Alibaba Cloud
Web应用防火墙

Monitoring and Alarm

Issue: 20200303

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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent**

ial, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
	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.
	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.
	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.
	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]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Contents

1 Configure alarm settings

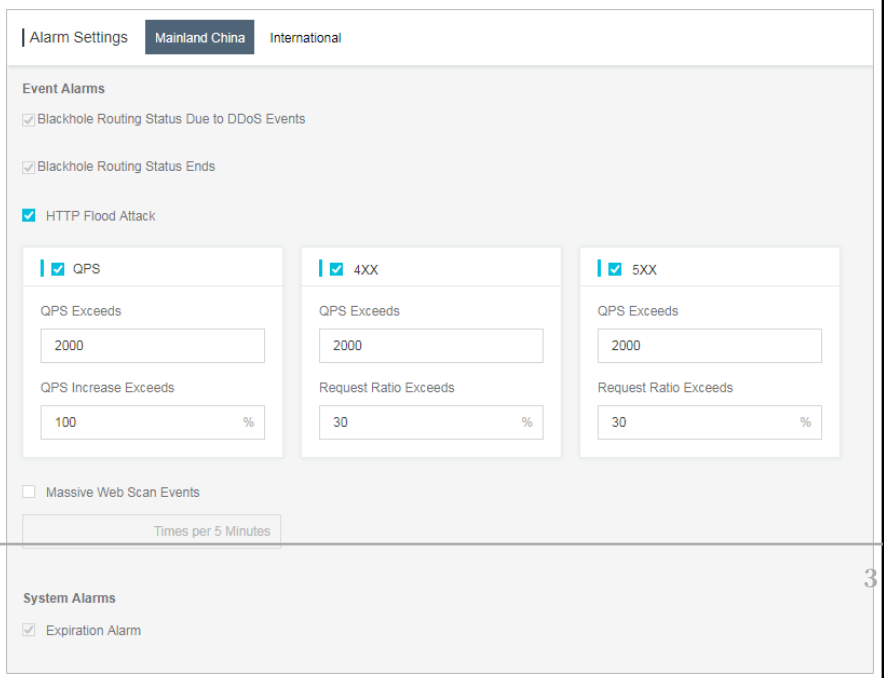
Context

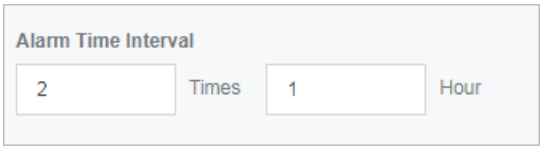
Alibaba Cloud WAF informs you about security events and system events through emails. You can configure the alarm triggering condition and alarm time interval.

Procedure

1. **Log on to the *Alibaba Cloud WAF console*.**
2. **On the top of the page, select the region: Mainland China or International.**

3. **On the Setting > Alarm Settings page, complete the following configuration.**

Configuration	Description
Triggered by	Specify which security or system event can trigger an alarm. **Note:** The default alarm cannot be disabled or configured. · Event Alarms - Blackhole Routing Status Due to DDoS Events (default) - Blackhole Routing Status Ends (default) - HTTP Flood Attack You must specify the conditions that trigger the alarm. ■ QPS exceeds a predefined maximum value (1 to 10,000,000) and increases by a predefined maximum rate (0% to 1,000%). ■ 4xx requests exceed a predefined maximum QPS (1 to 10,000,000) and occupy a predefined maximum proportion (0% to 1,000%). ■ 5xx requests exceed a predefined maximum QPS (1 to 10,000,000) and occupy a predefined maximum proportion (0% to 1,000%). - Massive Web Scan Events You must specify the maximum frequency per 5 minutes. · System Alarms: Expiration Alarm (default) 

Configuration	Description
Alarm Time Interval	Repeat alarms for xx (0 to 10) times per xx (0 to 24 hours). 

4. Click Save Settings.

2 Best practices of monitoring and alerting

2.1 Configure alert rules for Web Application Firewall

This topic describes how to configure alert rules for Web Application Firewall (WAF) in the CloudMonitor console. By configuring the alert notifications, you can learn about the traffic, connections, attacks, and other abnormal situations on WAF instances in a timely manner. The alert notifications can inform you immediately after the events occur and help you restore businesses at the earliest opportunity.

