ALIBABA CLOUD

Alibaba Cloud

CloudMonitor

Best Practices

Document Version: 20220713

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Receive alert notifications from a DingTalk group

This topic describes how to receive alert notifications from a DingTalk group.

Prerequisites

An alert contact is created. For more information, see Create an alert contact or alert contact group.

Context

After you add the webhook URL of a DingTalk chatbot to an existing alert contact, you can receive alert notifications from a DingTalk group.

Step 1: Create a DingTalk chatbot in DingTalk for PC

1. Start DingTalk for PC and go to the DingTalk group from which you want to receive alert notifications.
2. Click the **Group Settings** icon in the upper-right corner.
3. In the **Group Settings** panel, click **Group Assistant**.
4. In the **Group Assistant** panel, click **Add Robot**.
5. In the **Please choose which robot to add** section of the **ChatBot** dialog box, click **Custom**.
6. In the **Robot details** dialog box, click **Add**.
7. In the **Add Robot** dialog box, set the required parameters.
   - Enter a chatbot name, for example, CloudMonitor alert notifications.
   - Select **Custom Keywords** and add the following keywords: CloudMonitor, Cloud Service, Monitor, ECS, and Alert.
8. Select **I have read and accepted <<DingTalk Custom Robot Service Terms of Service>>** and click **Finished**.
9. Click **Copy** to copy the webhook URL.

Step 2: Add the webhook URL of the DingTalk chatbot to an alert contact

1.
2. In the left-side navigation pane, choose **Alerts > Alert Contacts**.
3. On the **Alert Contacts** tab, click **Edit** in the **Actions** column of the required alert contact.
4. In the **Set Alert Contact** panel, enter the webhook URL of the DingTalk chatbot.
5. Verify the parameters and click **OK**.

2.Configure alert rules for application groups by using alert templates

If your Alibaba Cloud account has a large number of cloud resources, you can manage these cloud resources by using application groups. To configure an alert rule for multiple application groups, you can create an alert template.

Context

This topic describes how to configure alert rules for Elastic Compute Service (ECS), ApsaraDB RDS, and Server Load Balancer (SLB) instances by using alert templates and application groups. CloudMonitor monitors these instances based on the configured alert rules.

Procedure

1. Create an alert contact.
   i.
   ii. In the left-side navigation pane, choose **Alerts > Alert Contacts**.
   iii. On the **Alert Contacts** tab, click **Create Alert Contact**.
   iv. In the **Set Alert Contact** panel, enter the name, email address, and DingTalk chatbot of the alert contact, and make sure that the **Alert Notification Information Language** parameter is set to the default value **Automatic**.

   > **Note** **Automatic** indicates that CloudMonitor automatically selects the language of alert notifications based on the language that you use to create your Alibaba Cloud account.

   v. Verify the parameters and click **OK**.
   vi. Optional. Activate the email address of the alert contact.

   By default, the email address of the alert contact is in the **Pending Activation** state. After the alert contact receives an email that contains the activation link, the alert contact must activate the email address within 24 hours. Otherwise, the alert contact cannot receive alert notifications. After the email address is activated, you can view the email address in the alert contact list.
2. Create an alert contact group. For example, you can create a group named InventoryManagementAlertGroup.
   i. On the **Alert Contacts** page, click the **Alert Contact Group** tab.
   ii. On the **Alert Contact Group** tab, click **Create Alert Contact Group**.
   iii. In the **Create Alert Contact Group** panel, enter a name for the alert contact group and add alert contacts to the alert contact group.
   iv. Click **Confirm**.
3. Create an application group. For example, you can create an application group named InventoryManagementOnlineEnvironment.

i. In the left-side navigation pane, click **Application Groups**.

ii. On the **Application Group** tab, click **Create Application Group** in the upper-right corner.

iii. In the **Create Application Group** panel, set the **Creation Method** parameter to **Create Based on Instance Name**, set the **Application Group Name** parameter to **InventoryManagementOnlineEnvironment**, and set the **Alert Contact Group** parameter to **InventoryManagementAlertGroup**. In the Dynamically Add Instances section, configure the rules are used to dynamically match the names of ECS, ApsaraDB RDS, and SLB instances.

iv. Click **OK**.

4. Create an alert template and apply it to an application group. For example, you can create an alert template named E-commerceBackgroundModuleTemplate.

i. In the left-side navigation pane, choose **Alerts > Alert Templates**.

ii. On the **Alert Templates** page, click **Create Alert Template**.

iii. In the **Create/Modify Alert Template** panel, set the **Template Name** parameter to **E-commerceBackgroundModuleTemplate**, and configure alert rules for ECS, ApsaraDB RDS, and SLB instances.

iv. Click **OK**.

v. In the **Alert Template Created/Modified** message, click **OK**.

vi. In the **Apply Templates to Groups** dialog box, select **InventoryManagementOnlineEnvironment** from the Select Groups drop-down list. Then, set the **Mute For**, **Effective From**, **Alert Callback**, and **Priority** parameters.

vii. Click **Confirm**.

viii. In the **Apply Templates to Groups** message, click **Confirm**.

5. View the health status of each instance that matches the alert rules in the application group.

i. In the left-side navigation pane, click **Application Groups**.

ii. On the **Application Group** tab, click the name of the application group that you want to manage.

iii. In the left-side navigation pane, click **Group Resources**.

You can view the health status of each instance that matches the alert rules in the application group.

If an instance has not triggered alerts, the **Health Status** column of the instance shows  . If an instance has triggered alerts, the **Health Status** column of the instance shows  .

3.Monitor resources based on tags

Large enterprises or organizations may maintain thousands of resources. If you use application groups to manage these resources, you must create thousands of applications groups. Manual maintenance is time-consuming and error-prone. CloudMonitor allows you to attach tags to resources, classify and manage resources based on tags, and automate resource monitoring. Tag-based monitoring helps you reduce monitoring costs.

