ALIBABA CLOUD

Alibaba Cloud

Application Real-time Monitoring Service

Dashboard and alerting

Document Version: 20210302

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

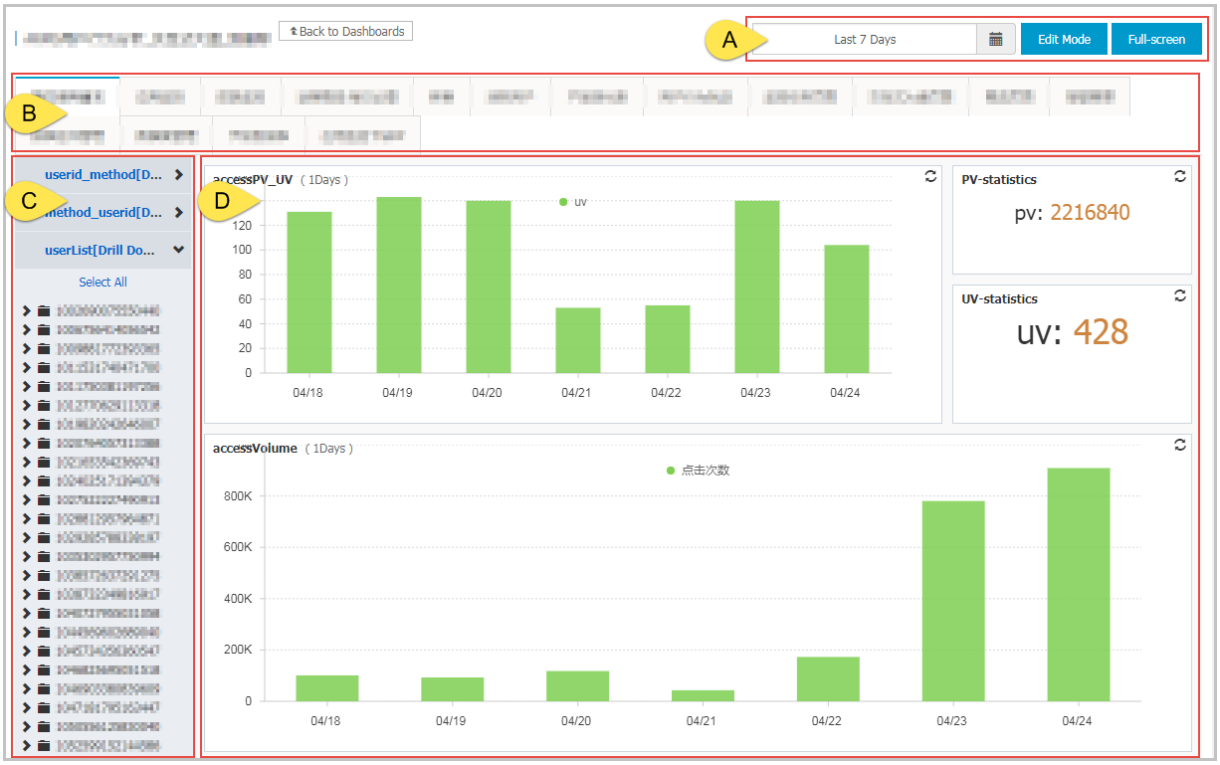
Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major 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.Dashboard overview

This topic introduces the user interface of Dashboards.
In the left-side navigation pane of the console, click **Dashboards**. On the Dashboards page, you can find all dashboards.

A dashboard consists of the control operation area, tab operation area, navigation tree operation area, and dataset presentation area. The following figure shows a complete dashboard.



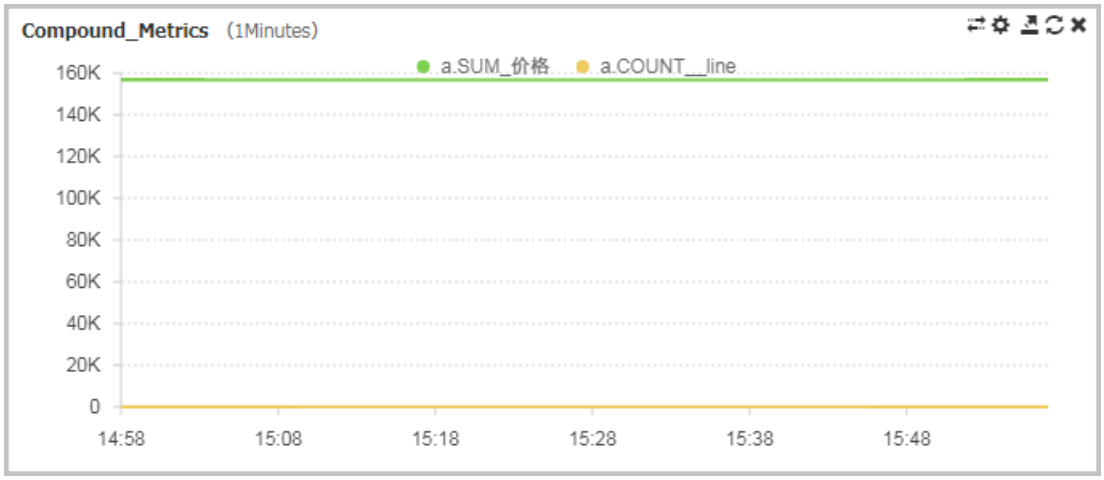
- [A] control operation area: You can add controls, such as a navigation tree, charts, and texts, and specify the time range.
- [B] tab operation area: You can add different tabs to group charts. For example, to show statistics of page views (PV) and unique visitors (UV) of NGINX pages, and also the statistics of error codes on these pages, you can use two separate tabs.
- [C] navigation tree operation area: You can set a navigation tree that has up to three levels. Drill down level by level and click a query dimension to view the relevant data in the right-side dataset presentation module. The navigation tree is automatically generated by ARMS based on user data, or manually imported in JSON format.
- [D] dataset presentation area: The core module of the dashboard. In this area, all data charts under different tabs are displayed based on the selected navigation tree dimension and time range, and multiple tabs can be added.

2.Create a dashboard

This topic describes how to create a dashboard and configure data for it.

Procedure

1. Log on to the ARMS console .
2. In the left-side navigation pane, click **Dashboards**. On the **Dashboards** page, choose **Create Dashboard > Custom Dashboard**.
3. In the **Create Dashboard** dialog box, enter the dashboard name and click **OK**. The new dashboard is displayed in the dashboard list but it does not contain any data. You must configure data for the dashboard.
4. Add a dataset.
   i. On the Dashboards page, click **Edit** in the **Actions** column corresponding to the dashboard that you created.
   ii. In the upper-right corner of the page, click **Interactive Control** and select a graph that you want to add.
   iii. In the New Interactive Chart dialog box, enter the chart name, select a **Dataset**, select the chart type, and specify other parameters as needed. Then, click **OK**. For example, if you set Chart Type to Line, the dataset is displayed in line chart.



5. Add a navigation tree.
   i. On the edit page of the dashboard, choose **Interactive Control > Navigation Tree Components** in the upper-right corner of the page.
   ii. In the Navigation Tree dialog box, enter the **Name** and select the **Dataset Type** and **Dataset**. ARMS automatically imports the multi-dimensional traversal values of this dataset into the **Data** field.
   iii. In the Navigation Tree dialog box, click **OK**. The navigation tree is displayed on the left side of the page.
6. Associate the dataset with the navigation tree.
   i. In the data display section, find the chart that you want to manage and click the gear icon in the upper-right corner of the chart.
   ii. In the **Dataset** section of the dialog box, select **Navigation Tree** from the **Dimension** drop-down list, and click **OK**. The dataset is associated with the navigation tree.

7. View the displayed dataset.
   You can select different dimensions in the navigation tree to view the data of the dataset.
8. In the time module, select **Today**, **This Week**, or **This Month** to view data within the specified time period. You can also specify the start time and end time.
9. After the dashboard is configured, click **Save** in the upper-right corner to save the configuration.
   ARMS automatically saves the configuration every 10 seconds to avoid losing data being edited.

   > **Note**
   > - ARMS automatically saves the configuration every 10 seconds to avoid losing data being edited.
   > - The size and position of the chart in the dashboard can be adjusted.

   - Adjust the size of a chart
     In edit mode, drag the handle in the lower-right corner of a chart to adjust the size.
   - Change the position of a chart
     In edit mode, drag a chart to change its position. After you have moved it to the desired position, release the pointer.

3.Manage a dashboard

This topic describes how to edit and delete a dashboard, and how to configure full-screen playback for a dashboard.

Procedure

1. Log on to the ARMS console .
2. In the left-side navigation pane, click **Dashboards**. On the **Dashboards** page, enter a dashboard name in the search bar and click **Search**.
3. Perform the following operations on the found dashboard:
   - To view details about the dashboard, click the dashboard name or **Browse** in the **Actions** column.
   - To edit the dashboard, click **Edit** in the **Actions** column.
   - To delete the dashboard, click **Delete** in the **Actions** column. In the **Delete** dialog box, click **Delete**.
   - To modify the dashboard name, click **Modify Alias** in the **Actions** column. In the **Edit** dialog box, enter a new dashboard name and click **OK**.
   - To copy the dashboard, choose **More > Copy** in the **Actions** column. In the **Create Dashboard** dialog box, enter a name for the new dashboard and click **OK**. Then, you can find the new dashboard on the Dashboards page and continue to edit the dashboard.
   - To share the dashboard, choose **More > Share** in the **Actions** column. After you turn on **Enable Share** in the **Generate Dashboard Share Link** dialog box, the system generates a share link for other users to open and view the dashboard.

Configure full-screen playback

1. On the **Dashboards** page, click the dashboard name or **Browse** in the **Actions** column.
2. In Browse mode, click **Full Screen** in the upper-right corner.
3. In the **Full Screen Settings** dialog box, select the tab to view in full screen mode and click **OK**.

The dashboard is played back in the full screen mode.

> **Note**
>
> - Press **Esc** to exit the full screen mode.
> - By default, the data is refreshed once every minute. If you have specific requirements, contact DingTalk service account arms160804.

4.Create an alert

By creating alerts, you can set alert rules for specific monitored objects. When a rule is triggered, the system sends an alert notification to the specified contact group in the specified alerting mode. This reminds you to take necessary actions to solve the problem.

Prerequisites

- A monitoring job is created. For more information, see Create an application monitoring job.
- Contacts are created. Only contact groups can be set for the notification receiver of an alert.

Context

Default behaviors of alert notifications:

- To prevent you from receiving a large number of alert notifications in a short period of time, the system sends only one message for repeated alerts within 24 hours.
- If no repeated alerts are generated within 5 minutes, the system sends a recovery email to notify you that the alert has been cleared.
- After a recovery email is sent, the alert status is reset. If this alert arises again, it is deemed as a new one.

An alert widget is essentially a data display method for datasets. When you create an alert widget, a dataset is created to store the underlying data of the alert widget.

> **Note** New alerts take effect within 10 minutes. The alert check may have a delay of 1 to 3 minutes.

Create an application monitoring alert

To create an alert for an application monitoring job on Java Virtual Machine-Garbage Collection (JVM-GC) times in corresponding-period comparison, perform the following operations:

1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Alert Policies**.
3. On the **Alert Policies** page, choose **Create Alarm > Application Monitoring Alarm** in the upper-right corner.
4. In the **Create Alarm** dialog box, enter all required information and click **Save**.
   i. Set **Alarm Name**. Example: alert on JVM-GC times in corresponding-period comparison.
   ii. Select an application for **Application Site** and an application group for **Application Group**.
   iii. Select the type of the monitoring metrics from the **Type** drop-down list. Example: **JVM_Monitoring**.
   iv. Set Dimension to **Traverse**.

v. Set Alarm Rules.