Context

CloudMonitor is a service that monitors Internet applications and Alibaba Cloud resources. It sends you notifications when alerts are triggered. You can customize alert rules to specify how the alert system checks the monitoring data and when it sends alert notifications. After you set alert rules for important metrics, you will be notified when exceptions are detected in these metrics. This allows you to manage exceptions quickly.

The alert feature of CloudMonitor is compatible with WAF, you can configure alert notification rules in the CloudMonitor console. CloudMonitor supports monitoring the following WAF data metrics.

Table 2-1: Monitor metrics for WAF

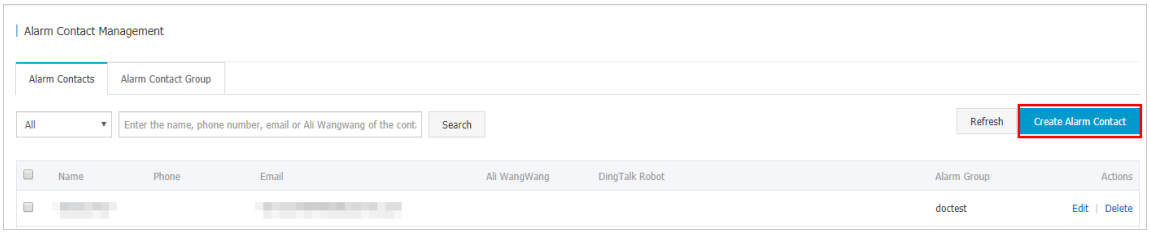
Monitor metric	Dimens	Unit	Description	Remarks
4XX_ratio	Domain name	%	The percentage of the 4xx HTTP status codes per minute (405 excluded).	The value is displayed in decimal notation in alert notifications.
5XX_ratio	Domain name	%	The percentage of the 5xx HTTP status codes per minute.	The value is displayed in decimal notation in alert notifications.
acl_blocks_5m	Domain name	Count	The number of requests blocked by access control within the last five minutes.	None

Monitor metric	Dimens	Unit	Description	Remarks
acl_rate_5m	Domain name	%	The percentage of requests blocked by access control within the last five minutes.	The value is displayed in decimal notation in alert notifications.
cc_blocks_5m	Domain name	Count	The number of requests blocked by HTTP flood protection within the last five minutes.	None
cc_rate_5m	Domain name	%	The percentage of requests blocked by HTTP flood protection within the last five minutes.	The value is displayed in decimal notation in alert notifications.
web_blocks_5m	Domain name	Count	The number of requests blocked by web attack protection within the last five minutes.	None
web_rate_5m	Domain name	%	The percentage of requests blocked by web attack protection within the last five minutes.	The value is displayed in decimal notation in alert notifications.
qps	Domain name	Count	The number of queries per second.	None
qps_ratio	Domain name	%	The growth rate of QPS every minute on the minute.	The value is displayed in percentage notation in alert notifications.
qps_ratio_down	Domain name	%	The decrease rate of QPS every minute on the minute.	The value is displayed in percentage notation in alert notifications.