Prerequisites

Tags are attached to resources of Alibaba Cloud services based on business needs.

Context

When you use CloudMonitor to manage resources based on tags, take note of the following limits:

- You can only use tags to manage Elastic Compute Service (ECS), ApsaraDB RDS, and Server Load Balancer (SLB) instances. Network interface controllers (NICs) and disks cannot be tagged.
- An application group supports a maximum of 3,000 resources for each Alibaba Cloud service. Resources are added to an application group in a random order. If the number of resources reaches the upper limit, your resources can no longer be added to the application group.
- You can view the monitoring charts of an application group five minutes after the application group is created.
- The system automatically generates alerts based on alert rules five minutes after an application group is created.

Attach the cloudmonitor-group tag to resources

When you create Alibaba Cloud resources, you can attach the cloudmonitor-group tag to the resources. Then, you can manage the resources in the CloudMonitor console. CloudMonitor automatically creates an application group for this tag. You can view the monitoring charts of the application group and manage the resources.

1. Attach the cloudmonitor-group tag when you create resources.
2. View the application group that is automatically created for the tag in the CloudMonitor console.

> **Note** For the automatically created application group, the **Alert Contact Group** parameter is set to **Default Contact Group** and the Template Name parameter is set to **Basic Template** by default. You can modify the parameters based on your business needs.

Specify tags in the CloudMonitor console

If you have attached tags other than cloudmonitor-group to Alibaba Cloud resources, you can create application groups in the CloudMonitor console based on the tags. Then, you can manage resources in the application groups based on your needs.

1.
2. Create an application group based on tags.

> **Note** After you create an application group, wait 2 minutes and then view the generated application group.

i. In the left-side navigation pane, click **Application Groups**.

ii. On the **Application Groups** tab, click **Create Application Group** in the upper-right corner.

iii. In the **Create Application Group** panel, set parameters for the application group.

When you create an application group, set the following parameters:

- Set the **Creation Method** parameter to **Create Based on Tags**. The system automatically generates an application group name.
- By default, **Default Contact Group** is selected from the **Alert Contact Group** drop-down list. You can select one or more alert contact groups based on your business needs.
- By default, **Basic Template** is selected from the Alert Template drop-down list. You can select one or more alert templates based on your business needs.
- You can set the **Resource Tag Key** and **Tag Value** parameters based on your business needs.
- Turn on **Initialize Agent Installation**. CloudMonitor automatically installs the CloudMonitor agent on the instances that belong to the application group.

iv. Click **OK**.

On the Application Groups tab, you can select **Resource Tag** from the drop-down list to filter the specified resources.

4.Use Grafana to view the monitoring data

This topic describes how to use Grafana to view the monitoring data in a visualized manner.

Procedure

1. Install the Grafana software.

   > **Note** The following example demonstrates how to install Grafana on CentOS. For more information about how to install Grafana on other operating systems, see Install Grafana.

   i. Log on to the server as the root user.

   ii. Run the following commands to install Grafana:

   > **Note** For more information about the software versions of Grafana and the operating systems that Grafana supports, visit the Grafana download page. In the following example, the Grafana installation package is *grafana-8.0.6-1.x86_64.rpm*, which indicates Grafana 8.0.6 for Linux.

   - Method 1:

     yum install *https://dl.grafana.com/oss/release/grafana-8.0.6-1.x86_64.rpm*

   - Method 2:

     wget *https://dl.grafana.com/oss/release/grafana-8.0.6-1.x86_64.rpm*

     sudo yum localinstall *grafana-8.0.6-1.x86_64.rpm*

   iii. Run the following command to start the Grafana service:

   service grafana-server start

2. Optional. Install Grafana panel plug-ins.

   If you need to view the monitoring data on a Grafana panel, such as **Pie Chart**, **Gantt**, or **Worldmap Panel**, you must install the corresponding panel plug-in. For more information about how to install Grafana panel plug-ins, visit the Grafana panel plug-in page.

3. Install the CloudMonitor data source plug-in.

   > **Note** The latest version of the plug-in is v2.0.1. Alert rules cannot be set for the monitoring data in this version.

   i. Run the following commands to download the plug-in to the */var/lib/grafana/plugins/* directory:

   cd /var/lib/grafana/plugins/

   wget https://github.com/aliyun/aliyun-cms-grafana/releases/download/*v2.0/aliyun_cms_grafana_datasource_v2.0.1.tar.gz*

ii. Run the following command to decompress the plug-in to the *aliyun_cms_grafana_datasource* directory:

tar -xzf aliyun_cms_grafana_datasource_v2.0.tar.gz

iii. Configure the plug-in.

a. Run the following commands to open the configuration file named *defaults.ini* in */usr/share/grafana/conf*:

cd /usr/share/grafana/conf

vi defaults.ini

b. Set the `allow_loading_unsigned_plugins` parameter to *aliyun_cms_grafana_datasource*. This allows Grafana to run the CloudMonitor data source plug-in that is not signed.

The following code demonstrates how to set the allow_loading_unsigned_plugins parameter to aliyun_cms_grafana_datasource:

```
allow_loading_unsigned_plugins = aliyun_cms_grafana_datasource
```

c. Press the Esc key, enter *:wq*, and then press the ENTER key to save and close the *defaults.ini* file.

iv. Run the following command to restart the Grafana service:

service grafana-server restart

4. Create a CloudMonitor data source.

Log on to Grafana after it is installed. The default port is 3000 and the default username is admin.

> **Notice** To prevent security risks, we recommend that you change the password the first time you log on to Grafana.

i. Log on to Grafana.

The format of the logon URL is `https://Grafana server IP address:3000`. For example, it can be https://192.168.XX.XX:3000.

ii. In the left-side navigation pane, click the  icon.

iii. On the **Data Sources** tab, click **Add data source** in the upper-right corner.

iv. On the **Add data source** page, click **CMS Grafana Service** at the bottom.

v. Enter the name and account information of the CloudMonitor data source.

