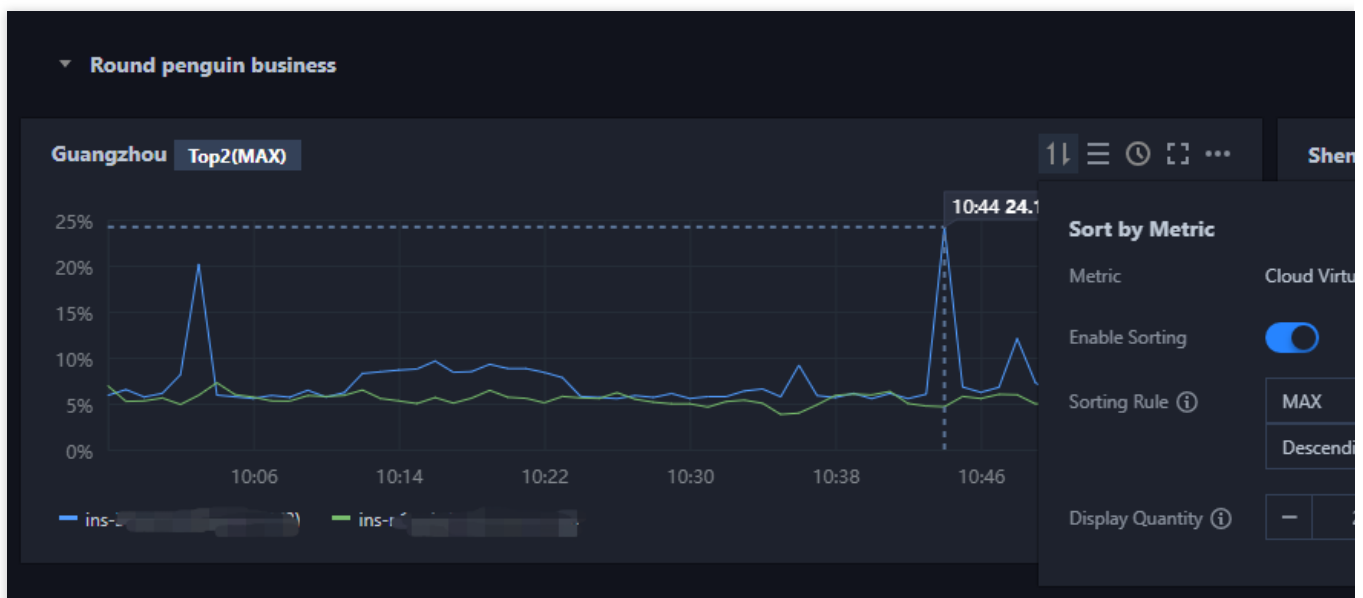


Step 5. Enable the TopN feature

When analyzing the charts, you can click



in the charts to enable the TopN feature and change the sorting rule and display quantity, making it easier for you to view the loads of machines in batches. The chart below only shows the top 2 backend instances of the Emperor Penguin business:



Using API to Pull Tencent Cloud Service Monitoring Data

Last updated : 2024-01-27 17:45:42

This document describes how to use APIs to pull the monitoring data of Tencent Cloud services.

API Overview

Tencent Cloud Observability Platform provides the following two types of APIs for querying metric monitoring data

API	Operation	Description
DescribeBaseMetrics	Queries the details of basic monitoring metric	This API is used to query the types of basic monitoring metrics under the corresponding namespace
GetMonitorData	Pulls metric monitoring data	This API is used to get the corresponding monitoring data of a metric in the object dimension

API limits

The `GetMonitorData` API supports getting the monitoring data of a certain metric for all instances under an account in batches.

The `GetMonitorData` API can be called 20 times per second (1,200 times/minute) by default. A single request can get the data of up to 10 instances and up to 1,440 data points.