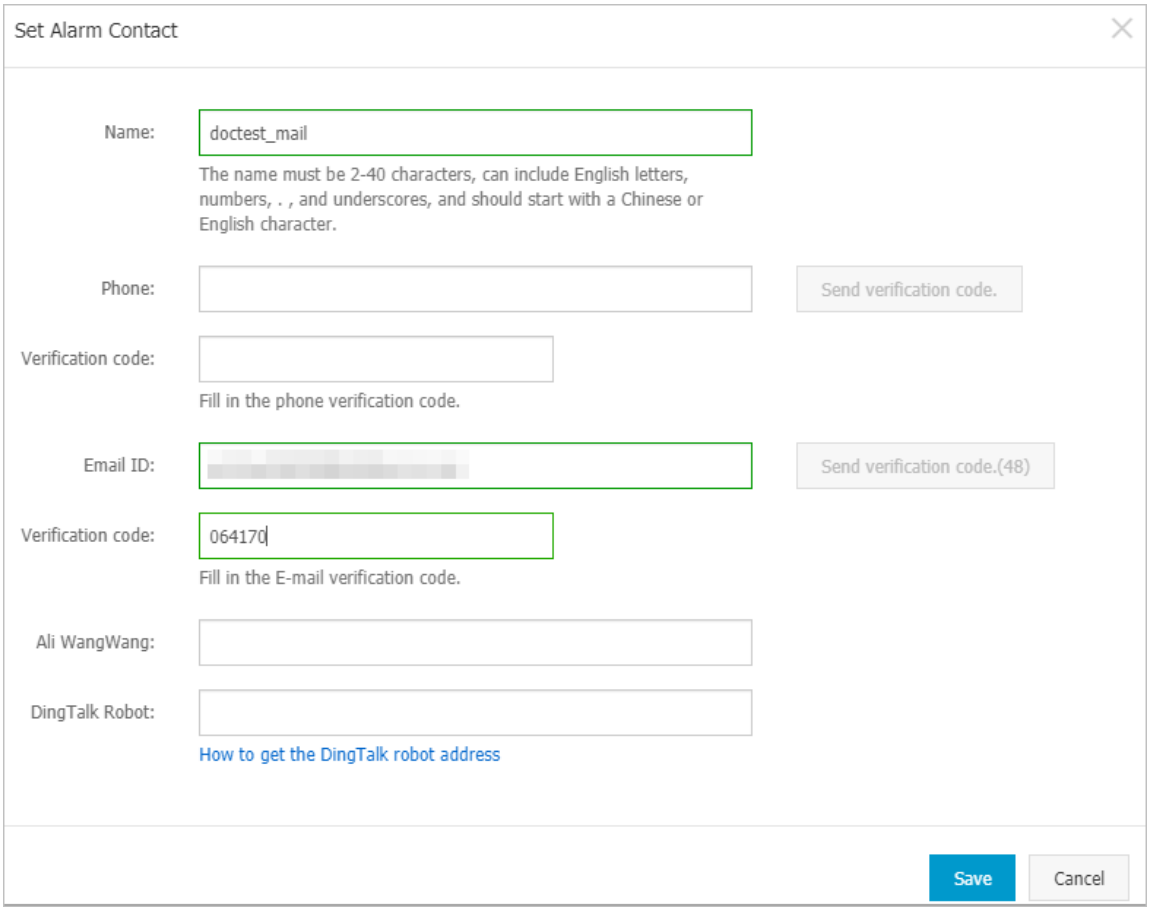
Procedure

1. Log on to the *CloudMonitor console*.

2. Optional: Add an alert recipient. If you have already specified a recipient, you can skip this step.
   a) In the left-side navigation pane, choose Alarms > Alarm Contacts.
   b) On the Alarm Contacts tab, click Create Alarm Contact in the upper-right corner.



   c) In the Set Alarm Contact dialog box that appears, enter the required contact information. Verify the Phone or Email ID, and then click Save.



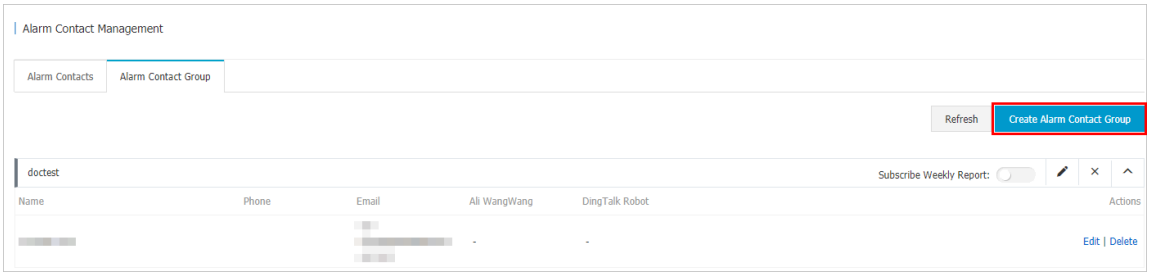
   The alert recipient is saved.