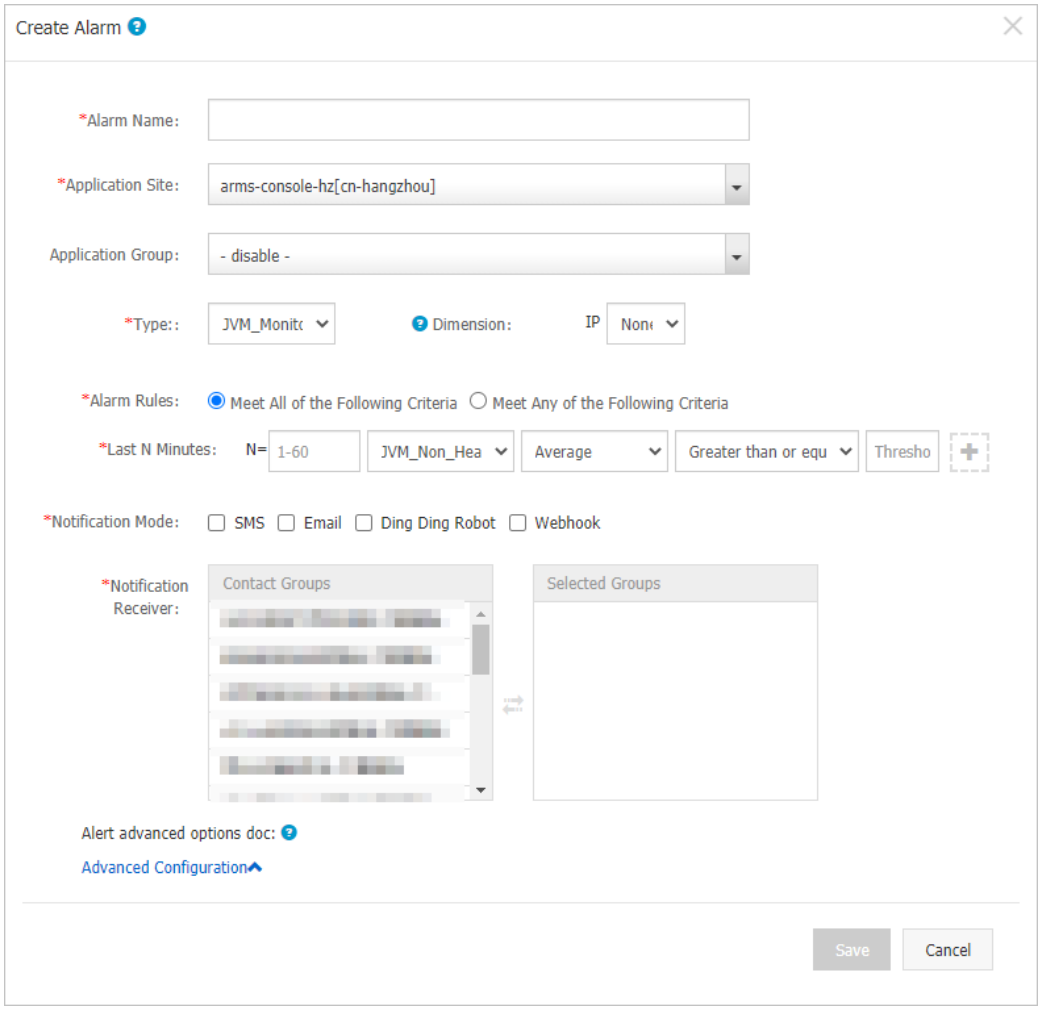
a. Select **Meet All of the Following Criteria**.

b. Edit the alert rule. For example, an alert is triggered when the value of N is 5 and the average value of JVM_FullGC increases by 100% compared with that in the previous hour.

> **Note** To add another alert rule, click the + icon on the right side of **Alarm Rules**.

vi. Set Notification Mode. For example, select Email.

vii. Set Notification Receiver. In the **Contact Groups** section, click the name of a contact group. If the contact group appears in the **Selected Groups** section, the setting is successful.



Create a browser monitoring alert

To create a page metric alert on the JS error rate and JS error count, perform the following operations:

1. In the left-side navigation pane, choose **Alerts > Alert Policies**.
2. On the **Alert Policies** page, choose **Create Alarm > Browser Monitoring Alarm** in the upper-right corner.
3. In the **Create Alarm** dialog box, enter all required information and click **Save**.
   i. Enter Alert Name such as page metric alert.

ii. In the **Application Site** field, select the monitoring job you created.

iii. Select the type of the monitoring metric from the **Type** drop-down list. Example: **Page_Metric**.

iv. Set Dimension to **Traverse**.

v. Set Alarm Rules.

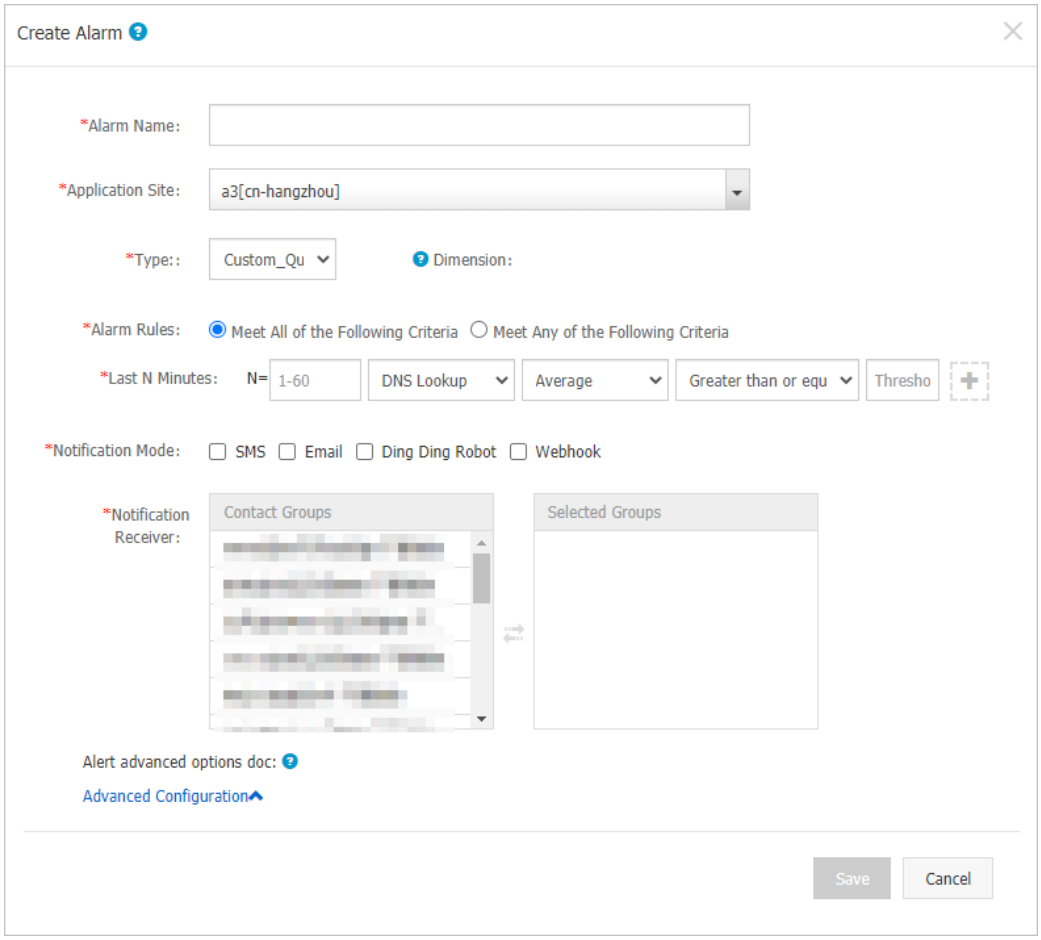
a. Select **Meet All of the Following Criteria**.

b. Edit the alert rule. For example, an alert is triggered when the value of N is 10 and the average value of JS error rate is at least 20.

c. To add another alert rule, click the **+** icon on the right side of Alarm Rules. For example, an alert is triggered when the value of N is 10 and the JS error count is at least 20.

vi. Set Notification Mode. For example, select SMS and Email.

vii. Set Notification Receiver. In the **Contact Groups** section, click the name of a contact group. If the contact group appears in the **Selected Groups** section, the setting is successful.



Create a Prometheus monitoring alert

To create an alert for a Prometheus monitoring job such as an alert on network receiving pressure, perform the following operations:

1. You can select one of the two available methods to go to the Create Alarm page.
   - On the **New DashBoard** page of the Prometheus Grafana dashboard, click theicon to go to the ARMS Prometheus **Create Alarm** dialog box.

- In the left-side navigation pane of the console, choose **Alerts > Alert Policies**. On the **Alert Policies** page, choose **Create Alarm > Prometheus** in the upper-right corner.

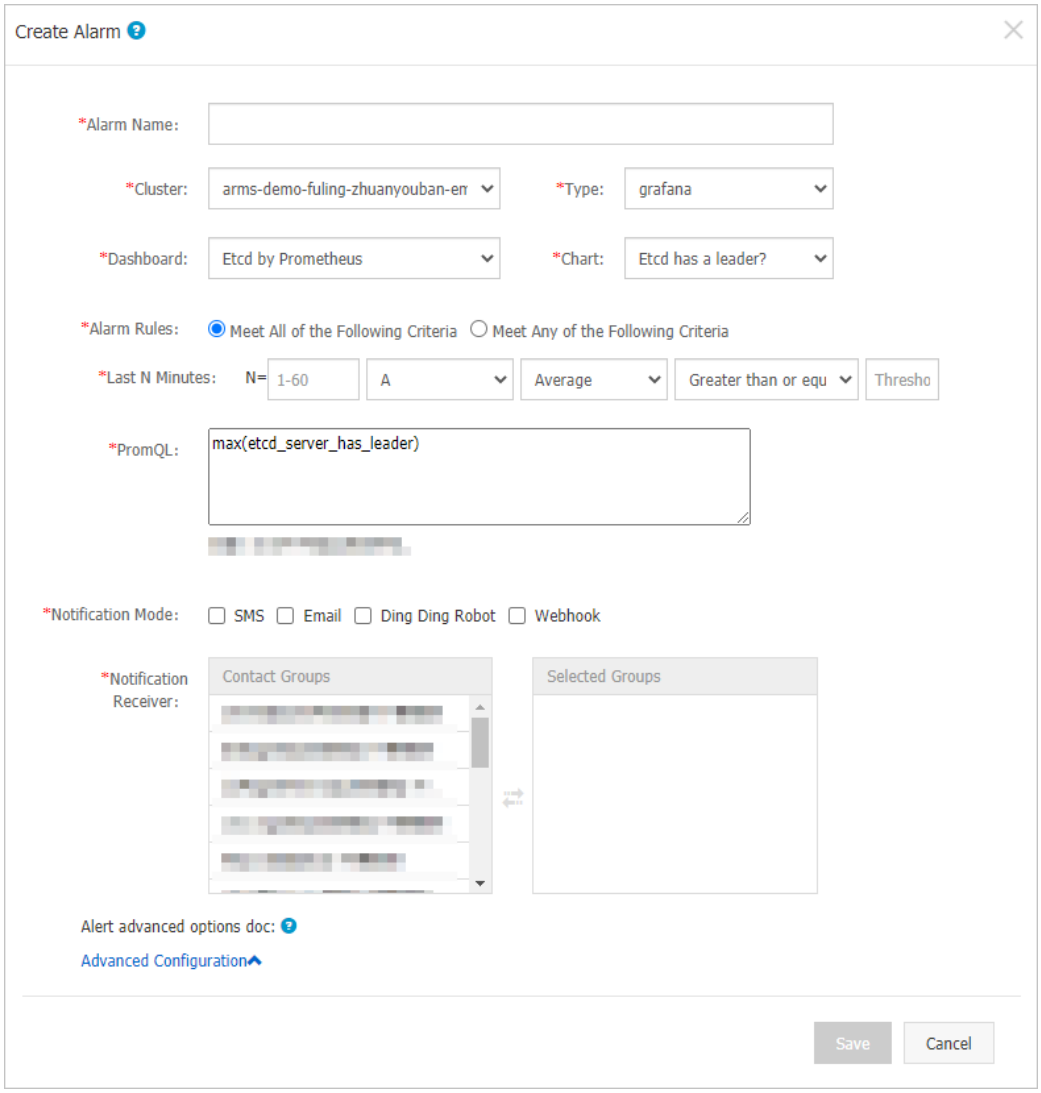
2. In the **Create Alarm** dialog box, enter all required information and click **Save**.
   i. Enter Alarm Name such as network receiving pressure alert.
   ii. Select the corresponding **cluster** of the Prometheus monitoring job.
   iii. Set **Type** to **grafana**.
   iv. Select the specific **dashboard** and **chart** to monitor.
   v. Set Alarm Rules.
      a. Select **Meet All of the Following Criteria**.
      b. Edit the alert rule. For example, an alert is triggered when the value of N is 5 and the average value of network receiving bytes (MB) is at least 3.

         > **Note** A Grafana chart may contain data of Curve A, Curve B, and Curve C. You can select one of them to monitor.

      c. In the **PromQL** field, edit the existing PromQL statement or enter a new PromQL statement.