The retention period of monitoring data varies by monitoring granularity as detailed below:

Monitoring Granularity	Retention Period
1 second	1 day
1 minute	15 days
5 minutes	31 days
1 hour	93 days
1 day	186 days

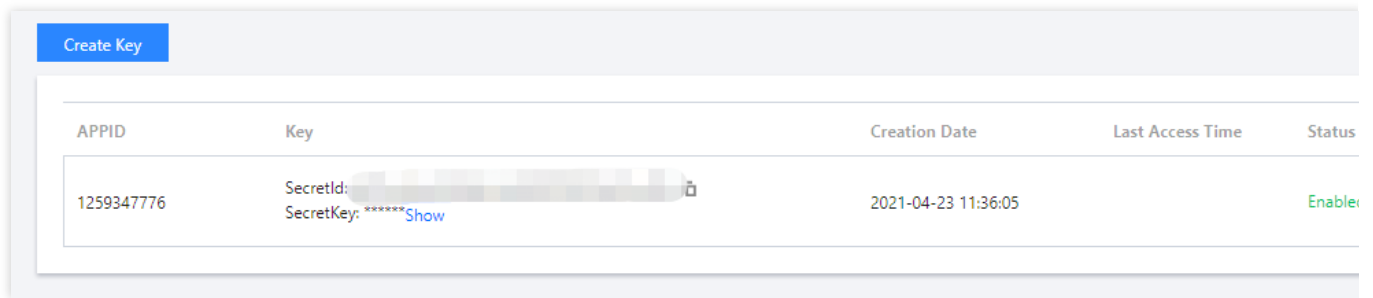
Note:

The monitoring data of 1-minute metrics related to CPU, memory, and network for CVM is retained for 31 days.

Preparations

Preparing personal key

1. Go to [Manage API Key](#).
2. If no key has been created, you need to click **Create Key** to create a key. If you have already created a key, you can click **Display** after `SecretKey` to get the key.



Create Key				
APPID	Key	Creation Date	Last Access Time	Status
1259347776	SecretId: [masked] SecretKey: ***** Show	2021-04-23 11:36:05		Enabled

Preparing Tencent Cloud service metric information

This document takes the CVM CPU utilization metric as an example.

1. Go to [CVM Monitoring Metrics](#).
2. Find the CPU utilization metric to view the CPU utilization metric name, dimension, statistical period, and other related information.

CVM Monitoring Metrics

[Download PDF](#)

Last updated: 2020-07-14 16:26:33

Namespace

Namespace=QCE/CVM

Monitoring Metrics

Basic metrics

Parameter	Metric Name	Calculation Method	Description	Unit	Dim
CPUUsage	CPU utilization	Percentage of the "user+nice+system+irq+softirq+idle+iowait" time of the CPU to the total time	Percentage of CPU utilization in real time when the CVM is running	%	Inst
CPULoadAvg	Average CPU load	Analyze data in <code>/proc/loadavg</code> and collect the average load of the system in the past one minute at 10 second intervals (this metric is not available for Windows CVMs)	Average number of tasks that are using and are about to use the CPU in a period of time	-	Inst

Directions

This demo describes how to use the [GetMonitorData](#) API to query the CPU utilization of a CVM instance.

1. Log in to [API Explorer](#).
2. Copy the [prepared personal key](#) into the corresponding **SecretId** and **SecretKey** text boxes.
3. Find **Region** in the **Input Parameters** section and select the relevant region.
4. Enter the [prepared Tencent Cloud service information](#) in the corresponding text boxes in the **Input Parameters** section.

Namespace: enter QCE/CVM.

MetricName: enter the name of the CPU utilization metric, i.e., **CPUUsage**.

Dimensions.N-Name: enter the supported dimension name, i.e., **Instanceld**.

Dimensions.N-Value: enter the corresponding **Instanceld** value (CVM instance ID), such as **ins—12345678**, which can be obtained through the [DescribeInstances](#) API of CVM.

Period: enter a statistical period supported by the metric, such as 300.

StartTime: enter the query start time in the format of `2020-12-20T19:51:23+08:00` (in the `datetime_iso` type).

EndTime: enter the query end time in the format of `2020-12-20T20:51:23+08:00` (in the `datetime_iso` type). **EndTime** cannot be earlier than **StartTime**.

GetMonitorData2018-07-24

Key for debugging:

More Options ▾

Input Parameters

☐ View Only Required Parameters

Region (See the document for more regions and billing)

ap-beijing

Namespace

QCE/CVM

MetricName

CPUUsage

Instances.N

1

Dimensions.N

1

Name

InstanceId

Value

ins-123456

Add

Add

Period

300

StartTime

2020-12-20T19:51:23+08:00

EndTime

2020-12-20T20:51:23+08:00

Code Generating

Online Call

Signature Generation

Parameter Description

Feedback

```
tool monitor GetMonitorData --cli-unfold-argument --region ap-beijing --Namespace QCE/CVM --MetricName CPUUsage --Period 300 --StartTime 51:23+08:00 --EndTime 2020-12-20T20:51:23+08:00 --Instances 0 --Dimensions 0 --Name InstanceId --Instances 0 --Dimensions 0 --Value ins-123456
```