3. Optional: Create an alert contact group. If you have already created an alert contact group, you can skip this step.

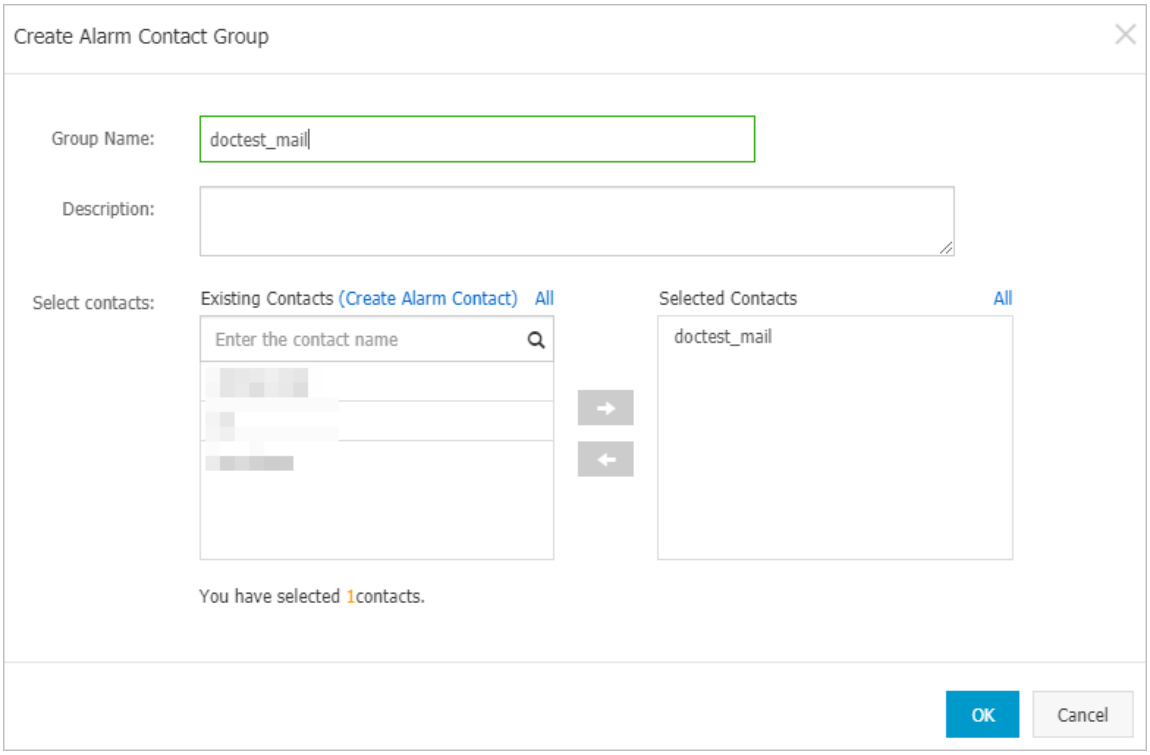
Note:

The recipients of alert notifications must be contact groups. You can add one or more recipients to a contact group.

a) **On the Alarm Contact Group tab, click Create Alarm Contact Group in the upper-right corner.**



b) **In the Create Alarm Contact Group dialog box that appears, enter a group name in the Group Name field. Select recipients from the left-side Existing Contacts list and add them to the right-side Selected Contacts list. Click OK.**