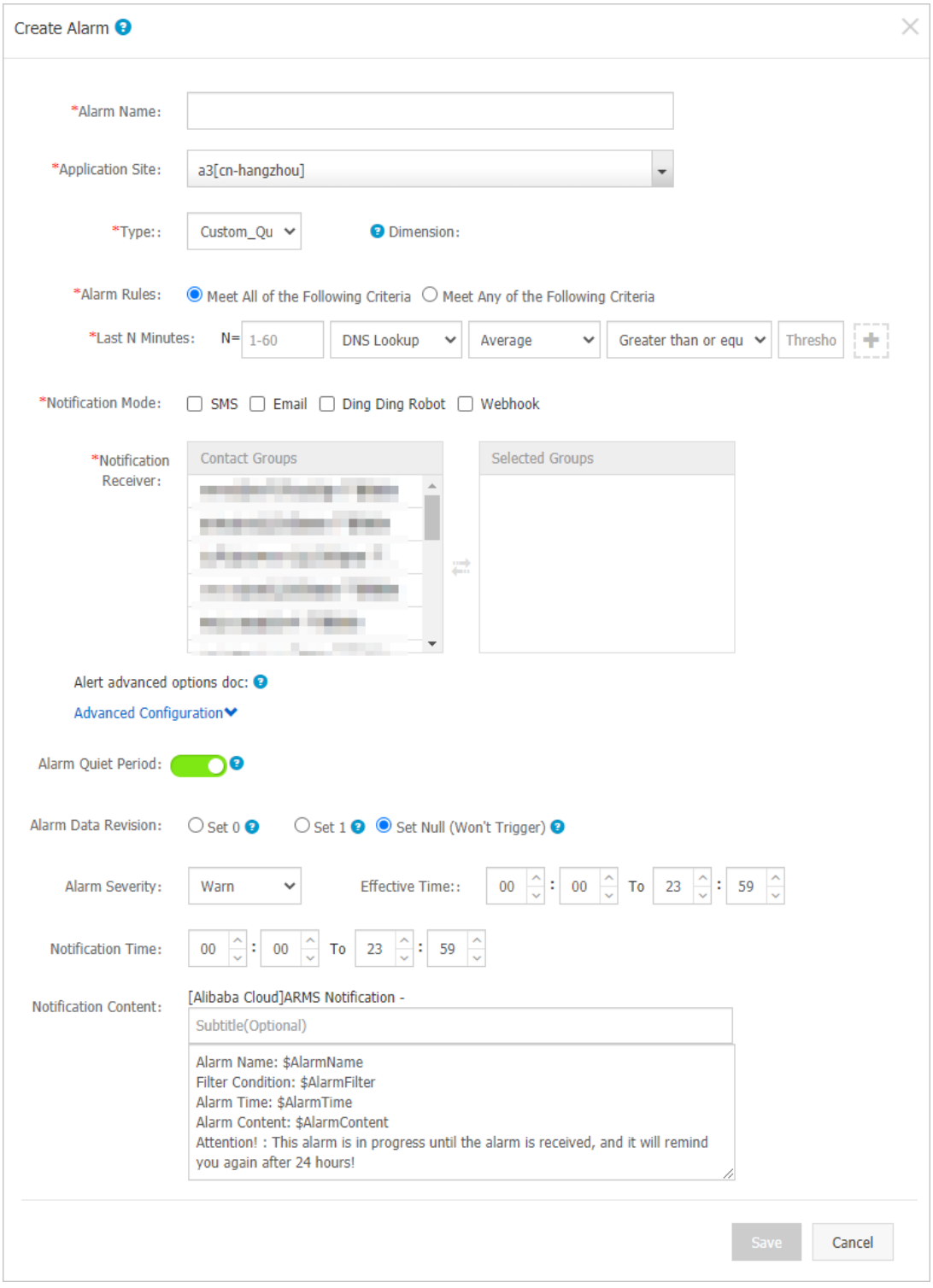
         > **Notice** An error may be reported if a PromQL statement contains a dollar sign ($). You must delete the equal sign (=) and the parameters on both sides of the dollar sign ($) from the statement that contains the dollar sign ($). For example, modify `sum (rate (container_network_receive_bytes_total{instance=~"^$HostIp.*"}[1m]))` to `sum (rate (container_network_receive_bytes_total[1m]))`

   vi. Set Notification Mode. For example, select SMS.
   vii. Set Notification Receiver. In the **Contact Groups** section, click the name of a contact group. If the contact group appears in the **Selected Groups** section, the setting is successful.



Description of basic fields

The following table describes the basic fields of the **Create Alarm** dialog box.



Field	Description	Remarks
Application Site	The monitoring job that has been created.	Select a value from the drop-down list.

Field	Description	Remarks
Type	The type of the metric.	The types for the three alerts are different: • Application monitoring alert: This displays application entry calls, the statistics for application call types, database metrics, JVM monitoring, host monitoring, and abnormal interface calls. • Browser monitoring alert: This shows page metrics, interface metrics, custom metrics, and page interface metrics. • Custom monitoring alert: This creates alerts based on existing drilled-down datasets and existing general datasets.
Dimension	The dimensions for alert metrics (datasets). You can select None, "=", or Traverse.	• When Dimension is set to None, the alert content shows the sum of all values of this dimension. • When Dimension is set to "=", you must enter the specific content. • When Dimension is set to Traverse, the alert content shows the dimension content that actually triggers the alert.
Last N Minutes	The system checks whether the data results in the last N minutes meet the trigger condition.	Valid values of N: 1 to 60.
Notification Mode	Email, SMS, ,Ding Ding Robot, and Webhook are supported.	You can select multiple modes. For more information about how to configure DingTalk robot, see .Enable DingTalk chatbot alert
Alert Quiet Period	You can enable or disable Alert Quiet Period. By default, Alert Quiet Period is enabled.	• When Alert Quiet Period is enabled: if data remains in the triggered state, the second alert notification is sent 24 hours after the first alert is triggered. When data is recovered, you receive a data recovery notification and the alert is cleared. If the data triggers the alert one more time, the alert notification is sent again. • When Alert Quiet Period is disabled: if the alert is continually triggered, the system sends the alert notification every minute.
Alert Severity	Valid values include Warn, Error, and Fatal.	-
Notification Time	The time when the alert was sent. No alert notification is sent out of this time period, but alert events are recorded.	For more information about alert event history, see Manage alerts.

Field	Description	Remarks
Notification Content	The custom content of the alert.	You can edit the default template. In the template, the four variables, $AlertName, $AlertFilter, $AlertTime, and $AlertContent, are preset. (Other preset variables are not supported currently.) The rest of the content can be customized.

Description of complex general fields: period-on-period and period-for-period

- Minute-on-minute comparison: Assume that β is the data (optionally average, sum, maximum, or minimum) in the last N minutes, and α is the data generated between the Nth and 2Nth minute. The minute-on-minute comparison is the percentage increase or decrease when β is compared with α.
- Minute-for-minute hourly comparison: Assume that β is the data (optionally average, sum, maximum or minimum) in the last N minutes, and α is the data generated during the last N minutes in the last hour. The minute-for-minute hourly comparison is the percentage increase or decrease when β is compared with α.
- Minute-for-minute daily comparison: Assume that β is the data (optionally average, sum, maximum or minimum) in the last N minutes, and α is the data generated during the last N minutes at the same time yesterday. The minute-for-minute daily comparison is the percentage increase or decrease when β is compared with α.

Description of complex general fields: Alert Data Revision Strategy

You can select "Zero fill", "One fill", or "Zero fill null" (default). This feature is generally used to fix anomalies in data, including no data, abnormal composite metrics, and abnormal period-on-period and period-for-period comparisons.

- Zero fill: fixes the value checked to 0.
- One fill: fixes the value checked to 1.
- Zero fill null: does not trigger the alert.

Scenarios:

- Anomaly 1: no data
  User A wants to use the alert feature to monitor the page views. When User A creates the alert, User A selects Browser Monitoring Alert. User A sets the alert rule: N is 5 and the sum of the page views is at most 10. If the page is not accessed, no data is reported and no alert is sent. To solve this problem, you can select "Zero fill" as the alert data revision policy. If you do not receive any data, it considered that zero data is received. This meets the alert rule and an alert is sent.
- Anomaly 2: abnormal composite metrics
  User B wants to use the alert feature to monitor the real-time unit price of a product. When User B creates the alert, User B selects Custom Monitoring Alert. User B sets the dataset of variable a to the current total price, and the dataset of variable b to the current total items. User B also sets the alert rule that N is 3 and the minimum value of current total price divided by current total items is at most 10. If the current total of items is 0, the value of the composite metric, current total price divided by current total items, does not exist. No alert is sent. To solve this problem, you can select "Zero fill" as the alert data revision policy. The value of the composite metric, current total price divided by current total items, is now considered to be 0. This meets the alert rule and an alert is sent.
- Anomaly 3: abnormal period-on-period and period-for-period comparisons

User C wants to use the alert function to monitor the CPU utilization of the node machine. When User C creates the alert, User C selects Application Monitoring Alert, and sets the alert rule: N is 3 and the average user CPU utilization of the node machine decreases by 100% compared with the previous monitoring period. If the CPU of the user fails to work in the last N minutes, α cannot be obtained. This means the period-on-period result does not exist. No alert is sent. To solve this problem, you can select the alert data revision strategy as "One fill", and consider the period-on-period comparison result as a decrease of 100%. This meets the alert rule and an alert is sent.

What's next

You can query and delete alert records in alert management.

5.Manage alerts

On the Alert Policies page, you can manage all the alert rules within your Alibaba Cloud account and query the history of alert events and alert notifications.

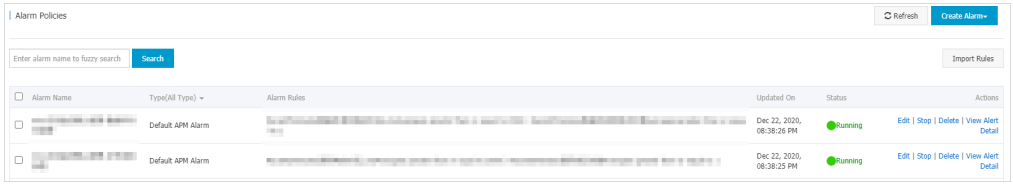
Manage alert rules

On the **Alert Policies** page, the alert rules that you created in Application Monitoring, Browser Monitoring, and Custom Monitoring are displayed. You can start, stop, edit, and delete the alert rules. You can also view alert details. For more information about how to create alert rules, see Create ARMS alerts.

1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Alert Policies**.
3. (Optional)On the **Alert Policies** page, enter the alert name in the search box and click **Search**.

   > **Note** You can enter part of an alert name in the search box to perform a fuzzy search.

4. You can perform the following operations on an alert rule in the **Actions** column based on your business requirements:
   - To edit an alert rule, click **Edit** in the Actions column. In the **Edit Alarm** dialog box, edit the alert rule and click **Save**.
   - To delete an alert rule, click **Delete** in the Actions column. In the **Delete** dialog box, click **Delete**.
   - To start a stopped alert rule, click **Start** in the Actions column. In the **OK** dialog box, click **Start**.
   - To stop a running alert rule, click **Stop** in the Actions column. In the **Stop** dialog box, click **OK**.
   - To view the alert event history and alert sending history, click the **Alert History** tab. You can then click the **Alert Event History** and **Alarm Post History** tabs to view the history.



Query the alert history

On the **Alert History** tab, you can view historical records that indicate when and why an alert rule was triggered. You can also view historical records about the alert notifications sent to specified alert contacts.

1. In the left-side navigation pane, choose **Alerts > Alert Policies**. On the Alert Policies page, click the Alert History tab.
2. On the **Alert History** tab, select or enter **Type**, **Trigger State**, and **Alert Name**, and then click **Search**.
3. On the **Alert Policies** page, you can view historical records of alert events.

   > **Note** Alert notifications are sent only if the alert rule is in the **Triggered** state. In this state, a red dot is displayed in the **Trigger** column.

4. Click the **Alarm Post History** tab to view the history of alert notifications that were sent for triggered alerts. The alert notifications include SMS messages and emails.

Related information

- Create ARMS alerts
- Create contacts
- Create a contact group

6.Manage alert templates

Application Real-Time Monitoring Service (ARMS) provides alert templates that allow you to create alerts in batches, improving efficiency in configuring alert rules.

Context

ARMS comes with the following alert templates:

- Application monitoring
  - DefaultAPM-DB-Alert: This template is used to generate alerts for long database response time and database call errors.
  - DefaultAPM-Exception-Alert: This template is used to generate alerts for call timeout and call errors.
  - DefaultAPM-Host-Alert: This template is used to generate alerts for high CPU usage and insufficient disk space.
  - DefaultAPM-Process-Alert: This template is used to generate alerts for process status.
  - DefaultAPM-GC-Alert: This template is used to generate alerts for excessive full garbage collection (GC) events, long full GC duration, and long young GC duration.
- Browser monitoring
  - DefaultRetcodeAlert: This template is used to generate alerts for a high JavaScript (JS) error rate and excessive JS errors.

Create alert templates

In addition to the default alert templates provided by ARMS, you can create custom alert templates as needed. ARMS allows you to create only two types of alert templates: browser monitoring and application monitoring.

1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Alert Template Management**.
3. On the **Alert Template Management** page, click **Create Alert Template** in the upper-right corner.
   - Click **Browser Monitoring Alert Template**. In the **Create Alert Template** dialog box, set all required parameters, and then click **Save**. For more information about the fields, see .
   - Click **Application Monitoring Alert Template**. In the **Create Alert Template** dialog box, set all required parameters, and then click **Save**. For more information about the fields, see .
4. (Optional)Select the alert template you created in the alert template list. In the **Actions** column, click **Create Alert**. In the **Create Alert** dialog box, set all required parameters and click **Save**. Choose **Alerts > Alert Policies**. On the **Alert Policies** page, click the **Alert Rules** tab. The alert rule you created appears in the alert list, indicating that you have created the alert rule by using the alert template you created.
5. (Optional)Select the alert template you created in the alert template list. In the **Actions** column, click **Batch Create Alerts**.
6. (Optional)In the **Batch Create Alerts** dialog box, click multiple applications in the **Unselected** section to add them to the **Selected** section. Click **Save**. In the **Note** dialog box, click **OK**. Choose **Alerts > Alert Policies**. On the **Alert Policies** page, click the **Alert Rules** tab. The alert rules you created in batches appear in the alert list, indicating that you have created the alert rules in batches by using the alert template you created.