Parameter	Description
Name	The name of the data source. You can use the default name **CMS Grafana Service**.
Aliyun UserId	The ID of your Alibaba Cloud account.
AccessKeyId	The AccessKey ID of your Alibaba Cloud account or a RAM user within the Alibaba Cloud account. For more information about how to obtain an AccessKey ID, see Obtain an AccessKey pair. **Note** The RAM user must be created by the current Alibaba Cloud account and authorized to read CloudMonitor data.
AccessKey	The AccessKey secret of your Alibaba Cloud account or a RAM user within the account. For more information about how to obtain an AccessKey secret, see Obtain an AccessKey pair. **Note** The RAM user must be created by the current Alibaba Cloud account and authorized to read CloudMonitor data.

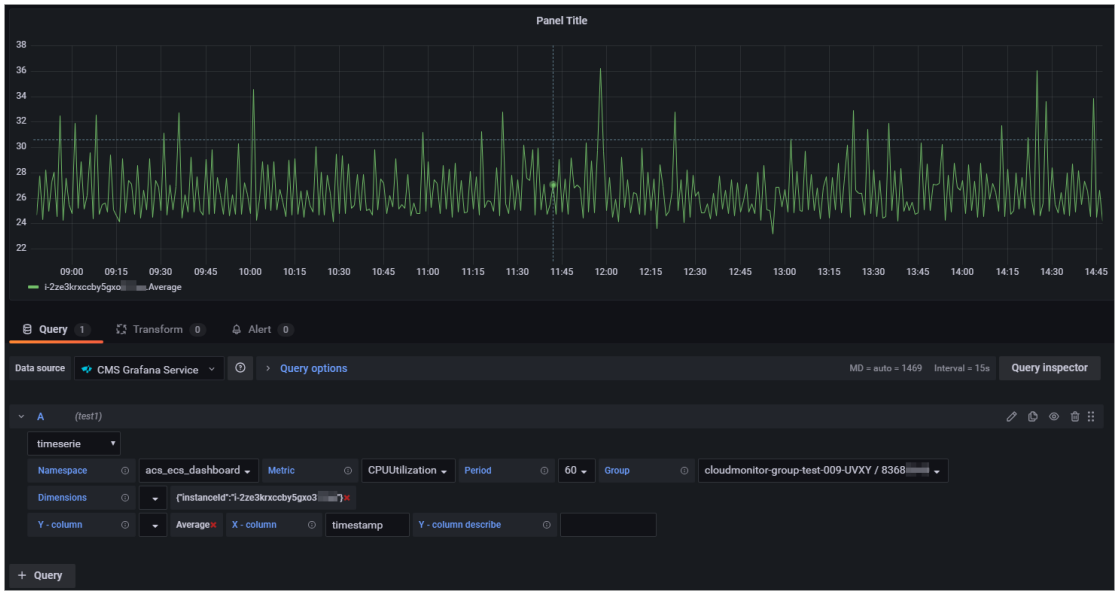
vi. Click **Save & Test**.

5. Add a dashboard and a monitoring chart.

i. In the left-side navigation pane, click the icon.

ii. On the **New dashboard** page, click **Add an empty panel**.

iii. On the **Query** tab, select the **CMS Grafana Service** data source and configure a metric for a specified Alibaba Cloud service.



The following table describes the parameters that are used to configure a metric for a specified Alibaba Cloud service.

Parameter	Description
Namespace	The namespace of the monitoring data that is reported. Specifies the value in the format of `acs_Service name`. For more information, see Appendix 1: Metrics.
Metric	The name of the metric for which the monitoring data is reported. For more information, see Appendix 1: Metrics.
Period	The intervals at which the monitoring data is reported. Unit: seconds. For more information, see Appendix 1: Metrics.
Group	The name and ID of the application group for which data of the specified metric is reported.
Dimensions	The dimensions that specify the resources for which the monitoring data is reported. Specify a dimension in the format of a key-value pair, such as `instanceId:i-2ze2d6j5uhg20x47****`. You can specify multiple dimensions at a time. For more information, see Appendix 1: Metrics.
Y-column	The statistical method that is used to report the monitoring data, such as Average, Maximum, Minimum, and Sum. For more information, see Appendix 1: Metrics.

iv. In the right-side pane, set the name, type, and layout of the monitoring chart.

v. In the upper-right corner, click **Apply**.

The monitoring chart is created.

vi. In the upper-right corner, click the  icon and set the name of the dashboard and the directory where it resides.

vii. Click **Save**.

The dashboard is created.

6. View monitoring data.

i. In the left-side navigation pane, choose  > **Manage**.

ii. On the **Manage** tab, click the dashboard.

View all monitoring charts on the dashboard.

5.Log monitoring

5.1. Use the log monitoring feature to monitor log keywords and configure alert rules

You can use the log monitoring feature of CloudMonitor to calculate the number of times that a specific keyword appears in the logs that are collected by Log Service. You can also use the log monitoring feature to configure an alert rule for the keyword. If the number of times that the keyword appears meets a specified condition, an alert is triggered. This topic describes how to create a metric to monitor a specific keyword in logs and how to configure an alert rule for the keyword.

Prerequisites

On-premises logs are collected and stored in Log Service. For more information, see Log Service.