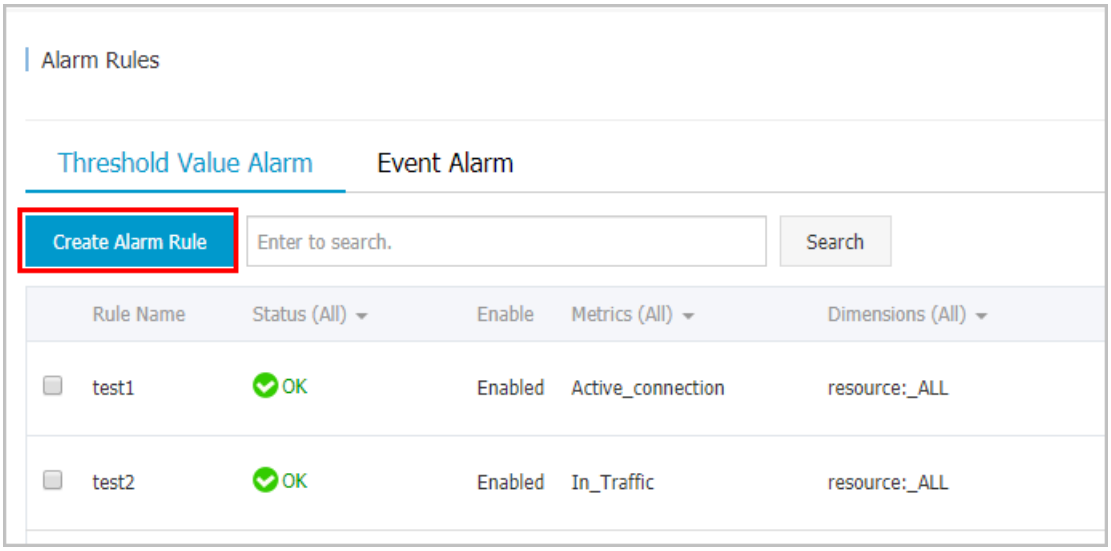


The contact group is created.

4. Create an alert rule

a) In the left-side navigation pane, choose Alarms > Alarm Rules.

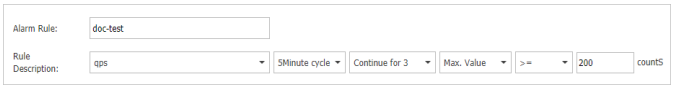
b) On the Threshold Value Alarm tab, click Create Alarm Rule.



c) Configure the alert rule on the Create Alarm Rule page and click Confirm. The following table lists the parameters and descriptions.

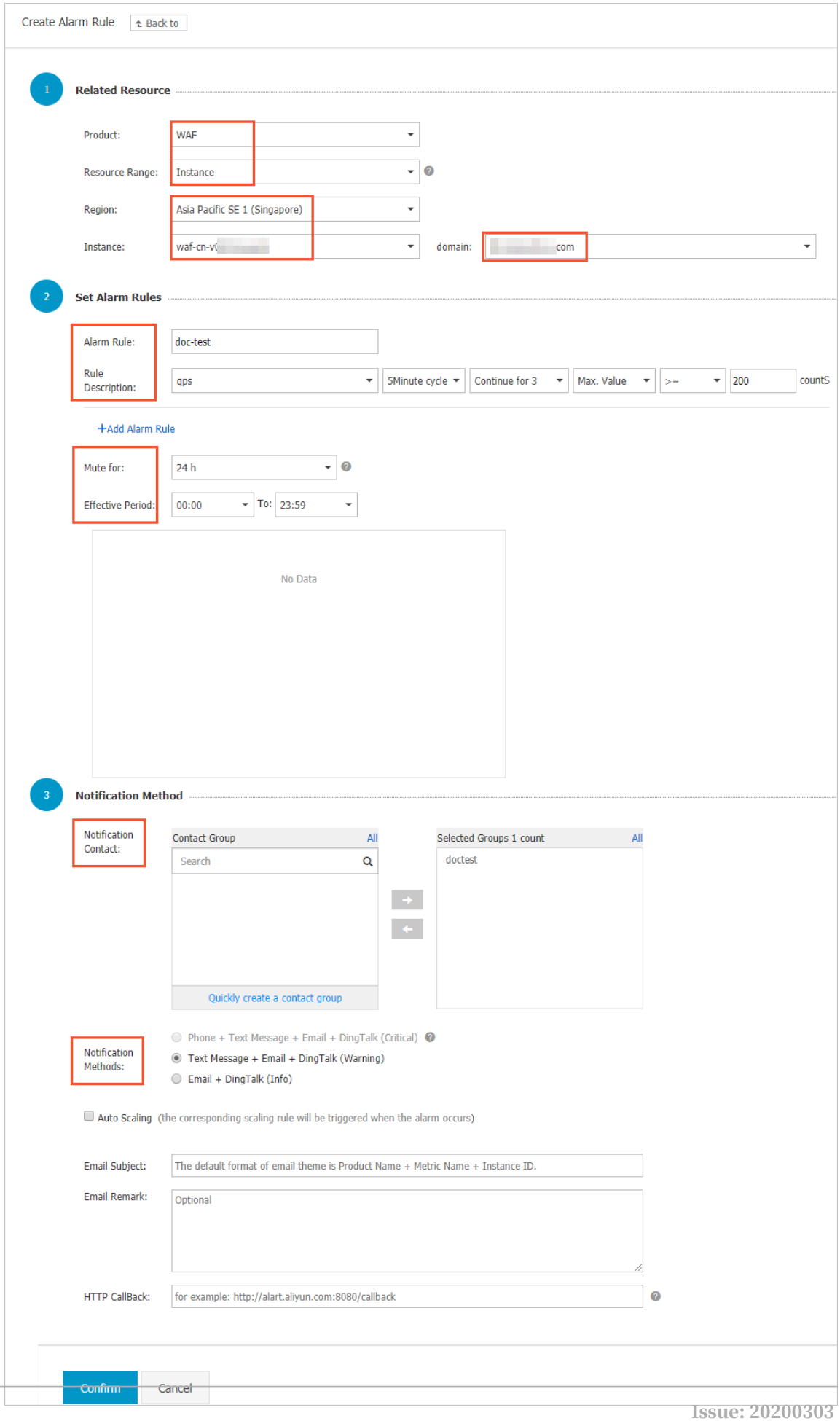
Type	Configuration item	Description
Related Resource	Product	Select WAF from the drop-down list.
	Resource Range	The resources to which the alert rule is applied. You can select All Resources or Instances. · All Resources: The alert rule is applied to all WAF instances. An alert is triggered when any of the WAF instances matches the specified rule. · Instances: The alert rule is applied to the selected WAF instances. An alert is triggered when one of the selected instances matches the specified rule.