☒ Java ☐ Python ☐ Nodejs ☐ PHP ☐ GO ☐ .NET ☐ C++

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;

import com.tencentcloudapi.monitor.v20180724.MonitorClient;
import com.tencentcloudapi.monitor.v20180724.models.*;

public class GetMonitorData
{
    public static void main(String [] args) {
        try{

            Credential cred = new Credential("SecretId", "SecretKey");

            HttpProfile httpProfile = new HttpProfile();
            httpProfile.setEndpoint("monitor.tencentcloudapi.com");

            ClientProfile clientProfile = new ClientProfile();
            clientProfile.setHttpProfile(httpProfile);

            MonitorClient client = new MonitorClient(cred, "ap-beijing", clientProfile);

            GetMonitorDataRequest req = new GetMonitorDataRequest();
            req.setNamespace("QCE/CVM");
            req.setMetricName("CPUUsage");
            req.setPeriod(300L);
            req.setStartTime("2020-12-20T19:51:23+08:00");
            req.setEndTime("2020-12-20T20:51:23+08:00");

            Instance[] instances1 = new Instance[1];
            Instance instance1 = new Instance();

            Dimension[] dimensions1 = new Dimension[1];
            Dimension dimension1 = new Dimension();
            dimension1.setName("InstanceId");
            dimension1.setValue("ins-123456");
            dimensions1[0] = dimension1;

            instance1.setDimensions(dimensions1);