Context

The following example shows the sample logs that are collected by Log Service:

```
2017-06-21 14:38:05 [INFO] [impl.FavServiceImpl] execute_fail and run time is 100msuserid=
2017-06-21 14:38:05 [WARN] [impl.ShopServiceImpl] execute_fail, wait moment 200ms
2017-06-21 14:38:05 [INFO] [impl.ShopServiceImpl] execute_fail and run time is 100ms,reason
:user_id invalid
2017-06-21 14:38:05 [INFO] [impl.FavServiceImpl] execute_success, wait moment ,reason:user_
id invalid
2017-06-21 14:38:05 [WARN] [impl.UserServiceImpl] execute_fail and run time is 100msuserid=
2017-06-21 14:38:06 [WARN] [impl.FavServiceImpl] execute_fail, wait moment userid=
2017-06-21 14:38:06 [ERROR] [impl.UserServiceImpl] userid=, action=, test=, wait moment ,re
ason:user_id invalid
2017-06-21 14:38:06 [ERROR] [impl.ShopServiceImpl] execute_success:send msg,200ms
```

In this example, ERROR is used as the keyword to describe how to use the log monitoring feature to create a metric and configure an alert rule to monitor the keyword. The key is level and the value is the content of a log. The following table describes the key-value pairs that are extracted from the sample logs.

Key	Value
level	`2017-06-21 14:38:05 [INFO] [impl.FavServiceImpl] execute_fail and run time is 100msuserid=`
level	`2017-06-21 14:38:05 [WARN] [impl.ShopServiceImpl] execute_fail, wait moment 200ms`
level	2017-06-21 14:38:06 [ERROR] [impl.ShopServiceImpl] execute_success:send msg,200ms

Procedure

1. Optional. Grant CloudMonitor the permissions to access Log Service.

   The first time you use the log monitoring feature, you must grant CloudMonitor the permissions to access Log Service.

   i.
   ii. In the left-side navigation pane, click **Log Monitoring**.
   iii. In the **Service-linked Role for CloudMonitor** dialog box, click **OK**.

2. Create a log monitoring metric to monitor the logs in which the value of the level field contains the keyword ERROR.

   i. In the upper-left corner of the **Log Monitoring** page, click **Create Log Monitoring Metric**.
   ii. In the **Associate Resource** step, set the parameters and click **Next**.
   iii. In the **Define Metric** step, set the parameters and click **Next**.

   iv. In the **Configure Alert Rule** step, configure an alert rule to monitor the keyword ERROR and click **Next**.
   v. In the **Creation Result** step, click **Close**.

3. View the monitoring data of the keyword ERROR.

   After you create the log monitoring metric, wait for 3 to 5 minutes. On the **Log Monitoring** page, find the metric whose monitoring chart you want to view and click the icon in the **Actions** column.

4. View the alert notifications that are sent for the keyword ERROR.

   If an ERROR-level log appears in Log Service, CloudMonitor sends an alert notification.

5.2. Use the log monitoring feature to monitor business logs and configure alert rules

This topic describes how to analyze logs that are collected by Log Service and configure an alert rule for the logs.

Prerequisites

On-premises logs are collected and stored in Log Service. For more information, see Log Service.

Procedure

1. Optional. Grant CloudMonitor the permissions to access Log Service.

   The first time you use the log monitoring feature, you must grant CloudMonitor the permissions to access Log Service.

   i.
   ii. In the left-side navigation pane, click **Log Monitoring**.

iii. In the **Service-linked Role for CloudMonitor** dialog box, click **OK**.

2. Create a log monitoring metric.
   i. In the upper-left corner of the **Log Monitoring** page, click **Create Log Monitoring Metric**.
   ii. In the **Associate Resource** step, select the resources that you want to associate and click **Next**.
   iii. In the **Define Metric** step, set the parameters and click **Next**.

   The following table describes the parameters.

   In the **Define Metric** step, click **Preview** to preview the aggregated log data of the last minute by using the specified statistical methods. Only the most recent 100 logs of the last minute are analyzed. The following figure shows a sample preview result.

   iv. In the **Configure Alert Rule** step, set the parameters and click **Next**.
   v. In the **Creation Result** step, click **Close**.
3. View the monitoring data of the metric.

   After you create the log monitoring metric, wait for 3 to 5 minutes. On the **Log Monitoring** page, find the metric whose monitoring chart you want to view and click the icon in the **Actions** column.

   
4. View the alert notifications that are sent for the metric.

   If the metric meets the specified condition in the alert rule, CloudMonitor sends an alert notification.

5.3. Use the log monitoring feature to analyze website access logs and configure alert rules

This topic describes how to use the log monitoring feature to create metrics for the queries per second (QPS), HTTP status code, and response time of website access logs, such as NGINX logs and Apache logs. This topic also describes how to configure alert rules for the metrics.

Prerequisites

- Website access logs are collected and stored in Log Service. For more information, see Log Service.
- CloudMonitor is granted the permissions to access Log Service. For more information, see Authorize CloudMonitor to access Log Service.

Background information

The following example shows a sample website access log that is collected by Log Service:

```
192.168.XX.XX - - [10/Jul/2019:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.032 129 200 168
"-" "Wget/1.11.4 Red Hat modified"
```

The following table describes the fields that are extracted from the sample log.

Field	Sample field	Description
time	2019-06-10 15:51:09	The time when the log was generated.
rt	0.032	The response time of the accessed website. Unit: seconds.
URL	/ubuntu.iso	The URL of the accessed website.
status	200	The HTTP status code that is returned when the website was accessed.
body	168	The size of the HTTP response body. The response header is not included.

Calculate the total QPS for all websites or calculate the QPS for each website

1.
2. In the left-side navigation pane, click **Log Monitoring**.
3. In the upper-left corner of the **Log Monitoring** page, click **Create Log Monitoring Metric**.
4. In the **Associate Resource** step, set the **Region**, **Project**, and **Logstore** parameters. Then, click **Next**.
5. In the **Define Metric** step, specify the statistical method that you want to use to analyze logs and the application group to which the metric belongs. Then, click **Next**.
   - **Metric Name**: Enter a name for the metric.
   - **Statistical Method**: Select **status** and **countps**.
   - **Group-by**: If you want to calculate the total QPS for all websites, you do not need to set this parameter. If you want to calculate the QPS for each website, set this parameter to **URL**.