Type	Configuration item	Description
	Region	This configuration item is required only if you select Instances from the Resource Range drop-down list. Select the region of the WAF instance. · For instances in mainland China, select China East 1 (Hangzhou). · For instances outside mainland China, select Asia Pacific SE 1 (Singapore).
	Instance	This configuration item is required only if you select Instances from the Resource Range drop-down list. By default, the WAF instance in the selected region is selected after you configure Region.
	Domain	This configuration item is required only if you select Instances from the Resource Range drop-down list. You can select one or more domain names from the domain names protected by the current instance.

Type	Configuration item	Description
Set Alarm Rules	Alarm Rule	Specifies a name for the alert rule.
	Rule Description	Specifies the conditions that trigger alerts. **Note:** We recommend that you set the thresholds of metrics based on your actual business requirements. For more information, see *Table 2-1: Monitor metrics for WAF*. A low threshold may frequently trigger alerts and negatively impact user experience. A high threshold may leave insufficient time for you to handle attacks. Sample alert rule description: The rule description: QPS, 5Minute cycle , Continue for 3, and the Max. Value > 200 . This alert rule indicates that the alert service detects the QPS data within any three cycles in a row. If the maximum QPS within three cycles is greater than 200, an alert is triggered. A data point is reported for each metric every 60 seconds. A total of 15 data points is reported within three consecutive cycles.  You can click Add Alarm Rule to add more alert rules. Specify the Alarm Rule and Rule Description for each alert rule.
	Mute for	Specifies a mute period. If the alert is not cleared within the mute period, a new alert notification is sent when the mute period ends. The minimum value is five minutes and the maximum value is 24 hours.

Type	Configuration item	Description
	Effective Period	The time period during which the alert rule remains effective. The system only sends alerts within the effective period. The system only records alerts if they occur before or after the effective period.
Notification Method	Notification Contact	The contact group that receives alerts.
	Notification Methods	Alert levels include Critical , Warning, and Info. The alerts of different levels are sent through different methods. Valid values: · Phone + Text Message + Email + DingTalk (Critical) Note: You can select this notification method only after you purchase a notification plan that supports phone calls. · Test Message + Email + DingTalk (Warning) · Email + DingTalk (Info)
	Auto Scaling	After you specify a scaling rule, the specified scaling rule is triggered when an alert occurs . This option is not required.
	Email Remark	Optional. The custom additional information in the alert notification email. Remarks will be included in the alert notification email.

Type	Configuration item	Description
	HTTP Callback	CloudMonitor uses a POST request to push an alert to the specified public URL address. Currently, only HTTP requests are supported.