            InstanceData reqData = new InstanceData();
            reqData.setInstances1(instance1);
```

5. After completing the above information, you can copy the code in the corresponding language on the **Code Generation** tab to integrate the relevant monitoring data into your self-built monitoring system. You can also use **Online Call** to send a request to query the monitoring data online.

Dynamic Alarm Threshold

Last updated : 2024-01-27 17:45:42

The dynamic threshold feature of Tencent Cloud Observability Platform can intelligently detect metric exceptions and send alarms without requiring you to set metric thresholds. This document describes this feature and its benefits and use cases in detail.

Background

As a national key project, the Seventh National Population Census of the People's Republic of China ("2020 Chinese Census") involved the use of multiple Tencent Cloud services, such as [CVM](#), [CLS](#), [TencentDB for MySQL](#), and [CDN](#). In addition, to ensure the stability of the monitored Tencent Cloud services, the project owner not only used Tencent Cloud service metrics but also reported many custom business metrics, such as service request time, error statistics, and online users. In this scenario characterized by abundant metrics, a huge user base, and considerable fluctuations in access requests during day and night, it was difficult to guarantee the accuracy and availability of monitoring results and alarms by using static thresholds to monitor metrics. This document describes how the dynamic threshold feature can be used in different scenarios by taking the 2020 Chinese Census project as an example.

Analysis

From the perspective of OPS personnel, the following key metrics need to be monitored:

Tencent Cloud service metrics: CPU utilization, memory utilization, traffic, bandwidth, and API success rate.

Custom metrics: request time, error statistics, and online users.

In order to detect any exceptions in the key metrics listed above in a timely manner, it is necessary to always monitor them. The traditional solution is to use static threshold alarms, where OPS personnel configure certain metric thresholds based on their experience. However, the following pain points arise during the configuration of metric thresholds:

Pain point 1. How to configure reasonable thresholds?

For each category of metrics, OPS personnel need to configure thresholds they think reasonable based on specific businesses, and the thresholds configured by different OPS personnel may vary.

Example: the static thresholds configured by experienced John Smith and less experienced Jane Smith for the same metric are different. For example, for the CPU utilization metric, John knows that an alarm needs to be triggered only when the CPU utilization of a certain machine reaches 85%, so the threshold he configures will meet the business needs. However, due to a lack of experience, Jane configures the alarm to be triggered when the CPU utilization

exceeds 50%, in which case, a lot of unreasonable alarm notifications may be generated and thus cause disturbance. Therefore, the threshold policies configured by different OPS personnel may differ significantly.

Pain point 2. How to ensure the consistent reasonableness of thresholds?

A reasonable threshold previously configured for a metric may become unreasonable as your business changes, so you need to check whether it is reasonable every day. If you don't adjust it, alarms may not be triggered correctly as expected.

Example: suppose the traffic of a certain business is low after its launch due to a small user base and fluctuates around 100 MB, so John Smith configures alarms to be triggered when the traffic exceeds 120 MB based on the value range of the metric. However, as the business grows, the value of the traffic metric gradually increases and fluctuates around 150 MB. At this point, the threshold of 120 MB previously configured becomes unreasonable, and John will need to manually adjust the threshold to 170 MB based on the traffic fluctuation.