Manage alert templates

You can enable or disable the Auto-Generation feature of an alert template, as well as edit, delete, and copy the alert template.

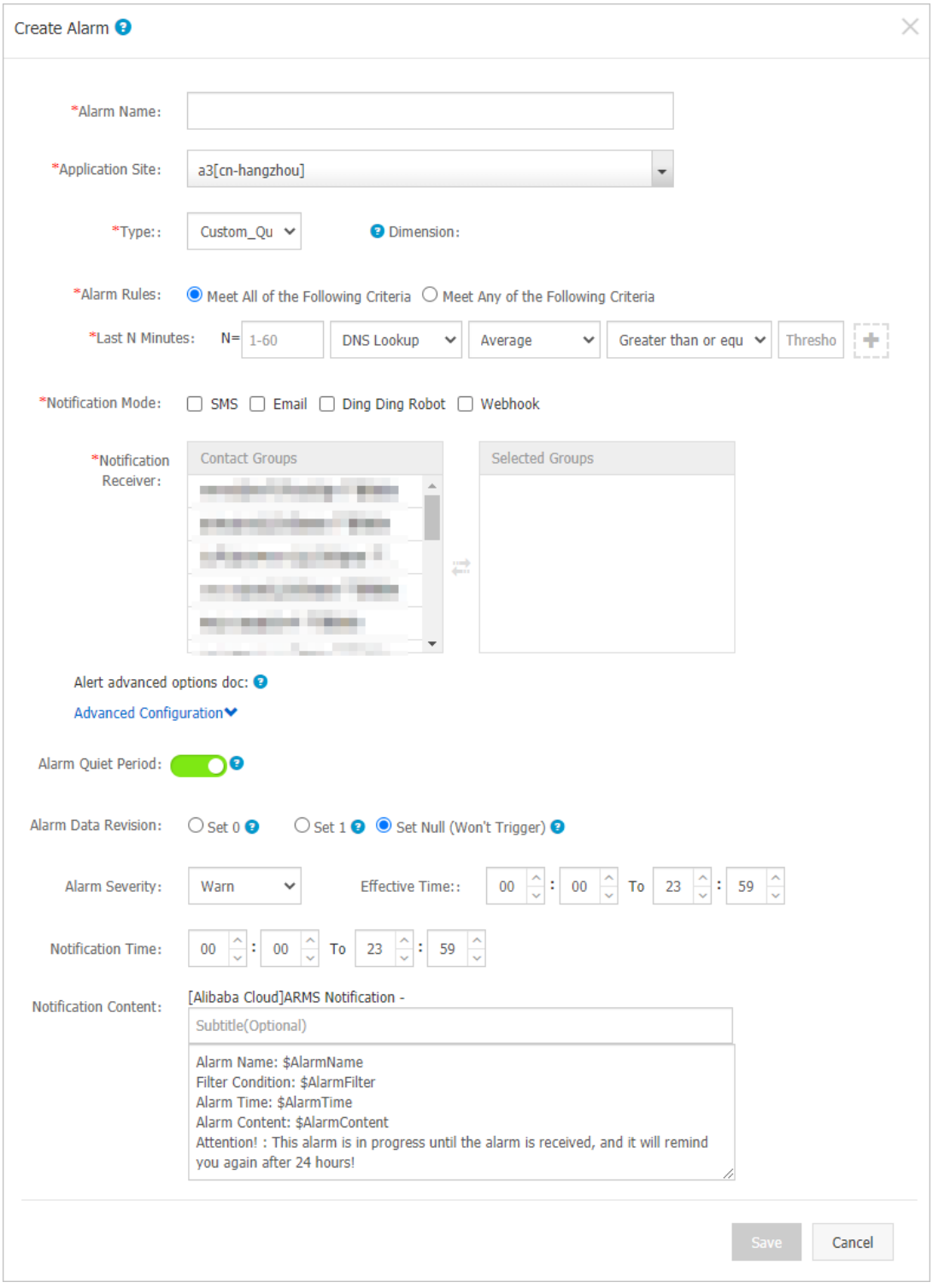
1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Alert Template Management**.
3. Find the target alert template in the alert template list, and click the buttons in the **Actions** column as needed.
   - To automatically create alert rules for a new application, click **Enable Auto-Generation**. In the **Stop** dialog box, click **OK**. If you do not need to automatically create alert rules for the new application, click **Disable Auto-Generation**. In the **Stop** dialog box, click **OK**.

     > **Note** For a newly created alert template, the Auto-Generation feature is enabled by default.

   - To edit an alert template, click **Edit**. In the **Edit Alert Template** dialog box, edit the alert template, and click **Save**.
   - To delete an alert template, click **Delete**. In the **Delete** dialog box, click **Delete**.
   - To copy an alert template, click **Copy**. In the **Edit Alert Template** dialog box, edit the alert template, and click **Save**.

Description of basic fields

The following table describes the basic fields in the **create alarm** dialog box.



Field	Description	Remarks
Application Site	The monitoring job that has been created.	Select a value from the drop-down list.

Field	Description	Remarks
Type	The type of the metric.	The types for the three alerts are different: • Application monitoring alert: This displays application entry calls, the statistics for application call types, database metrics, JVM monitoring, host monitoring, and abnormal interface calls. • Browser monitoring alert: This shows page metrics, interface metrics, custom metrics, and page interface metrics. • Custom monitoring alert: This creates alerts based on existing drilled-down datasets and existing general datasets.
Dimension	The dimensions for alert metrics (datasets). You can select None, "=", or Traverse.	• When Dimension is set to None, the alert content shows the sum of all values of this dimension. • When Dimension is set to "=", you must enter the specific content. • When Dimension is set to Traverse, the alert content shows the dimension content that actually triggers the alert.
Last N Minutes	The system checks whether the data results in the last N minutes meet the trigger condition.	Valid values of N: 1 to 60.
Notification Mode	Email, SMS, ,Ding Ding Robot, and Webhook are supported.	You can select multiple modes. If you need to set a DingTalk robot alarm see . Set DingTalk robot alert
Alert Quiet Period	You can enable or disable Alert Quiet Period. By default, Alert Quiet Period is enabled.	• When Alert Quiet Period is enabled: if data remains in the triggered state, the second alert notification is sent 24 hours after the first alert is triggered. When data is recovered, you receive a data recovery notification and the alert is cleared. If the data triggers the alert one more time, the alert notification is sent again. • When Alert Quiet Period is disabled: if the alert is continually triggered, the system sends the alert notification every minute.
Alert Severity	Valid values include Warn, Error, and Fatal.	-
Notification Time	The time when the alert was sent. No alert notification is sent out of this time period, but alert events are recorded.	For more information about viewing alert event records, see . Manage alarms .

Field	Description	Remarks
Notification Content	The custom content of the alert.	You can edit the default template. In the template, the four variables, $AlertName, $AlertFilter, $AlertTime, and $AlertContent, are preset. (Other preset variables are not supported currently.) The rest of the content can be customized.

Related information

- Create an alert
- Create common alert rules

7.Create contacts

When an alert rule is triggered, notifications are sent to the contact group that you specified. Before you create a contact group, you must create contacts. When you create a contact, you can specify the mobile phone number and email address of the contact to receive notifications. You can also provide a DingTalk chatbot webhook URL used to automatically send alert notifications.

Prerequisites

To add a DingTalk chatbot as a contact, you must obtain its webhook URL first. For more information, see Configure a DingTalk chatbot to send alert notifications.

Procedure

1. Log on to the ARMS console .
2. In the left-side navigation pane of the console, choose **Alerts > Contacts**.
3. On the **Contacts** tab, click **New contact** in the upper-right corner.
4. In the **New contact** dialog box, edit the contact information, specify whether to receive system notifications, and click **OK**.
   - To add a contact, specify the **Name**, **Mobile phone number** and **Mailbox** fields.

     > **Note** You must specify one of the Mobile phone number and Mailbox parameters. Each phone number or email address must be used for only one contact. You can create a maximum of 100 contacts.

   - To add a DingTalk chatbot, enter the name and the webhook URL of the chatbot.

     > **Note** For more information about how to obtain the webhook URL of the DingTalk chatbot, see Configure a DingTalk chatbot to send alert notifications.

Subsequent operations

- To search for contacts, on the **Contacts** tab, select **Name**, **Cell phone number**, or **Email** in the drop-down list, then enter the entire or a part of the selected name, phone number or email in the search box, and click the icon.
- To edit a contact, click **Editing** in the **Actions** column of the contact, edit the information in the **Edit contacts** dialog box, then click **OK**.
- To delete a single contact, click **Delete** in the **Actions** column of the contact, then click **OK** in the message.
- To delete multiple contacts, select the contacts, click **Batch Delete Contacts**, then click **OK** in the dialog box.

Related information

- Create a contact group
- Configure a DingTalk chatbot to send alert notifications
- Create ARMS alerts
- Manage alerts

8.Create a contact group

When you create an alert rule, you can specify a contact group as the receiver of alert notifications. If the alert rule is triggered, Application Real-Time Monitoring Service (ARMS) sends alert notifications to the contacts in the contact group. This topic describes how to create a contact group.

Prerequisites

. For more information, see Create contacts.

Procedure

1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Contacts**.
3. On the **Contact Group** tab, click **Create a contact group** in the upper-right corner.
4. In the **Create a contact group** dialog box, enter a group name in the **Group Name** field, select alert contacts in the **Alarm contact** list, and then click **OK**.

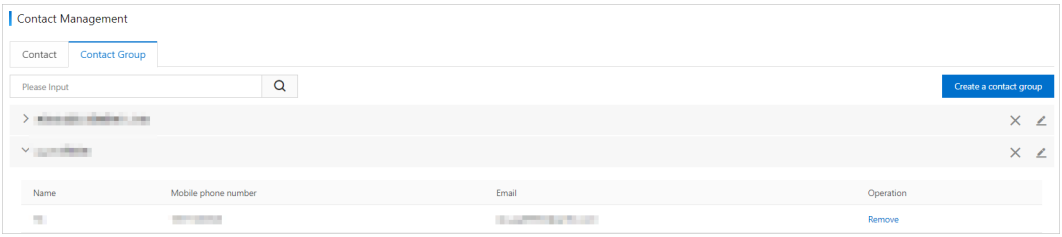
> **Note** If no alert contacts are displayed in the **Alarm contact** list, you must first create an alert contact. For more information, see Create contacts.

What to do next

- To search for a contact group, go to the **Contact Group** tab, enter the contact group name or keywords of the name in the search box, and then click the icon.

> **Notice** The search is case-sensitive.

- To edit a contact group, click the icon to the right of the contact group. In the **Edit Contact Group** dialog box, edit the contact group and click **OK**.
- To view the contacts in a contact group, click the icon to the left of the contact group to show the group.



> **Note** You can remove one or more contacts from a contact group in shown mode. To remove a contact, find the alert contact and click **Remove** in the **Operation** column.

- To delete a contact group, click the icon to the right of the contact group. In the dialog box that appears, click **OK**.

> **Notice** Before you delete a contact group, make sure that no monitoring tasks to which the contact group is attached are running. Otherwise, the alerting feature cannot function as expected.