     > **Note** CloudMonitor can analyze up to 1,000 URLs in your website access logs.
6. In the **Configure Alert Rule** step, set the parameters and click **Next**.
7. In the **Creation Result** step, click **Close**.
8. On the **Log Monitoring** page, click the name of the metric to view the website QPS on the monitoring chart.

Calculate the number of HTTP requests whose status code is 4XX or 5XX

1.
2. In the left-side navigation pane, click **Log Monitoring**.

3. In the upper-left corner of the **Log Monitoring** page, click **Create Log Monitoring Metric**.
4. In the **Associate Resource** step, set the **Region**, **Project**, and **Logstore** parameters. Then, click **Next**.
5. In the **Define Metric** step, specify the statistical method that you want to use to analyze logs and the application group to which the metric belongs. Then, click **Next**.
   - **Metric Name**: Enter a name for the metric.
   - **Statistical Method**: Select **status** and **Count**.
   - **Log Filter**: Set the filter condition to `status>=400` and `status<=599`.
   - **Group-by**: If you want to calculate the number of HTTP requests whose status code is 4XX or 5XX, you do not need to set this parameter. If you want to calculate the number of HTTP request for each status code that ranges from 400 to 599, set this parameter to **URL**.

     > **Note** CloudMonitor can analyze up to 1,000 URLs in your website access logs.

   
6. In the **Configure Alert Rule** step, set the parameters and click **Next**.
7. In the **Creation Result** step, click **Close**.
8. On the **Log Monitoring** page, click the name of the metric to view the number of HTTP requests whose status code is 4XX or 5XX on the monitoring chart.

6.Monitor the availability of services in a VPC

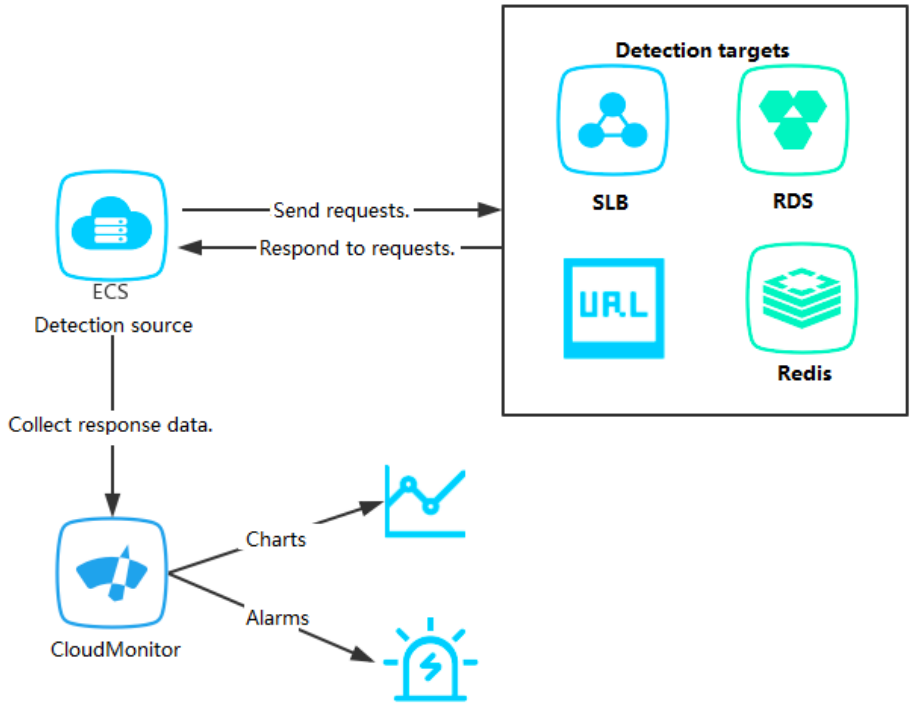
This topic describes how to use CloudMonitor to monitor the availability of services in a virtual private cloud (VPC).

Context

As an increasing number of users migrate their services from the classic network to VPCs that are safer and more reliable, users need to monitor the availability of services in VPCs. This topic describes how to monitor the availability of services in a VPC, including Elastic Compute Service (ECS), ApsaraDB RDS, ApsaraDB for Redis, and Server Load Balancer (SLB).

Before you begin

The following figure shows how to monitor the availability of services in a VPC.



Before you can monitor the availability of services in a VPC, you must install the CloudMonitor agent on the ECS instances that will be used as monitoring nodes. To monitor the availability of a service in a VPC, create an availability monitoring task in the CloudMonitor console, select a monitoring node, and specify the URL or port of the monitored target. After the availability monitoring task is created, the CloudMonitor agent on the monitoring node sends an HTTP request or a Telnet request to the URL or port every minute. The CloudMonitor agent collects the response time and status codes and reports the monitoring results to CloudMonitor. CloudMonitor displays the monitoring results in a chart and generates alerts if the connection times out or the monitoring fails.

Procedure

> **Note**
>
> - You must install the CloudMonitor agent on the ECS instances that will be used as monitoring nodes.
> - You must create an application group and add the monitoring nodes to the group.

1.
2. In the left-side navigation pane, click **Application Groups**.
3. On the **Application Groups** tab, click the name of the application group.
4. In the left-side navigation pane, click **Availability Monitoring**.
5. Click **Add Availability Monitoring**.
6. In the **Create Task** step on the Create/Modify Availability Monitoring panel, set the parameters based on your needs.