Pain point 3. How to reflect the direction of metrics?

For metrics whose upward and downward changes require attention, it is necessary to configure several thresholds to ensure the accuracy of alarms.

Example: the directions of changes that require attention vary by metric. For example:

For metrics whose expected value is 100%, such as API success rate, only a decline in the metric value is considered exceptional and therefore requires attention.

For metrics of which a low value is expected, such as failure rate, error statistics, and request time, only an increase in the metric value is considered exceptional and therefore requires attention.

For metrics whose value has no obvious directions, such as traffic and online users, both an increase and a decrease are considered exceptional.

The dynamic threshold feature can adaptively extract the characteristics of the metric curve such as the trend, period, and fluctuation based on the historical trend and then automatically calculate a reasonable upper threshold. It greatly simplifies the configuration and maintenance of reasonable thresholds.

Use Cases

The following lists the use cases where dynamic thresholds are suitable as well as their characteristics, detailed metrics, and recommended configurations:

Use Case	Metrics	Characteristics	Recommended Configurations
Percentage	Success rate, failure rate, packet loss rate, traffic hit rate, outbound traffic utilization, query	Such metrics range between 0 and 100%. Users will only concern if such metrics reach certain levels. For example, users will only care when the disk	If there is a definite threshold limit, we recommend you use dynamic thresholds together with static thresholds, with the sensitivity set to low-to-medium

	rejection rate, and bandwidth utilization	utilization exceeds 95%. It is suitable to use static thresholds or both static and dynamic ones for such metrics.	and the duration set to 2–4 consecutive statistical periods.
Network traffic	Network inbound bandwidth, network outbound bandwidth, network inbound packets, and network outbound packets	The values of such metrics fluctuate greatly in an uncertain range over time. It is suitable to use dynamic thresholds for such metrics.	We recommend you set the sensitivity to medium-to-high and the duration to 2–4 consecutive statistical periods.
Delay	Delays, delay distance, and delay time	The values of such metrics typically fluctuate slightly in an uncertain range. It is suitable to use dynamic thresholds for such metrics.	We recommend you set the sensitivity to low-to-medium and the duration to 3–10 consecutive statistical periods.
Others	Slow queries, TencentDB threads, Redis connections, TCP connections, disk QPS, IO wait time, temp tables, full-table scans, and unconsumed CKafka messages	It is suitable to use dynamic thresholds for such metrics.	We recommend you set the sensitivity to medium-to-high and the duration to 1–5 consecutive statistical periods.

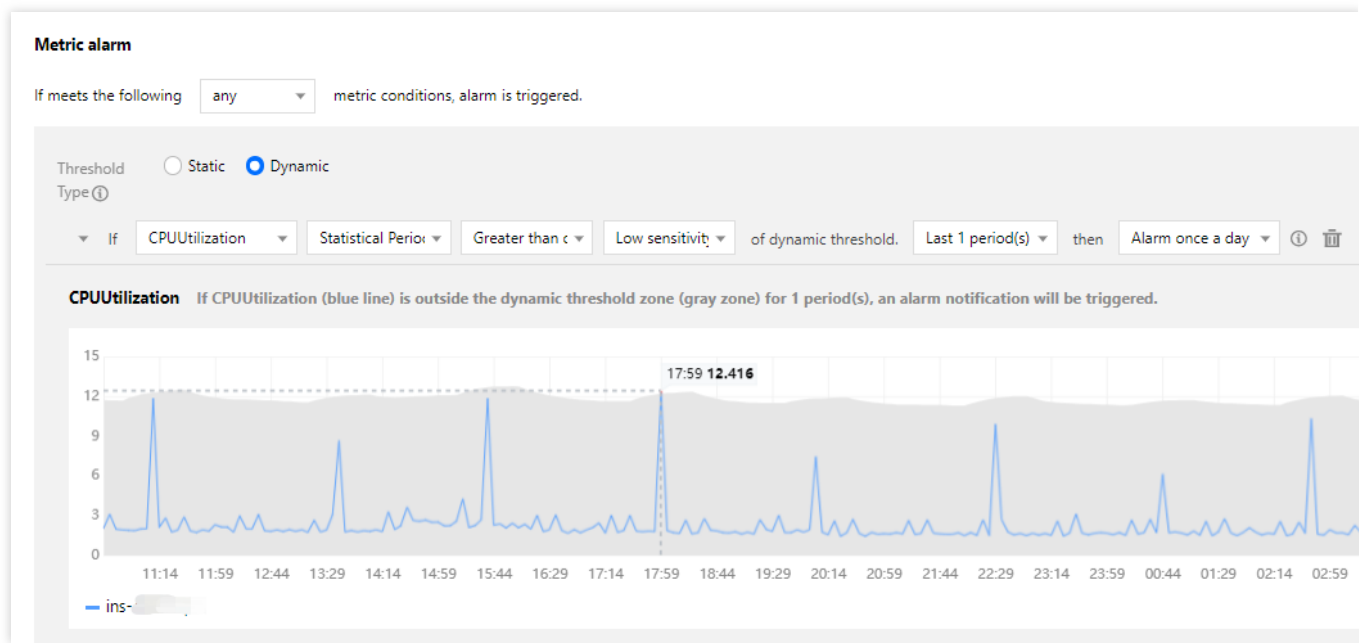
The following describes how to configure a dynamic threshold for a metric in each use case.

Percentage metrics

For percentage metrics, the value range is fixed, typically 0–100%. Users will only concern if such metrics reach certain levels. For example, users will only care when the disk utilization exceeds 95%. It is suitable to use static thresholds or both static and dynamic ones for such metrics. However, we recommend you consider using a medium sensitivity in practical applications.

Scenario 1: if you definitely know when a serious problem might occur with regard to a metric, such as CPU utilization, where alarms are configured to be triggered typically when the threshold of 90% is reached, you can consider using a static threshold.