Related information

- Create contacts
- Configure a DingTalk chatbot to send alert notifications
- Create ARMS alerts
- Manage alerts

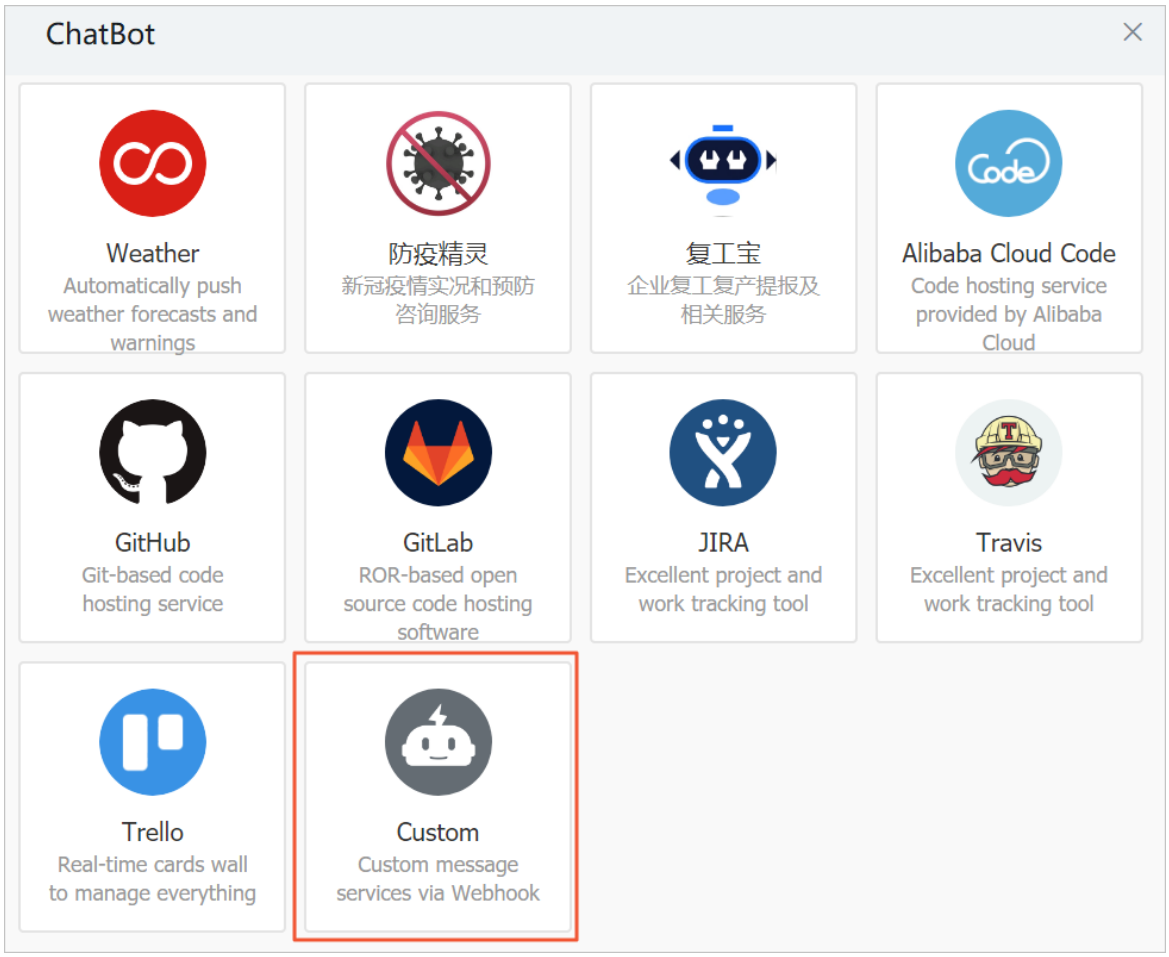
9.Configure a DingTalk chatbot to send alert notifications

After you configure a DingTalk chatbot to send notifications, you can specify DingTalk groups to receive alert notifications. This topic describes how to configure a DingTalk chatbot to send notifications.

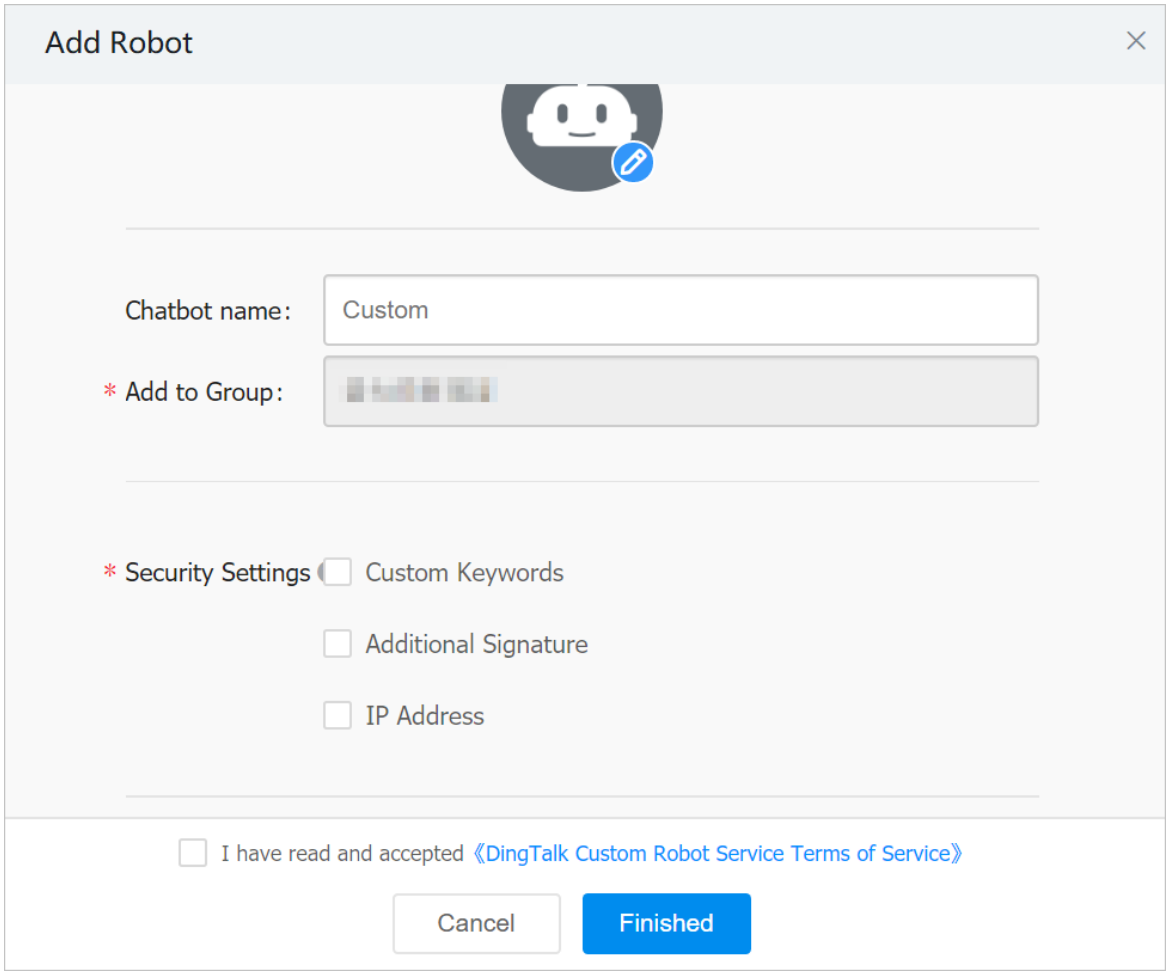
Add a custom DingTalk chatbot and obtain the webhook URL

Perform the following steps to add a custom DingTalk chatbot and obtain the webhook URL:

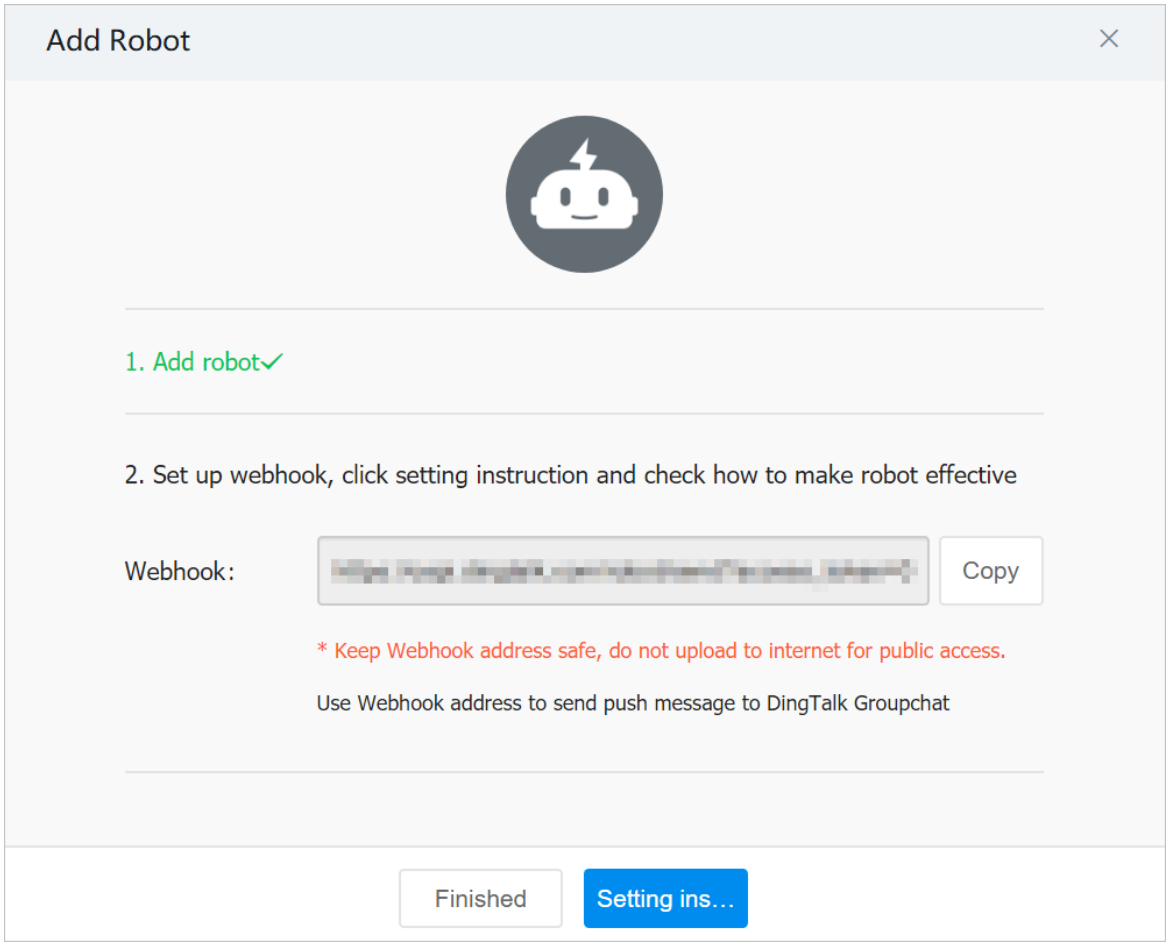
1. Run the DingTalk client on a PC, go to the DingTalk group to which you want to add an alert chatbot, and then click the Group Settings icon in the upper-right corner.
2. In the **Group Settings** panel, click **Group Assistant**.
3. In the **Group Assistant** panel, click **Add Robot**.
4. In the **ChatBot** dialog box, click the **+** icon in the **Add Robot** card. Then, click **Custom**.



5. In the **Robot details** dialog box, click **Add**.
6. In the **Add Robot** dialog box, edit the profile picture, enter a name, and then select at least one of the options in the **Security Settings** section. Read and select **I have read and accepted DingTalk Custom Robot Service Terms of Service**. Click **Finished**.



7. In the **Add Robot** dialog box, copy the webhook URL that the system generates for the chatbot.



Create a contact

You can create a contact or add the address to the existing contact information. In this example, a contact is created.

1. Log on to the Application Real-Time Monitoring Service (ARMS) console.
2. In the left-side navigation pane, choose **Alerts > Contacts**.
3. On the **Contact** tab, click **New contact** in the upper-right corner.
4. In the **New contact** dialog box, enter the Webhook URL obtained in Add a custom DingTalk chatbot and obtain the webhook URL. Then, click **OK**.

Create a contact group

You must create a contact group, because alerts triggered by an alert rule can be sent to alert contact groups instead of alert contacts.

1. Log on to the ARMS console.
2. In the left-side navigation pane, choose **Alerts > Contacts**.
3. On the **Contact Group** tab, click **Create a contact group** in the upper-right corner.
4. In the **Create a contact group** dialog box, set **Group name**, enter the contact created in Create a contact in the **Alarm contact** field, and then click **OK**.

Create an alert

If no alert is created, create an alert first. You can configure DingTalk chatbots to send notifications for browser monitoring, application monitoring, custom monitoring, and Prometheus monitoring alerts. In this example, an application monitoring alert is created.

1. Log on to the ARMS console.
2. In the left-side navigation pane, choose **Alerts > Alert Policies**.
3. On the **Alarm Policies** page, choose **Create Alarm > Application Monitoring Alarm** in the upper-right corner.
4. In the **Create Alarm** dialog box, set the parameters and click **Save**.
   i. Set **Alarm Name**. For example, you can enter alert on JVM-GC times in period-over-period comparison.
   ii. From the **Application Site** drop-down list, select an application.
   iii. From the **Type** drop-down list, select the type of metric that you want to monitor. For example, you can select **JVM_Monitoring**.
   iv. Set **Dimension** to **Traverse**.
   v. Set Alarm Rules.
      a. Select **Meet All of the Following Criteria**.
      b. Create an alert rule. For example, an alert is triggered when the value of N is 5 and the average value of JVM_FullGC increases by 100% compared with that in the previous hour.

      > **Note** To add another alert rule, click the **+** icon next to **Last N Minutes**.

   vi. In the **Notification Mode** section, select **Ding Ding Robot**.
   vii. Set Notification Receiver to the contact group created in Create a contact group. In the **Contact Groups** list, click the name of the contact group. If the contact group appears in the **Selected Groups** list, the setting is successful.

Edit an alert

Perform the following steps to edit an alert. In this example, an application monitoring alert is edited.