Parameter	Description
Task Name	The name of the availability monitoring task.
Monitoring Nodes	The IDs of the instances that you want to monitor.
Monitored Target	The object that you want to monitor. Valid values: ○ **URL or IP Address**: Select this option only when you need to monitor ECS instances. ○ **ApsaraDB RDS**: Select this option only when you need to monitor ApsaraDB RDS instances. ○ **ApsaraDB for Redis**: Select this option only when you need to monitor ApsaraDB for Redis instances.
Detection Type	The method that you want to use to monitor the object. If you set the **Monitored Target** parameter to **URL or IP Address**, you can select one of the following methods: ○ **HTTP(S)**: If you select this option, enter the URL of the object that you want to monitor. ○ **TELNET**: If you select this option, enter the IP address of the object that you want to monitor. ○ **PING**: If you select this option, enter the IP address of the object that you want to monitor. If you set the **Monitored Target** parameter to **ApsaraDB RDS** or **ApsaraDB for Redis**, you can select one of the following methods: ○ **TELNET**: If you select this option, enter the instance ID and connection URL of the object that you want to monitor. ○ **PING**: If you select this option, enter the instance ID and connection URL of the object that you want to monitor.

Parameter	Description
Request Method	The request method. Valid values: **HEAD**, **GET**, and **POST**. **Note** This parameter is required if you set the **Monitored Target** parameter to **URL or IP Address** and the **Detection Type** parameter to **HTTP(S)**.
Monitoring Frequency	The interval at which detection requests are sent. Valid values: 15 Seconds, 30 Seconds, 1 Minutes, 2 Minutes, 5 Minutes, 15 Minutes, 30 Minutes, and 60 Minutes. For example, if you select 1 Minutes, CloudMonitor sends a detection request to the monitored object every minute. **Note** This parameter is required if you set the **Monitored Target** parameter to **URL or IP Address** and the **Detection Type** parameter to **HTTP(S)**.
Headers	The HTTP or HTTPS headers of detection requests for site monitoring. A header must be in the `parameter1:value1` format. Separate multiple headers with commas (,). **Note** This parameter is required if you set the **Monitored Target** parameter to **URL or IP Address** and the **Detection Type** parameter to **HTTP(S)**.
POST Content	The content of POST requests for site monitoring. The content must be in the `parameter1=value1&parameter2=value2` format and must be English. **Note** This parameter is required if you set the **Monitored Target** parameter to **URL or IP Address**, the **Detection Type** parameter to **HTTP(S)**, and the **Request Method** parameter to **POST**.

Parameter	Description
Match Response Content	The response content that you want to match and the method used to match the response content. If you specify response content, the monitoring task checks whether the first 64 KB of the HTTP response body contains the response content that you specify. Valid values: ○ **Generate Alerts If Response Contains** ○ **Generate Alerts If Response Does Not Contain** **Note** This parameter is required if you set the **Monitored Target** parameter to **URL or IP Address** and the **Detection Type** parameter to **HTTP(S)**.

Note

- To monitor whether local processes on ECS instances in a VPC respond, select the ECS instances to be monitored in the Monitoring Nodes section, set the Monitored Target parameter to URL or IP Address, and enter the addresses in `localhost:port/path` format in the Detection Type section.
- To monitor whether an SLB instance in a VPC responds, select an ECS instance that resides in the same VPC as the SLB instance in the Monitoring Nodes section, set the Monitored Target parameter to URL or IP Address, and enter the address of the SLB instance in the Detection Type section.
- To monitor whether an ApsaraDB RDS or ApsaraDB for Redis instance in a VPC responds to an ECS instance, add the ApsaraDB RDS or ApsaraDB for Redis instance to the application group of the ECS instance, select the ECS instance in the Monitoring Nodes section, and set the Monitored Target parameter to ApsaraDB RDS or ApsaraDB for Redis.

7. Click **Next**.
8. In the **Configure Alert Rule** step, set the parameters.

Parameter	Description
Status Code	If the status code reaches the specified value, an alert is triggered. An alert is triggered regardless of which value of the **Status Code** parameter or the **Response Time** parameter reaches the threshold. CloudMonitor sends alert notifications to the alert contact group of the application group.

Parameter	Description
Response Time	If the response time reaches the specified value, an alert is triggered. If the value of the **Status Code** parameter or the **Response Time** parameter meets the threshold value, an alert is triggered. CloudMonitor sends alert notifications to the alert group of a specified application group.
Notification Methods	The methods that are used to send alert notifications. Valid values: Info (Email + DingTalk)
Alert Callback	The URL that is used to receive the alert notifications sent from CloudMonitor by using HTTP POST requests. The URL must be accessible over the Internet. You can enter only an HTTP URL. For more information about how to set callback URLs, see Use the alert callback feature to send notifications about threshold-triggered alerts.
Mute for	The interval at which CloudMonitor sends alert notifications until the alert that is triggered based on the alert rule is cleared. Valid values: 5 Minutes, 15 Minutes, 30 Minutes, 60 Minutes, 3 Hours, 6 Hours, 12 Hours, and 24 Hours. An alert is triggered when the conditions of an alert rule are met. CloudMonitor does not resend an alert notification when the alert is triggered again within the mute period. CloudMonitor starts to resend alert notifications if the alert is not cleared after the mute period ends.
Effective Time	The validity period of the alert rule. The system sends alert notifications only within the validity period of an alert rule and records events when the validity period expires.

9. Click **OK**.
10. Find the monitoring task and click in the **Actions** column.

    You can view the monitoring details in the monitoring charts.

7.Query monitoring data by calling API operations

This topic describes how to call API operations to query monitoring data of various Alibaba Cloud services.

Large enterprises have their own operations and maintenance (O&M) and monitoring systems. When they migrate business to Alibaba Cloud, these enterprises need to integrate monitoring data of cloud resources with their existing systems. This topic describes how to use Cloud Monitor API operations to query monitoring data of various services. This way, you can integrate monitoring data of Alibaba Cloud with your existing systems.

API operations for querying monitoring data of metrics

Cloud Monitor provides the following operations to query monitoring data of metrics:

- Operation used to query services: queries services that can be monitored by Cloud Monitor. For more information, see DescribeProjectMeta.
- Operation used to query metrics: queries metrics that are available for a monitored service. For more information, see DescribeMetricMetaList.
- Operations used to query monitoring data: query monitoring data based on services and metrics. For more information, see DescribeMetricMetaList and DescribeMetricLast.