Scenario 2: if you feel that an alarm threshold of 90% for a percentage metric cannot help you detect some problems in advance, you can consider using a dynamic threshold as shown below, where alarms will be triggered when the metric value surges, thus allowing you to solve problems as soon as they arise.



Scenario 3: if you feel that alarms are helpful only if they are triggered when the threshold of 60% is reached in case of metric value surge, you can consider using a dynamic threshold in combination with a static threshold.

Network traffic metrics

The values of such metrics fluctuate greatly in an uncertain range over time. It is suitable to use dynamic thresholds for such metrics.

Scenario 1: if you, as an OPS specialist, need to observe data generated a very long time ago to determine a reasonable threshold for a metric, or if you still cannot determine the optimum threshold after observation, a dynamic threshold can save you worry.

Scenario 2: if you clearly know that an exception occurs when the value of the traffic metric exceeds or drops below a certain threshold, you can consider using a dynamic threshold in combination with a static threshold.

Delay metrics

The values of such metrics typically fluctuate slightly in an uncertain range. It is suitable to use dynamic thresholds for such metrics.

As there are a lot of glitches in delay metrics, we recommend you use a medium sensitivity and a longer duration to filter out glitches and improve the alarm quality.

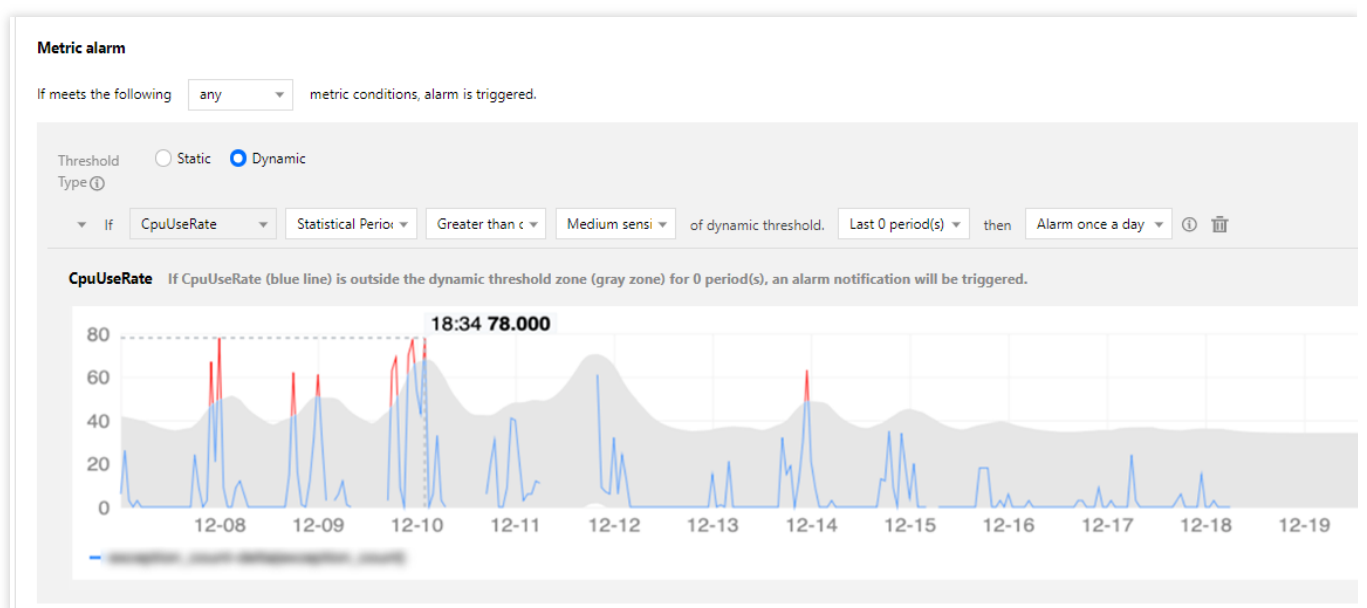


Other metrics

Exception statistics

At the early stage of your business, if it is difficult for you to configure a reasonable threshold, and you need to manually adjust it every day based on the changes, then you can use a dynamic threshold.

Dynamic thresholds can adaptively track the trends of the metrics, thus making it easier for you to determine reasonable thresholds.



As long as you select a "greater or less than" threshold, the system will adaptively identify sudden increases and decreases and send alarm notifications.

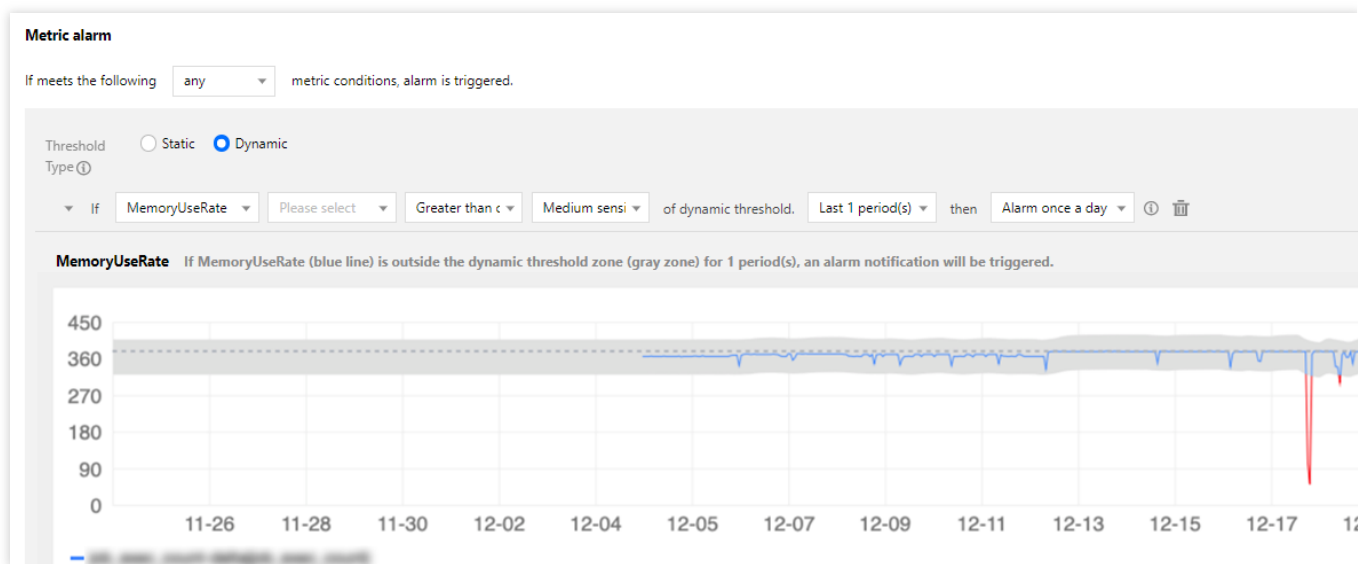
Statistics metrics

It can be found through observation that most metric values are around 350. An exception may have occurred when the metric value increases or decreases suddenly. If you use static thresholds, you would need to configure two reasonable upper and lower thresholds.

After a period of operation, the metric value may increase to 550, which is just in alignment with the current business conditions.

If you use static thresholds, you will keep receiving alarm notifications and need to reconfigure thresholds that are suitable at the current stage.

If you use a dynamic threshold, only when the metric value suddenly increases from 350 to 550, alarm notifications will be sent until the metric value stabilizes at 550 when the algorithm intelligently identifies the current value as a normal value.



Note:

For more information on how to configure dynamic thresholds, please see [How to Use Dynamic Thresholds](#).

Using API to Create Alarm Policy

Last updated : 2024-01-27 17:45:42

This document describes how to use [CreateAlarmPolicyAPI](#) and [Binding Policy ObjectAPI](#) to create alarm policies and bind alarm objects.

Preparations

Before calling the [Create Alarm PolicyAPI](#) , you need to prepare the following information.

Preparing personal key

1. Go to the [API Key Management](#) page in the CAM console.
2. Click **Show** to get the `SecretKey` .

Note:

If no key has been created, please click **Create Key** to create one.

Preparing alarm policy type

You can query all policy types through the [DescribeAllNamespacesAPI](#) in the following steps:

1. Log in to the [API Explorer console](#) and enter the input parameters as shown below:

Parameter	Description
SecretId, SecretKey	Enter the prepared `SecretId` and `SecretKey`
Region	Select the corresponding region
SceneType	Enter `ST_ALARM`
Module	Enter `monitor`
MonitorTypes.N	Optional

2. Click **Online Call** > **Send Request** to get the response, where `Response.QceNamespacesNew.N.Id` is the `Namespace` required by alarm policy creation.

Note:

Here, `Namespace` is the alarm policy type, which is different from the Tencent Cloud service namespace used to pull monitoring data.

Preparing metric list

You can query all alarm metrics under the policy type through the [DescribeAlarmMetricsAPI](#).

1. Log in to the [API Explorer console](#) and enter the input parameters as shown below:

Parameter	Description
SecretId, SecretKey	Enter the prepared `SecretId` and `SecretKey`
Region	Select the corresponding region
Module	Enter `monitor`
MonitorType	Enter `MT_QCE`
Namespace	Enter the alarm policy type obtained in the "Preparing alarm policy type" step, i.e., `Response.QceNamespacesNew.N.Id` in the returned result

2. Click **Online Call > Send Request** on the right to get the response, where `Response.Metrics.N` lists all the alarm metrics under the policy type.

Preparing event list

You can query all the alarm events under the policy type through the [DescribeAlarmEventsAPI](#).

1. Log in to the [API Explorer console](#) and enter the input parameters as shown below:

Parameter	Description
SecretId, SecretKey	Enter the prepared `SecretId` and `SecretKey`
Region	Select the corresponding region
Module	Enter `monitor`
Namespace	Enter the alarm policy type obtained in the "Preparing alarm policy type" step, i.e., `Response.QceNamespacesNew.N.Id` in the returned result

2. Click **Online Call > Send Request** to get the response, where `Response.Events.N.EventName` is the `EventName` required by alarm policy creation.

Directions

This document describes how to use APIs such as [CreateAlarmPolicyAPI](#) to create an alarm policy for CVM - basic monitoring with the following example.

Creating alarm policy

1. Log in to the [API Explorer console](#).
2. Copy the [prepared personal key](#) into the corresponding `SecretId` and `SecretKey` text boxes.
3. Find **Region** in the **Input Parameters** section and select the relevant region.
4. Enter `monitor` in **Module**, a custom policy name in **PolicyName**, and `MT_QCE` in **MonitorType**.
5. Enter the alarm policy type obtained in the [Preparing alarm policy type](#) step above in **Namespace**. For example, the alarm policy type of CVM - basic monitoring is `cvm_device`.
6. In the CVM - basic monitoring use case, `Remark` and `Enable` are optional, while `ProjectId` is required.

Remark: remark, which is optional.

Enable: whether to enable the alarm policy. 0 indicates to disable, 1 indicates to enable, and the default value is 1. This parameter is optional.

ProjectId: project ID, which should be 0 for CVM - basic monitoring.

Note:

`ProjectId` is the project ID. -1 indicates no project, 0 indicates the default project, and the default value is -1. This parameter is optional depending on the policy type. For example, some alarm policy types don't have the concept of project (such as VPC), and the default value `-1` can be used. If the alarm policy type has the concept of project (such as CVM - basic monitoring), an error will be reported if the default value `-1` is passed in, and the input parameter should be changed to "0".

7. Configure `Condition` as shown below:

Parameter	Required	Description
IsUnionRule	Yes	Metric trigger condition operator. Valid values: 0 (OR), 1 (AND). OR means that the alarm will be sent when any condition is triggered, while AND means that the alarm will be sent when all conditions are triggered