1. Log on to the ARMS console.
2. In the left-side navigation pane, choose **Alerts > Alert Policies**.
3. On the page that appears, find the alert that you want to edit, and click **Edit** in the **Actions** column.
4. In the **Edit Alarm** dialog box, modify related parameters and click **Save**.
   i. Change the value of Notification Mode to **Ding Ding Robot**.
   ii. Change the value of Notification Receiver to the contact group created in Create a contact group. In the **Contact Groups** list, click the name of the contact group. If the contact group appears in the **Selected Groups** list, the setting is successful.

Result

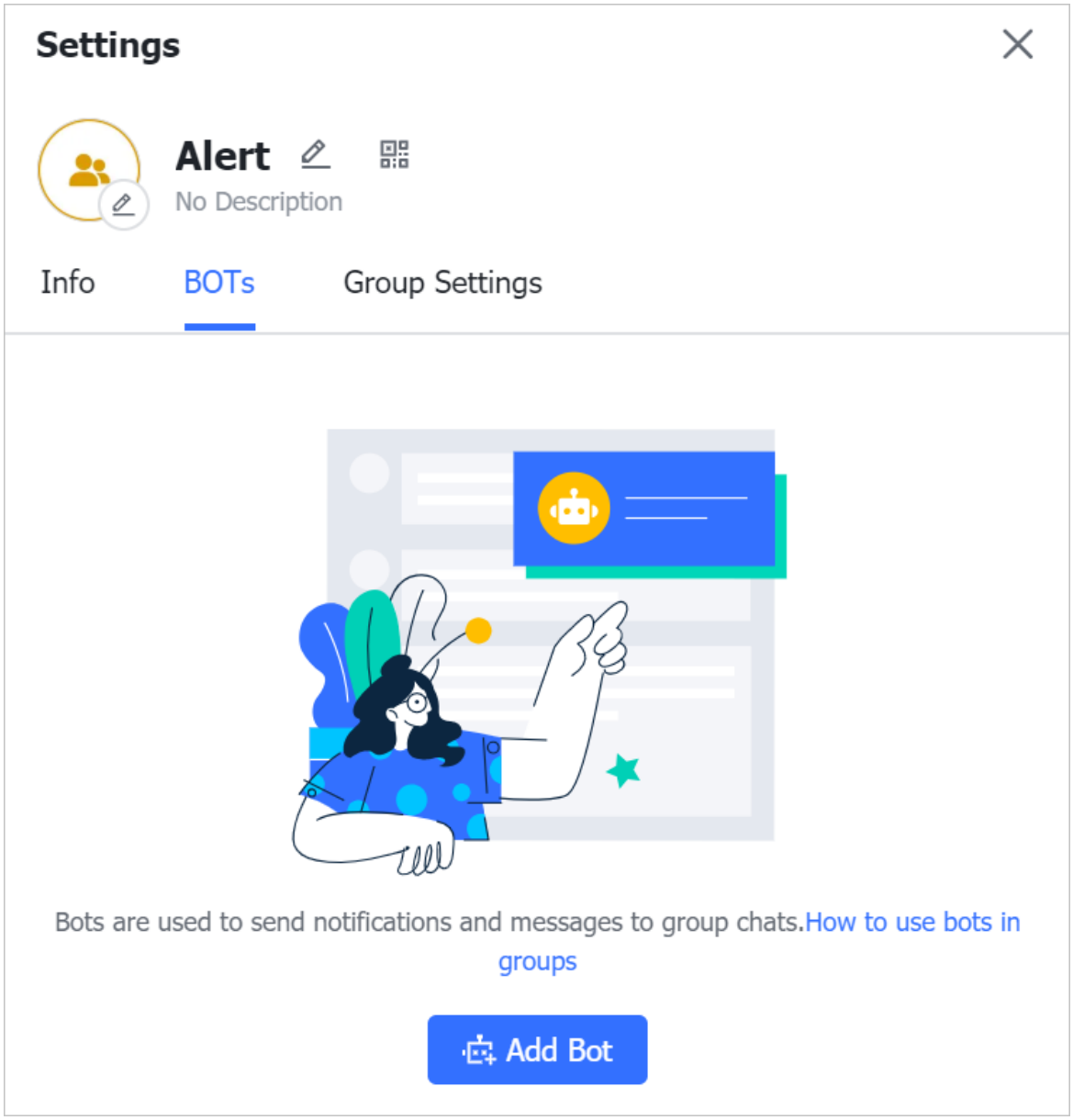
At this point, a DingTalk chatbot is configured to send notifications. When an alert is triggered, you will receive an alert notification from the specified DingTalk group.

10.Create webhook alerts

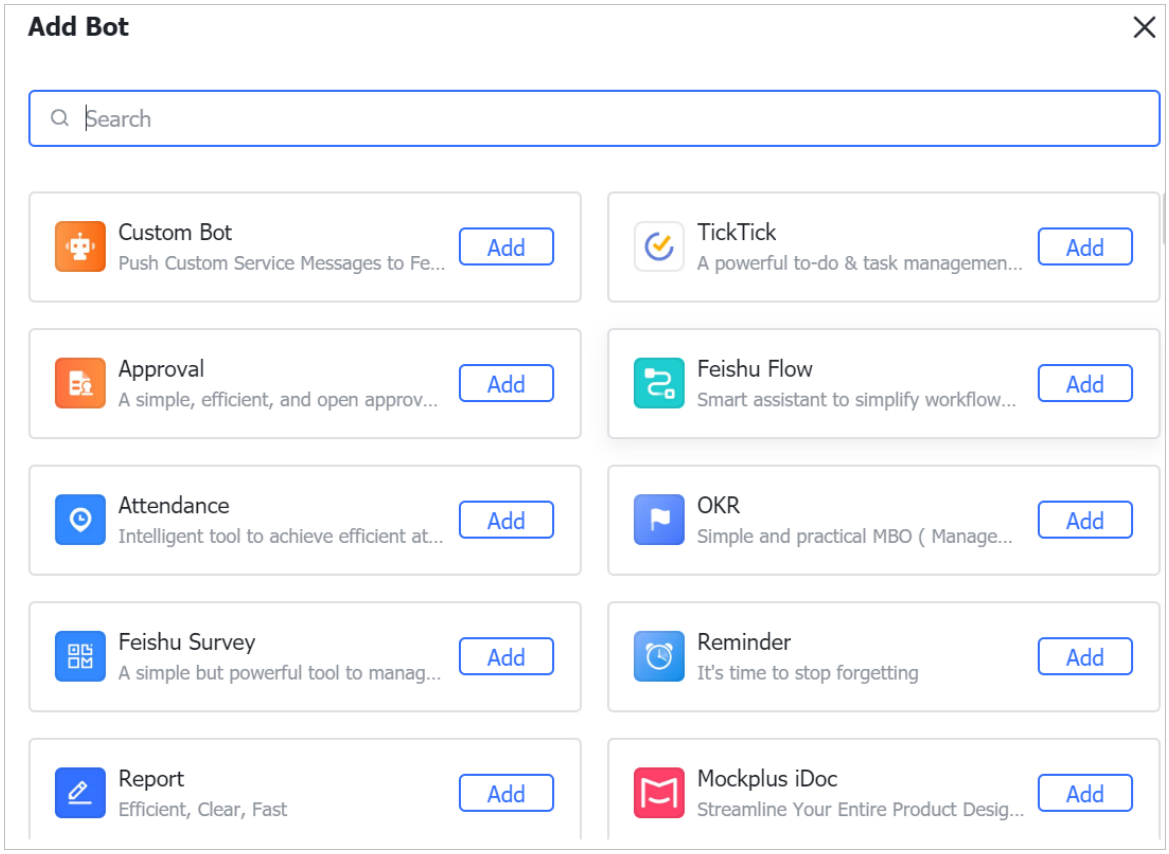
After you configure a webhook alert, you can send alert notifications to a specified webhook URL. Prometheus can send Webhook alert notifications to applications such as Feishu, WeChat, and DingTalk. This topic describes how to create a webhook alert. Feishu is used in this example.

Step 1: Create a webhook URL

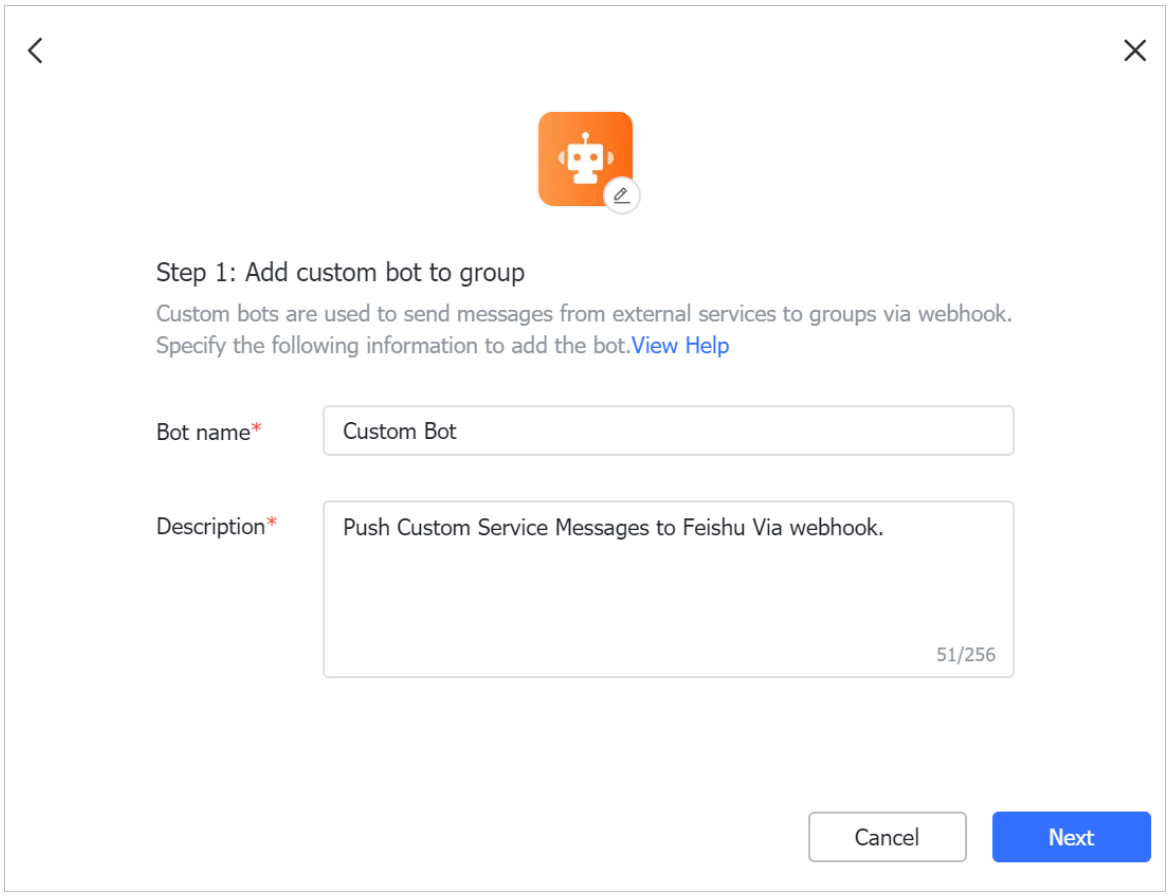
1. Open and log on to Feishu.
2. Click the + icon, and then click **Add Group** to create a Feishu group where alert notifications are sent.
3. Click the group settings icon, and then click the **Bots** tab.
4. On the **BOTs** tab, click **Add Bot**.