Create Alarm Rule
Back to
1
Related Resource
Product:
WAF
Resource Range:
Instance
Region:
Asia Pacific SE 1 (Singapore)
Instance:
waf-cn-v
domain:
com
2
Set Alarm Rules
Alarm Rule:
doc-test
Rule Description:
qps
5Minute cycle
Continue for 3
Max. Value
>=
200
countS
+Add Alarm Rule
Mute for:
24 h
Effective Period:
00:00
To:
23:59
No Data
3
Notification Method
Notification Contact:
Contact Group
All
Search
Quickly create a contact group
Selected Groups 1 count
All
doctest
Notification Methods:
Phone + Text Message + Email + DingTalk (Critical)
Text Message + Email + DingTalk (Warning)
Email + DingTalk (Info)
Auto Scaling (the corresponding scaling rule will be triggered when the alarm occurs)
Email Subject:
The default format of email theme is Product Name + Metric Name + Instance ID.
Email Remark:
Optional
HTTP CallBack:
for example: http://alart.aliyun.com:8080/callback
Confirm
Cancel

You have created a WAF alert rule. When the WAF monitoring metrics meet the conditions described in an alert rule, alert notifications are sent to the specified contact group.

2.2 Configure event monitoring for Web Application Firewall

This topic describes how to configure the event monitoring rules for Web Application Firewall (WAF) in the CloudMonitor console. You can configure CloudMonitor to monitor events on the domains protected by WAF and send alerts to you. This can help you recover your workloads at the earliest opportunity. CloudMonitor can detect WAF events detected by access control, HTTP flood attack protection, web attack protection, and anti-scanning.

Context

CloudMonitor is a service that monitors Internet applications and Alibaba Cloud resources. It supports event monitoring, which allows you to query and analyze system events occur on cloud services. With event monitoring, you can easily track how your cloud services are used.

You can query the access control, HTTP flood attacks, web attacks, and anti-scanning events that have been detected on the domains protected by WAF and add alert rules for the events. You can configure CloudMonitor to alert you about critical events by sending messages or returning callbacks. Supported messages include SMS messages, emails, and DingTalk messages. With the event monitoring feature, you can build an automated operations and maintenance system to detect and handle events at the earliest opportunity. For more information, see *An overview of event monitoring*.

CloudMonitor supports monitoring the following WAF events.

Table 2-2: WAF events

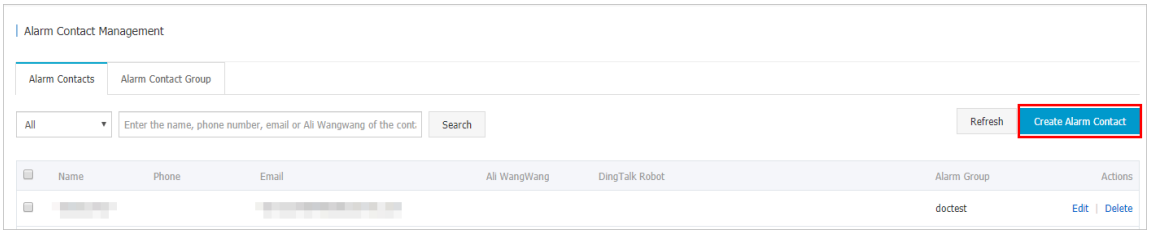
Event	Description	Type	Status value	Event level
waf_event_aclattack	Access control events	acl	start/end	CRITICAL
waf_event_ccattack	HTTP flood attack events	cc	start/end	CRITICAL

Event	Description	Type	Status value	Event level
waf_event_webattack	Web attack events	web	start/end	CRITICAL
waf_event_webscan	Anti-scanning events	webscan	start/end	CRITICAL