Usage notes:

- The DescribeMetricList and DescribeMetricLast operations allow you to query data of a specific metric for all your instances. You can create multiple threads to query the monitoring data of multiple metrics at a time. Alternatively, you can create a single thread to obtain the monitoring data of multiple metrics one by one.
- The DescribeMetricList operation supports a maximum of 20 queries per second (QPS), whereas the DescribeMetricLast operation supports a maximum of 30 QPS.
- The DescribeMetricLast operation is applicable to scenarios where you need to obtain the most recent monitoring data at regular intervals. The time window automatically slides forward. For each window, the most recent record is retrieved.
- Services may take some time to report monitoring data to Cloud Monitor. The delay varies with different services. We recommend that you extend the time window by 5 to 10 minutes when you call the DescribeMetricLast operation to query the latest data.
- Cloud Monitor retains data that is obtained every few seconds for 7 days, and data that is obtained every few minutes for 31 days.
- If you want to query the aggregate data of all your instances, you do not need to specify the Dimensions parameter.

Example

The following example demonstrates how to call the DescribeMetricLast operation to query the latest monitoring data and the DescribeMetricList operation to query the monitoring data in a specified time range.

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
```

```
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.cms.model.v20190101. *;
/**
 * You can call the DescribeMetricList operation to query the monitoring data of a specifie
d instance in a specified time range.
 * The DescribeMetricList operation can query the monitoring data of multiple instances at
a time.
 * To query the monitoring data of multiple instances in a specified time range, specify th
ese instances for the query. You can specify a maximum of 10 instances at a time.
 * Query monitoring data in a specified time range.
 */
public class DescribeMetricList {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>",
"<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        DescribeMetricListRequest request = new DescribeMetricListRequest();
        // You can call the DescribeMetricMetaList and DescribeProjectMeta operations to qu
ery the namespace and metric.
        request.setNamespace("acs_ecs_dashboard");
        request.setMetricName("cpu_total");
        // The Period parameter is set to 60, which specifies that monitoring data is obtai
ned every 60 seconds. The value of the Period parameter varies with metrics. The period of
most metrics are set to 60 seconds by default.
        request.setPeriod("60");
        // The number of entries to return on each page. A maximum of 1,000 entries can be
returned for each query.
        request.setLength("1000");
        // The beginning of the time range to query.
        request.setStartTime("2019-07-22 11:00:00");
        // The end of the time range to query.
        request.setEndTime("2019-07-22 12:00:00");
        // Set the Dimensions parameter to filter monitoring data. The value can be a JSON
array or a JSON object.
        request.setDimensions("[{\"instanceId\":\"i-8vb******\"}]");
        try {
            DescribeMetricListResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.cms.model.v20190101. *;
/**
 * Query the latest monitoring data.
 **/
public class DescribeMetricLast {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>",
"<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        DescribeMetricLastRequest request = new DescribeMetricLastRequest();
         // You can call the DescribeMetricMetaList and DescribeProjectMeta operations to q
uery the namespace and metric.
        request.setNamespace("acs_ecs_dashboard");
        request.setMetricName("cpu_total");
        // Set the Dimensions parameter to filter monitoring data. The value can be a JSON
array or a JSON object.
        request.setDimensions("[{\"instanceId\":\"i-8vb6p*****\"}]");
        // The number of entries to return on each page. A maximum of 1,000 entries can be
returned for each query.
        request.setLength("1000");
        // The beginning of the time range to query.
        request.setStartTime("2019-07-22 11:00:00");
        // The end of the time range to query.
        request.setEndTime("2019-07-22 12:00:00");
        request.setPeriod("60");
        try {
            DescribeMetricLastResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

8.Automate O&M based on status change events of ECS instances

This topic describes how CloudMonitor automatically processes the status change events of Elastic Compute Service (ECS) instances by using Message Service (MNS) queues.

Prerequisites

- A queue is created in the MNS console, for example, ecs-cms-event.

  For more information, see Create a queue.
- A system event-triggered alert rule is created in the CloudMonitor console.

  For more information, see Create a system event-triggered alert rule.
- Python dependencies are installed.

  All the code in this topic is written in Python 3.6. You can also use other programming languages, such as Java and PHP.

  For more information, see Install CloudMonitor SDK for Python.

Context

In addition to the existing system events, CloudMonitor supports the status change events for ECS. The status change events include interruption notification events that are applied to preemptible instances. A status change event is triggered when the status of an ECS instance changes. The status changes can be caused by operations that you perform in the ECS console and by calling API operations or using SDKs, auto scaling, overdue payments, and system exceptions.

CloudMonitor provides the following notification methods for system events: MNS queues, Function Compute, callback URLs, and Log Service. This topic uses MNS queues as an example to describe three best practices about how CloudMonitor automatically processes the status change events of ECS instances.

Procedure

CloudMonitor sends all status change events of ECS instances to MNS. MNS receives messages and handles the messages.

- Practice 1: Record all creation and release events of ECS instances

  You cannot query ECS instances that have been released in the ECS console. If you need to query released ECS instances, you can store status change events of all ECS instances in a database or Log Service. When an ECS instance is created, CloudMonitor sends a Pending event. When an ECS instance is released, CloudMonitor sends a Deleted event.

  i. Create a Conf file.

  The Conf file must contain the following parameters: `endpoint`, `access_key`, `access_key_secret`, `region_id` (for example, cn-beijing), and `queue_name`.