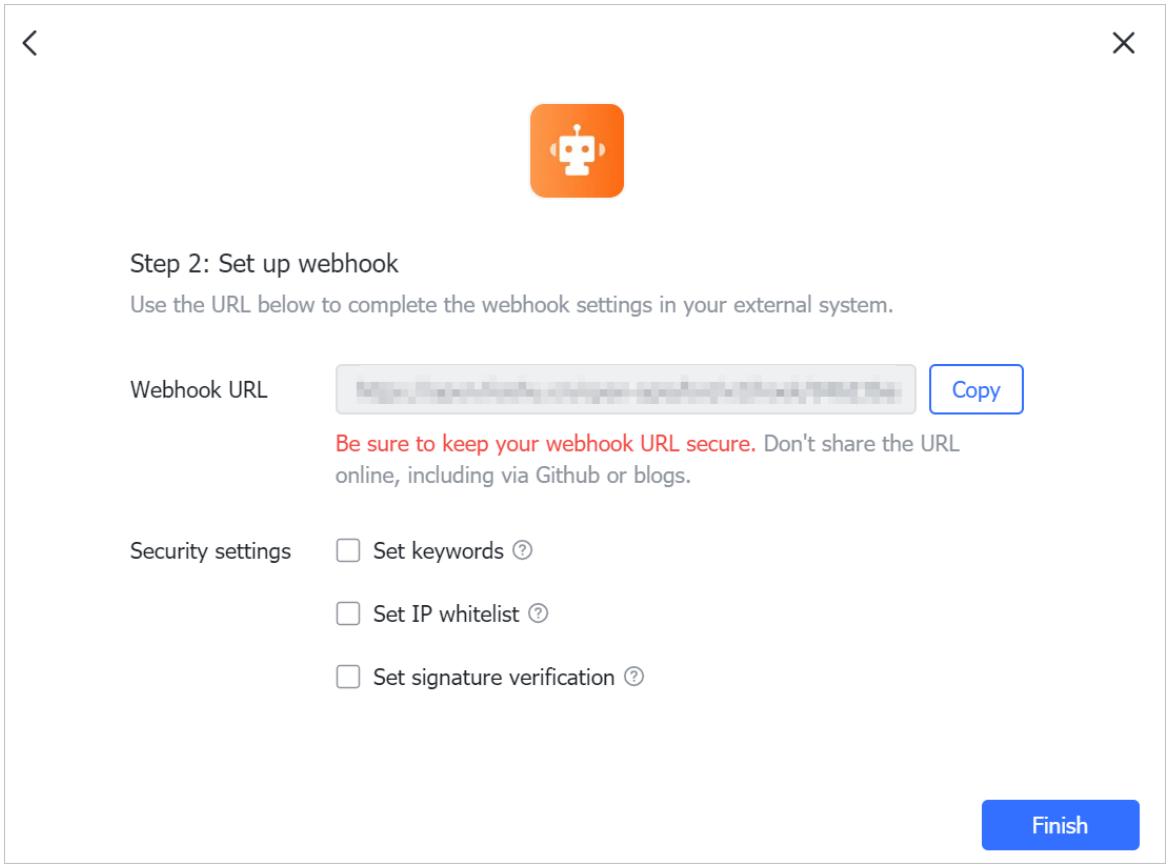
5. In the **Add Bot** dialog box, select **Custom Bot**.



6. Set the Bot name and Description parameters, and then click **Next**.



7. Click **Copy** to save the value of the Webhook URL parameter, and then click **Finish**.



Step 2: Create a webhook alert

1. Log on to the ARMS console.
2. In the left-side navigation pane, choose **Alerts > Contacts**.
3. On the **Contact** tab, click **Create a webhook** in the upper-right corner.
4. In the **Create Webhook** dialog box, set the parameters.
   The following table describes the parameters.

Parameter	Description
Webhook Name	Required. The name of the custom webhook.
Post or Get	Required. The request method. The requested URL cannot exceed 100 characters in length. In this example, select Post and paste the webhook URL that is saved in the Step 1: Create a webhook URL section to the field.

Parameter	Description
Header and Param	Optional. The request header. The header cannot exceed 200 characters in length. You can click **+ Add** to add headers or parameters. The default request header is Content-Type: text/plain; charset=UTF-8. The total number of headers and parameters cannot exceed 6. In this example, set the following two headers: ◦ Arms-Content-Type : json ◦ Content-Type : application/json
Body	Optional. This parameter is available if you use the post method. You can use $content in the body string as a placeholder of alert content. The body cannot exceed 500 characters in length. In this example, use the following alert content format: `{"msg_type": "text","content": {"text": "$content"}}`

5. (Optional)Click **Test** to verify whether the configurations are valid.
6. Click **Create**.

11.Tutorials

11.1. Create common alert rules

Application Real-Time Monitoring Service (ARMS) provides multiple alert rule configuration templates for typical scenarios, such as application monitoring alert scenarios and browser monitoring alert scenarios. You can use the templates to create frequently used alert rules.

Context

ARMS comes with the following alert templates:

- Application monitoring
  - DefaultAPM-DB-Alert: This template is used to generate alerts for long database response time and database call errors.
  - DefaultAPM-Exception-Alert: This template is used to generate alerts for call timeout and call errors.
  - DefaultAPM-Host-Alert: This template is used to generate alerts for high CPU usage and insufficient disk space.
  - DefaultAPM-Process-Alert: This template is used to generate alerts for process status.
  - DefaultAPM-GC-Alert: This template is used to generate alerts for excessive full garbage collection (GC) events, long full GC duration, and long young GC duration.
- Browser monitoring
  - DefaultRetcodeAlert: This template is used to generate alerts for a high JavaScript (JS) error rate and excessive JS errors.

Procedure

1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Alert Template Management** to go to the **Alert Template Management** page.

Create alert rules for application monitoring by using the DefaultAPM-DB-Alert template

To create an alert rule for determining whether database exceptions occur, find the **DefaultAPM-DB-Alert** template and click **Create Alert** in the **Actions** column.

As shown in the preceding figure, this alert rule uses two metrics as alert triggers. An alert event is generated when one of the following criteria is met:

- Long database response time: The response time of the call to the application database is greater than or equal to 2 seconds per minute on average within 5 minutes. Set Type to **Database_Metric** and the metric to **DB_RT_ms**. The rule is that the average response time in the last 5 minutes is greater than or equal to 2000.
- Database call error: The average number of errors that occur when the application calls a database is greater than or equal to once per minute during 5 minutes. Set Type to **Database_Metric** and the metric to **DB_ErrorCount**. The rule is that the average number of errors in the last 5 minutes is greater than or equal to 1.

Create alert rules for application monitoring by using the DefaultAPM-Exception-Alert template

To create an alert rule for determining whether application call exceptions occur, find the **DefaultAPM-Exception-Alert** template and click **Create Alert** in the **Actions** column.

As shown in the preceding figure, this alert rule uses two metrics as alert triggers. An alert event is generated when one of the following criteria is met:

- Call timeout: The average time of inbound API calls of the application is greater than or equal to 2 seconds per minute during 5 minutes. Set Type to **Invocation_Statistic** and the metric to **Invocation_RT_ms**. The rule is that the average time in the last 5 minutes is greater than or equal to 2000.
- Call error: The number of errors that occur during inbound API calls of the application is greater than or equal to once per minute during 5 minutes. Set Type to **Invocation_Statistic**, and the metric to **Invocation_ErrorCount**. The rule is that the average number of call errors in the last 5 minutes is greater than or equal to 1.

Create alert rules for application monitoring by using the DefaultAPM-Host-Alert template

To create an alert rule for determining whether any exceptions occur to the node on which an application is installed, find the **DefaultAPM-Host-Alert** template and click **Create Alert** in the **Actions** column.

As shown in the preceding figure, this alert rule uses two metrics as alert triggers. An alert event is generated when one of the following criteria is met:

- High CPU usage: The CPU usage of the node on which the application is installed is greater than or equal to 90% per minute during 5 minutes. Set Type to **Host_Monitoring**, and the metric to **Node_User_CPU_percent**. The rule is that the average value in the last 5 minutes is greater than or equal to 90.
- Insufficient disk space: The average free disk space of the node on which the application is installed is less than or equal to 1 MB per minute during 5 minutes. Set Type to **Host_Monitoring**, and the metric to **Node_Disk_Free_byte**. The rule is that the average free disk space in the last 5 minutes is less than or equal to 1048576, that is, 1 MB.

Create alert rules for application monitoring by using the DefaultAPM-Process-Alert template

To create an alert rule for determining whether process exceptions occur, find the **DefaultAPM-Process-Alert** template and click **Create Alert** in the **Actions** column.

As shown in the preceding figure, this alert rule uses one metric as the alert trigger. An alert event is generated when the following criteria is met:

- Process status: A process exception occurs. Set Type to **JVM_Monitoring**, and the metric to **JVM_ThreadCount**. The rule is that the average value in the last 1 minute drops by more than 50% from the previous hour.

Create alert rules for application monitoring by using the DefaultAPM-GC-Alert template

To create an alert rule for determining whether garbage collection (GC) exceptions occur, find the **DefaultAPM-GC-Alert** template and click **Create Alert** in the **Actions** column.

As shown in the preceding figure, this alert rule uses three metrics as alert triggers. An alert event is generated when all the following three criteria are met:

- Excessive full GC events: The average number of full GC events is greater than or equal to twice per minute during 10 minutes. Set Type to **JVM_Monitoring** and the metric to **JVM_FullGC_Count**. The rule is that the average value in the last 10 minutes is greater than or equal to 2.
- Excessive full GC time consumption: The average time required for full GC events is greater than or equal to 10 seconds per minute during 10 minutes. Set Type to **JVM_Monitoring** and the metric to **JVM_FullGC_Time_ms**. The rule is that the average value in the last 10 minutes is greater than or equal to 10000, that is, 10 seconds.
- Excessive young GC time consumption: The total time required by young GC events of the application is greater than or equal to 5 seconds during 1 minutes. Set Type to **JVM_Monitoring** and the metric to **JVM_YoungGC_RT_ms**. The rule is that the total time in the last 1 minute is greater than or equal to 5000, that is, 5 seconds. You can modify the threshold as needed.

Create alert rules for browser monitoring by using the JS exception alert template

To create an alert rule for determining whether JavaScript (JS) exceptions occur, find the **DefaultRetcodeAlert** template and click **Create Alert** in the **Actions** column.

As shown in the preceding figure, this alert rule uses two metrics as alert triggers. An alert event is generated when the following two criteria are met:

- High JS error rate: The average JS error rate of the frontend application is greater than or equal to 20% during 10 minutes. Set Type to **Page_Metric** and the metric to **JS_Error_Rate**. The rule is that the average JS error rate in the last 10 minutes is greater than or equal to 0.2, that is, 20%.
- Excessive JS errors: The total number of JS errors of the frontend application is greater than or equal to 20 during 10 minutes. Set Type to **Page_Metric** and the metric to **JS_Error_Count**. The rule is that the total number of JS errors during the last 10 minutes is greater than or equal to 20.

References

For more information about the fields in alert rules and **advanced configuration**, see Description of basic fields.

12.Troubleshooting

12.1. Why is no alert notification received after I set an ARMS alert rule?

Condition

Application Real-Time Monitoring (ARMS) alert rules have been set, but no alert notification is received.

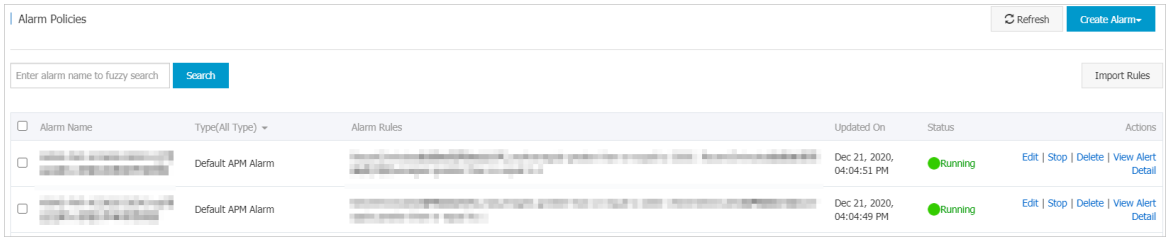
Cause

Except for default emergency alert rules, all other ARMS alert rules require that your system check for exceptions, determine the alert rule status, and generate alert events at an interval of one minute. An alert event is either triggered or not triggered. An alert notification is sent only when the alert event is triggered and the corresponding alert rule is not in a quiet period. If you cannot receive any alert notification after you set an alert rule, perform the following steps for troubleshooting.

Solution

Procedure

1. Log on to the ARMS console .
2. In the left-side navigation pane, choose **Alerts > Alert Policies**. On the **Alert Rules** tab, enter the target alert name in the search box, and then click **Search**. View the status in the **Status** column.



   - If the status is **Stopped**, click **Start** in the **Actions** column. In the **OK** dialog box, click **Start**. If you still cannot receive any alert notifications after you restart the alert rule, proceed with step 3.
   - If the status is **Running**, proceed with step 3.
3. Click **View Alert Detail** in the **Actions** column. On the **Alert History** tab, click the **Alert Event History** tab, and check whether the alert event has been triggered in the **Trigger** column.

   > **Note** A green icon in the **Trigger** column indicates that the alert event is not triggered. A red icon indicates that the alert event has been triggered.

   - If the alert event is not triggered, check whether the threshold for the alert rule is wrong in the **Alert Detail** column. If the threshold is wrong, on the **Alert Rules** tab, find the target alert rule, and click **Edit** in the **Actions** column. In the **Edit Alert** dialog box, reconfigure the threshold for the alert rule.
   - If the alert event has been triggered, proceed with step 4.
   - If no record exists in the alert event history, proceed with step 6.
4. On the **Alert History** tab, click the **Alert Post History** tab, and check for alert post records.

- If an alert post record exists but you still do not receive any alert notification, the upper limit on incoming messages may have been reached. Each mobile phone contact can receive up to 100 short messages per day, and each email contact can receive up to 50 emails per day. After the upper limit is exceeded, no more alert notifications can be received.
- If no alert post record exists, the alert may be in a quiet period. In this case, proceed with step 5.

5. On the **Alert Post History** tab, click the time range in the upper-right corner. In the list that appears, click **Last 24 Hours** to check for alert post records in the last 24 hours.
   - If an alert post record exists, click the **Alert Rules** tab, find the target alert rule, and then click **Edit** in the **Actions** column. In the **Advanced Configuration** section of the **Edit Alert** dialog box, turn off the **Alert Quiet Period** switch.