Rules.N	Yes	<p>List of alarm trigger conditions, which can be configured by referring to the <code>`AlarmPolicyRule`</code> parameter</p> <p>MetricName: enter the <code>`MetricName`</code> (Metrics.N.MetricName) returned in the Preparing metric list step</p> <p>Period: enter the <code>`Period`</code> (Metrics.N.MetricConfig.Period) returned in the Preparing metric list step</p> <p>Operator: enter the <code>`Operator`</code> (Metrics.N.MetricConfig.Operator) returned in the Preparing metric list step</p> <p>Value: enter a threshold without the unit, such as 80</p> <p>ContinuePeriod: enter the <code>`ContinuePeriod`</code> (Metrics.N.MetricConfig.ContinuePeriod) returned in the Preparing metric list step</p> <p>NoticeFrequency: set the alarm frequency in seconds. Parameter description: alarm interval in seconds. Valid values: 0 (do not repeat), 300 (alarm once every 5 minutes), 600 (alarm once every 10 minutes), 900 (alarm once every 15 minutes), 1800 (alarm once every 30 minutes), 3600 (alarm once every hour), 7200 (alarm</p>

once every 2 hours), 10800 (alarm once every 3 hours), 21600 (alarm once every 6 hours), 43200 (alarm once every 12 hours), 86400 (alarm once every day)
IsPowerNotice: set whether the alarm frequency grows exponentially. Valid values: 0 (no), 1 (yes)
 Other parameters can be left empty

8. If you want to trigger an event alarm, you need to configure the EventCondition parameter. Under EventCondition, you only need to enter the EventName obtained in the [Preparing event list](#) step in Rules.N.MetricName, and you can leave other parameters empty.
9. Enter the alarm notification template ID in Noticelds.N, such as notice-qvq836vc, which can be obtained through the [DescribeAlarmNoticesAPI](#).
10. After entering the above parameters, click **Online Call > Send Request**. The following figure shows the successful creation of the alarm policy for CVM - basic monitoring.
11. After the creation is successful, you can view the alarm policy on the [Alarm Policy](#) page in the Tencent Cloud Observability Platform console.

Binding alarm object

1. Log in to the [API Explorer console](#).
2. Copy the [prepared personal key](#) into the corresponding `SecretId` and `SecretKey` text boxes.
3. Find **Region** in the **Input Parameters** section and select the relevant region.
4. Enter `monitor` in **Module**.
5. Enter 0 in **GroupId**.
6. Enter either the `InstanceGroupId` or `Dimensions` as detailed below:

InstanceGroupId: instance group ID. If you want to bind alarm objects by instance group, you need to pass in the instance group ID (such as 1234), which can be found on the [instance group](#) page in the Tencent Cloud Observability Platform console by clicking the corresponding instance name as shown below:

Dimensions.N: if you want to bind an alarm policy by instance ID, you need to enter `Dimensions` as detailed below:

Parameter	Description
RegionId, Region	Please see the instance region description; for example, for the Guangzhou region, `RegionId` is `1`, and `Region` is `gz`
Dimensions	Enter the CVM instance ID, which can be obtained through the DescribeInstancesAPI API. The input parameter format is {"unInstanceId":"ins-xxxxxxx"}
EventDimensions	Enter the globally unique instance ID, which can be obtained through the DescribeInstancesAPI API. The input parameter format is {"uuid":"9d51a69e-0e4a-4120-ae58-9c073c851e24"}

7. Enter the PolicyId (Response.PolicyId) returned in the [Creating alarm policy](#) step in PolicyId, such as policy-zg2sk27j.
8. After entering the above parameters, click **Online Call > Send Request**. The following figure shows that the alarm policy is successfully bound.
9. After the creation is successful, you can view the number of instances associated with the corresponding alarm policy on the [Alarm Policy](#) page in the Tencent Cloud Observability Platform console.