> **Note** To obtain the `endpoint`, you can log on to the MNS console, go to the **Queues** page, and then click **Get Endpoint**.

```
class Conf:
    endpoint = 'http://<id>.mns.<region>.aliyuncs.com/'
    access_key = '<access_key>'
    access_key_secret = '<access_key_secrect>'
    region_id = 'cn-beijing'
    queue_name = 'test'
    vsever_group_id = '<your_vserver_group_id>'
```

ii. Use the MNS SDK to develop an MNS client for receiving messages from MNS.

```
# -*- coding: utf-8 -*-
import json
from mns.mns_exception import MNSExceptionBase
import logging
from mns.account import Account
from . import Conf
class MNSClient(object):
    def __init__(self):
        self.account =  Account(Conf.endpoint, Conf.access_key, Conf.access_key_secre
t)
        self.queue_name = Conf.queue_name
        self.listeners = dict()
    def regist_listener(self, listener, eventname='Instance:StateChange'):
        if eventname in self.listeners.keys():
            self.listeners.get(eventname).append(listener)
        else:
            self.listeners[eventname] = [listener]
    def run(self):
        queue = self.account.get_queue(self.queue_name)
        while True:
            try:
                message = queue.receive_message(wait_seconds=5)
                event = json.loads(message.message_body)
                if event['name'] in self.listeners:
                    for listener in self.listeners.get(event['name']):
                        listener.process(event)
                queue.delete_message(receipt_handle=message.receipt_handle)
            except MNSExceptionBase as e:
                if e.type == 'QueueNotExist':
                    logging.error('Queue %s not exist, please create queue before rec
eive message.', self.queue_name)
                else:
                    logging.error('No Message, continue waiting')
class BasicListener(object):
    def process(self, event):
        pass
```

The preceding code is used to receive messages from MNS and delete the messages after the listener is called to consume the messages.

iii. Register a listener to consume events. The following listener generates a log entry after it receives a Pending or Deleted event.

```
 # -*- coding: utf-8 -*-
import logging
from .mns_client import BasicListener
class ListenerLog(BasicListener):
    def process(self, event):
        state = event['content']['state']
        resource_id = event['content']['resourceId']
        if state == 'Panding':
            logging.info(f'The instance {resource_id} state is {state}')
        elif state == 'Deleted':
            logging.info(f'The instance {resource_id} state is {state}')
```

Add the following code to the Main function:

```
mns_client = MNSClient()
mns_client.regist_listener(ListenerLog())
mns_client.run()
```

In the production environment, you can store the events in a database or Log Service for subsequent queries and audits.

- Practice 2: Automatically restart ECS instances that are shut down

  In scenarios where ECS instances may be shut down unexpectedly, you may need to automatically restart the ECS instances.

  You can reuse the MNS client developed in Practice 1 and create another listener. When you receive a Stopped event for an ECS instance, you can run the start command on the ECS instance to start it.

```
# -*- coding: utf-8 -*-
import logging
from aliyunsdkecs.request.v20140526 import StartInstanceRequest
from aliyunsdkcore.client import AcsClient
from .mns_client import BasicListener
from .config import Conf
class ECSClient(object):
    def __init__(self, acs_client):
        self.client = acs_client
    # Start the ECS instance.
    def start_instance(self, instance_id):
        logging.info(f'Start instance {instance_id} ...')
        request = StartInstanceRequest.StartInstanceRequest()
        request.set_accept_format('json')
        request.set_InstanceId(instance_id)
        self.client.do_action_with_exception(request)
class ListenerStart(BasicListener):
    def __init__(self):
        acs_client = AcsClient(Conf.access_key, Conf.access_key_secret, Conf.region_id)
        self.ecs_client = ECSClient(acs_client)
    def process(self, event):
        detail = event['content']
        instance_id = detail['resourceId']
        if detail['state'] == 'Stopped':
            self.ecs_client.start_instance(instance_id)
```

In the production environment, you can listen to Starting, Running, or Stopped events after the start command is run. Then, you can perform further O&M by using a timer and a counter based on whether the ECS instance is started.

- Practice 3: Automatically remove preemptible instances from SLB before they are released

  An interruption notification event is triggered about 5 minutes before a preemptible instance is released. During the 5 minutes, you can perform specific operations to prevent your services from being interrupted. For example, you can remove the preemptible instance from a Server Load Balancer (SLB) instance.

  You can reuse the MNS client developed in Practice 1 and create another listener. When the listener receives the interruption notification event for a preemptible instance, you can call the SLB SDK to remove the preemptible instance from an SLB instance.

```
# -*- coding: utf-8 -*-
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.request import CommonRequest
from .mns_client import BasicListener
from .config import Conf
class SLBClient(object):
    def __init__(self):
        self.client = AcsClient(Conf.access_key, Conf.access_key_secret, Conf.region_id)
        self.request = CommonRequest()
        self.request.set_method('POST')
        self.request.set_accept_format('json')
        self.request.set_version('2014-05-15')
        self.request.set_domain('slb.aliyuncs.com')
        self.request.add_query_param('RegionId', Conf.region_id)
    def remove_vserver_group_backend_servers(self, vserver_group_id, instance_id):
        self.request.set_action_name('RemoveVServerGroupBackendServers')
        self.request.add_query_param('VServerGroupId', vserver_group_id)
        self.request.add_query_param('BackendServers',
                                     "[{'ServerId':'" + instance_id + "','Port':'80','Wei
ght':'100'}]")
        response = self.client.do_action_with_exception(self.request)
        return str(response, encoding='utf-8')
class ListenerSLB(BasicListener):
    def __init__(self, vsever_group_id):
        self.slb_caller = SLBClient()
        self.vsever_group_id = Conf.vsever_group_id
    def process(self, event):
        detail = event['content']
        instance_id = detail['instanceId']
        if detail['action'] == 'delete':
            self.slb_caller.remove_vserver_group_backend_servers(self.vsever_group_id, in
stance_id)
```

Notice

For interruption notification events, set the event name in the following way: `mns_client.regis t_listener(ListenerSLB(Conf.vsever_group_id), 'Instance:PreemptibleInstanceInterruption' )`.

In the production environment, you can apply for another preemptible instance and add it as a backend server to SLB to ensure the performance of your services.