     > **Note** After you turn on the **Alert Quiet Period** switch, if the alert stays in the triggered state, an alert notification is sent only 24 hours after the first alert notification is sent. After you turn off this switch, ARMS sends an alert notification every minute.

   - If the alert post record section is still empty, the alert notification method or the contact configuration may be invalid.
     - In this case, click the **Alert Rules** tab, find the target alert rule, and then click **Edit** in the **Actions** column. In the **Edit Alert** dialog box, select a correct notification method.
     - Alternatively, in the left-side navigation pane, choose **Alerts > Contacts**. On the **Contact Management** tab, check whether the settings of Cell Phone Number, Email, DingTalk Robot, and Contact Group are correct. If any setting is invalid, reconfigure it.
6. In the left-side navigation pane, choose **Application Monitoring > Applications**. On the **Applications** page, check whether any data is generated for the application that is associated with the alert rule.
   - If no data is generated for the application, the application is not connected to ARMS and therefore no alert event is generated. In this case, check and solve the data generation problem.
   - If data is generated for the application but no data exists in a dimension of the alert rule, for example, no data exists in the Page_Name dimension of a Page_Metric browser monitoring alert, the dimension value may be invalid. In this case, click the **Alert Rules** tab, find the target alert rule, and then click **Edit** in the **Actions** column. In the **Edit Alert** dialog box, set **Dimension** to **Traversal**, and then reconfigure the dimension value by referring to the traversal alert details displayed on the **Alert Event History** tab.
7. If you still do not receive any alert notifications, contact the ARMS DingTalk account, arms160804.

13.References

13.1. Metrics of alert rules

Each alert metric belongs to a specific type. Each type has one or more dimensions. This topic describes the alert rule metrics for Application Real-Time Monitoring (ARMS) application monitoring and browser monitoring.

Background

Call errors differ from abnormal calls.

- A call error is identified when the returned status code for an entire external call is greater than 400.
- An abnormal call is a call during which an error is thrown due to an exception.

Metrics of application monitoring alert rules

Type	Dimension	Metric
Invocation_Statistic	Interface_Name	**Invocation_RT_ms**: indicates the response time of calls to the application entry point (including both HTTP and Dubbo calls), database, and internal system. Unit: millisecond. You can use this metric to check for slow requests and determine whether any application exception occurs. **Note** Compared with the response time metric displayed on pages, this metric also covers database calls and system internal calls.
		Invocation_Count: indicates the number of calls to the application entry point (including both HTTP and Dubbo calls), database, and internal system. You can use this metric to analyze the number of calls of an application, to determine the traffic volume and whether any exception occurs in the application. **Note** Compared with the number of calls displayed on pages, this metric also covers database calls and system internal calls.
		Invocation_ErrorCount: indicates the number of errors in calls to the application entry point (including both HTTP and Dubbo calls), database, and internal system. You can use this metric to determine whether application exceptions or application call errors occur. **Note** Compared with the number of errors displayed on pages, this metric also covers database calls and system internal calls.

Type	Dimension	Metric
Invocation_Type	Invocation_Type	**App_Inbound_Invoke_RT_ms**: indicates the response time of calls to the application entry point, including both HTTP and Dubbo calls. Unit: millisecond. You can use this metric to check for exceptions over the entire service trace.
		App_Inbound_Invoke_Request: indicates the number of calls to the application entry point, including both HTTP and Dubbo calls. You can use this metric to analyze the number of access requests over the entire service trace, to determine the traffic volume and whether any exception occurs in the application.
		App_Inbound_Exception_Rate: indicates the error rate of calls to the application entry point, including both HTTP and Dubbo calls. You can use this metric to check for errors over an entire service trace. For external services, you can determine the number of errors that occur when users failed to access the system.
		App_Outbound_Invoke_RT_ms: indicates the response time of downstream dependent service calls over the trace, such as inter-HTTP service calls and database calls. Unit: millisecond. You can use this metric to check for slow access to downstream systems, and determine whether any exception occurs in the application.
		App_Outbound_Invoke_Request: indicates the number of downstream dependent service calls over the trace, such as inter-HTTP service calls and database calls. You can use this metric to determine whether any exception occurs over the trace.
		App_Outbound_Exception_Rate: indicates the error rate of downstream dependent service calls over the trace, such as inter-HTTP service calls and database calls. You can use this metric to check for errors in downstream application calls, and determine whether exceptions occur over the entire trace.
Database_Metric	Database_Name	**DB_RT_ms**: indicates the response time for the application to call a database. Unit: millisecond. You can use this metric to check for slow database access from the application, and determine whether exceptions occur when the application calls the database or whether any exception exists in the database environment.
		DB_Count: indicates the number of database calls initiated by an application. You can use this metric to determine whether the application causes excessive pressure on the database and whether application exceptions occur.

Type	Dimension	Metric
		DB_ErrorCount: indicates the number of errors in database calls initiated by the application. You can use this metric to determine whether the application exception is caused by a database, and whether exceptions occur in the database or the database environment.
JVM_Monitoring	IP	**JVM_GcPsMarkSweepCount**: indicates the number of JVM tag cleanup events. This metric is not frequently used in alerts.
		JVM_Non_Heap_Used_byte: indicates the JVM non-heap memory utilization. You can use this metric to determine whether an application occupies too much non-heap memory in scenarios where non-heap memory is used. You can also use this metric to determine whether exceptions occur in the application.
		JVM_GcG1YoungGenCount: indicates the number of Garbage-First Garbage Collector (G1GC) events in the JVM_Young zone. This metric is not frequently used in alerts.
		JVM_FullGC_Count: indicates the number of full garbage collection (GC) events. If the value is too large, you can determine that exceptions occur in the application.
		JVM_FullGC_Time_ms: indicates the time used for full GC. If the value is too large, you can determine that exceptions occur in the application.
		JVM_Non_Heap_Init_byte: indicates the initial value of JVM non-heap memory. This metric is not frequently used in alerts.
		JVM_Non_Heap_Committed_byte: indicates the committed JVM non-heap memory. This metric is not frequently used in alerts.
		JVM_Young_GC_Instant_Count: indicates the number of JVM young GC events. If the value is too large, you can determine that exceptions occur in the application.
		JVM_GcG1OldGenCount: indicates the number of G1GC events in the JVM_Old zone. This metric is not frequently used in alerts.
		JVM_Non_Heap_Max_byte: indicates the maximum value of JVM non-heap memory. This metric is not frequently used in alerts.
		JVM_Heap_Total_byte: indicates the total capacity of JVM heap memory. If the value is too large, you can determine that exceptions occur in the system.

Type	Dimension	Metric
		JVM_ThreadCount: indicates the total number of JVM threads. You can use this metric to determine whether an application runs properly (a value greater than 0 indicates that the application runs properly), or whether a large number of threads exist, for example, in thread pool scenarios.
		JVM_GcPsScavengeCount: indicates the total number of JVM GC events. This metric is not frequently used in alerts.
		JVM_Young_GC_Time_Instant_ms: indicates the time used for JVM young GC. If the value is too large, you can determine that exceptions occur in the application.
Host_Monitoring	IP	**Node_User_CPU_percent**: indicates the CPU usage of a node. You can use this metric to determine whether an application occupies too much CPU resources. We recommend that the CPU usage of the application keep below 90%.
		Node_Net_In_Errs: indicates the number of error packets received by a node. You can use this metric to determine whether exceptions occur in the network where the node is located. This metric is not frequently used in alerts.
		Node_Disk_Free_byte: indicates the free disk space of a node. You can use this metric to determine whether a node disk is fully occupied. If a disk is fully occupied, exceptions may occur in the application.
		Node_Net_Out_Errs: indicates the number of error packets sent by a node. You can use this metric to determine whether exceptions occur in the network where the node is located. This metric is not frequently used in alerts.
		Node_Load: indicates the node load. You can use this metric to determine whether the current workload of a node is too high. For a node with N cores, keep the load below N. This metric is not frequently used in alerts.
		Node_MEM_Free_Byte: indicates the free memory of a node. You can use this metric to determine whether a node has sufficient memory. If the free memory of a node is low, exceptions such as out of memory (OOM) may occur.
Exception_Invocation	Interface_Name	**Exception_Call_Count**: indicates the number of abnormal calls of an application. An abnormal call is a call during which an error is thrown due to an exception. You can use this metric to determine whether a call stack throws errors and whether application call exceptions occur.

Type	Dimension	Metric
		Exception_Call_RT_ms: indicates the response time of abnormal calls of an application. An abnormal call is a call during which an error is thrown due to an exception. You can use this metric to determine the impact of errors thrown by the call stack on the call response time, and determine whether application call exceptions occur.

Metrics of browser monitoring alert rules

Type	Dimension	Metric
Page_API_Metric	**Page_Name**, **API_Name**	**Api_Fail_Time_ms**: indicates the average request time when the API call of a page fails. You can use this metric to check whether the API request time of a page is normal.
		Api_Request_Count: indicates the number of API calls of a specific page. You can use this metric to determine whether the API requests of a page are normal.
		Api_Success_Time_ms: indicates the average response time of successful API requests on a specific page. You can use this metric to check whether the response time of API requests on a page is normal.
		Api_Success_Rate: indicates the ratio of successful API calls to total calls on a specific page. You can use this metric to check whether API calls on a page are normal.
Custom_Statistics_Metric	**Custom_Statistics_Key**	**Custom_Statistics_Sum**: indicates the summation field that is manually reported. In ARMS, this field is accumulated for custom tracing points.
		Custom_Statistics_Average: indicates the average of the values that are manually reported. ARMS calculates the average value for custom tracing points.
		DNS_Lookup_ms: indicates the time consumed for page DNS connection. You can use this metric to determine whether the page access speed is normal.
		Custom_first_screen_time_ms: indicates the first page display time that is manually reported. You can use this custom performance metric to determine whether the page access speed is normal.
		Custom_first_time_to_interact_ms: indicates the first time available for interaction that is manually reported. You can use this custom performance metric to determine whether the page access speed is normal.

Type	Dimension	Metric
Page_Metric	Page_Name	**First paint time_ms**: indicates the period from the time when a request is initiated to the time when the browser parses the bytes of the first batch of HTML documents. You can use this metric to determine whether the page access speed is normal.
		Resource_Download_ms: indicates the time consumed for page resource loading. You can use this metric to determine whether the page access speed is normal.
		Custom_t1-t10_ms: are custom performance fields that are manually reported as needed. You can customize these fields.
		Page_View: indicates how many times a page is viewed.
		Time_to_First_Byte_TTFB_ms: indicates the response time of network requests. You can use this metric to determine whether the page access speed is normal.
		DOM_Ready_ms: indicates the HTML loading time, or the time elapsed before the document object model (DOM) is ready. You can use this metric to determine whether the page access speed is normal.
		DOM_Parsing_ms: indicates the time consumed for parsing the DOM on the page. You can use this metric to determine whether the page access speed is normal.
		Page_Satisfaction: is calculated based on the first render time. Satisfactory cases refer to the cases with the first render time less than 2,000 ms. Tolerable cases refer to the cases with the first render time greater than 2,000 ms and less than 8,000 ms. Satisfaction = (Satisfactory cases + Tolerable cases/2)/Total sample cases
		JS_Error_Count: indicates the number of JavaScript (JS) errors on a page.
		Time_to_First_interaction_ms: indicates the time when the page becomes operable. You can use this metric to determine whether the page speed is normal.
		Content_Download_ms: indicates the time consumed for transmitting page data. You can use this metric to determine whether the page access speed is normal.
		Fully_Loaded_Time_ms: indicates the time consumed for loading a page completely. Load = First render time + DOM parsing time + JS synchronization time + Resource loading time. You can use this metric to determine whether the page access speed is normal.

Type	Dimension	Metric
		TCP_Connection_ms: indicates the time consumed for connecting to a page over TCP. You can use this metric to determine whether the page access speed is normal.
		First_meaningful_paint_ms: indicates the time when the main content of the page first appears on the screen. You can use this metric to determine whether the page access speed is normal.
		The_uv_of_the_fail_api: indicates the number of users who fail to call an API. You can use this metric to determine the impact of API call errors.
		JS_Error_Rate: indicates the ratio of page views with JS errors to total page views. A higher JS error rate means a higher severity of JS errors.
		SSL_Connection_ms: indicates the time consumed for establishing a Secure Sockets Layer (SSL) connection. You can use this metric to determine whether the page speed is normal.
API_Metric	**API_Name**	**API_Success_Rate**: indicates the ratio of successful API calls to total API calls. You can use this metric to determine whether API calls are normal.
		API_Success_Time_ms: indicates the average response time of all successful API calls. You can use this metric to determine whether the response time is normal.
		The_uv_of_the_fail_api: indicates the number of users who fail to call an API. You can use this metric to determine the impact of API call errors.
		API_Fail_Time_ms: indicates the average response time of all failed API calls. You can use this metric to determine whether the response time is normal.
		API_Request_Count: indicates how many times an API is called. You can use this metric to determine whether API calls are normal.

Related information

- Create an alert
- Create common alert rules