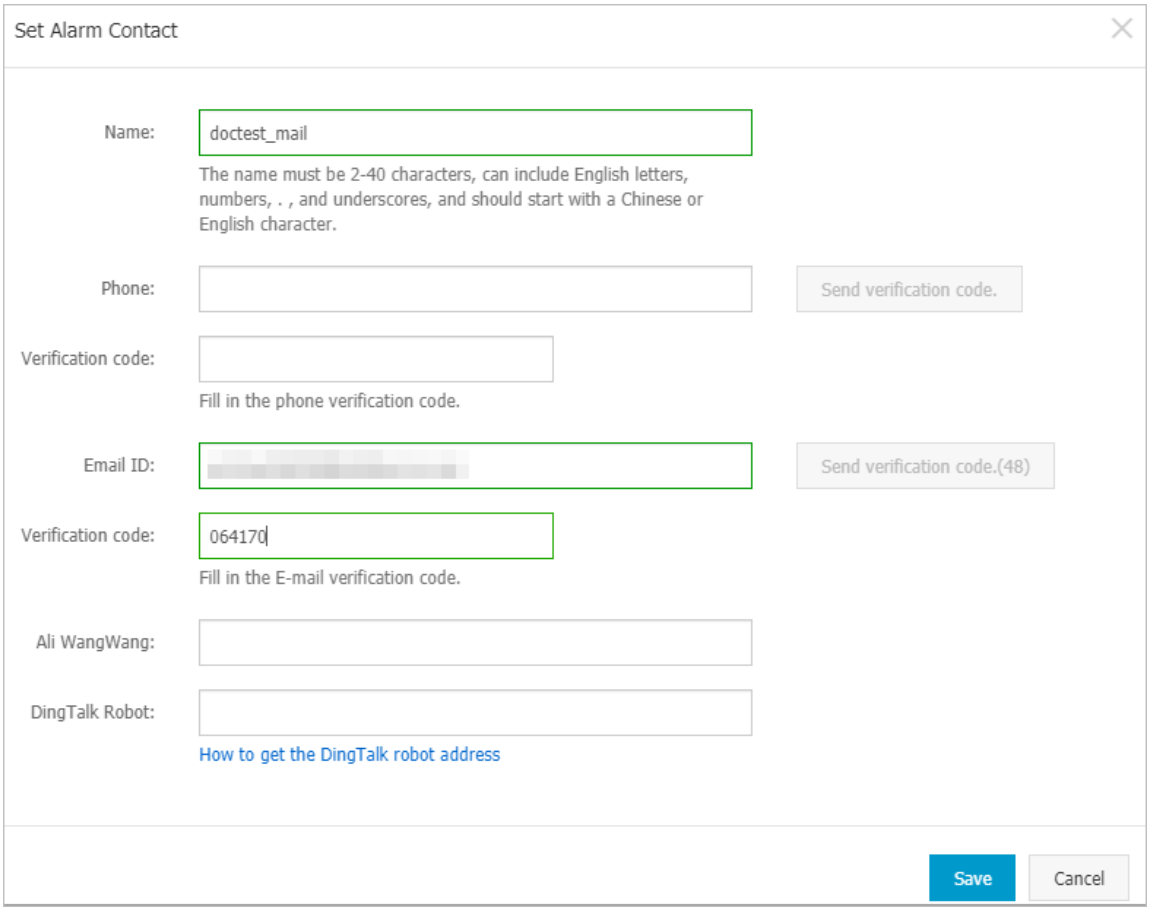
Procedure

1. **Log on to the** *CloudMonitor console*.

2. Optional: Add an alert recipient. If you have already specified a recipient, you can skip this step.
   a) In the left-side navigation pane, choose Alarms > Alarm Contacts.
   b) On the Alarm Contacts tab, click Create Alarm Contact in the upper-right corner.



   c) In the Set Alarm Contact dialog box that appears, enter the required contact information. Verify the Phone or Email ID, and then click Save.



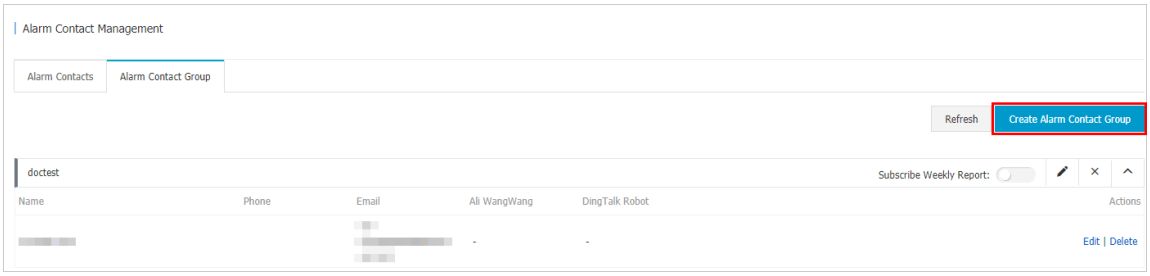
   The alert recipient is saved.

3. Optional: Create an alert contact group. If you have already created an alert contact group, you can skip this step.

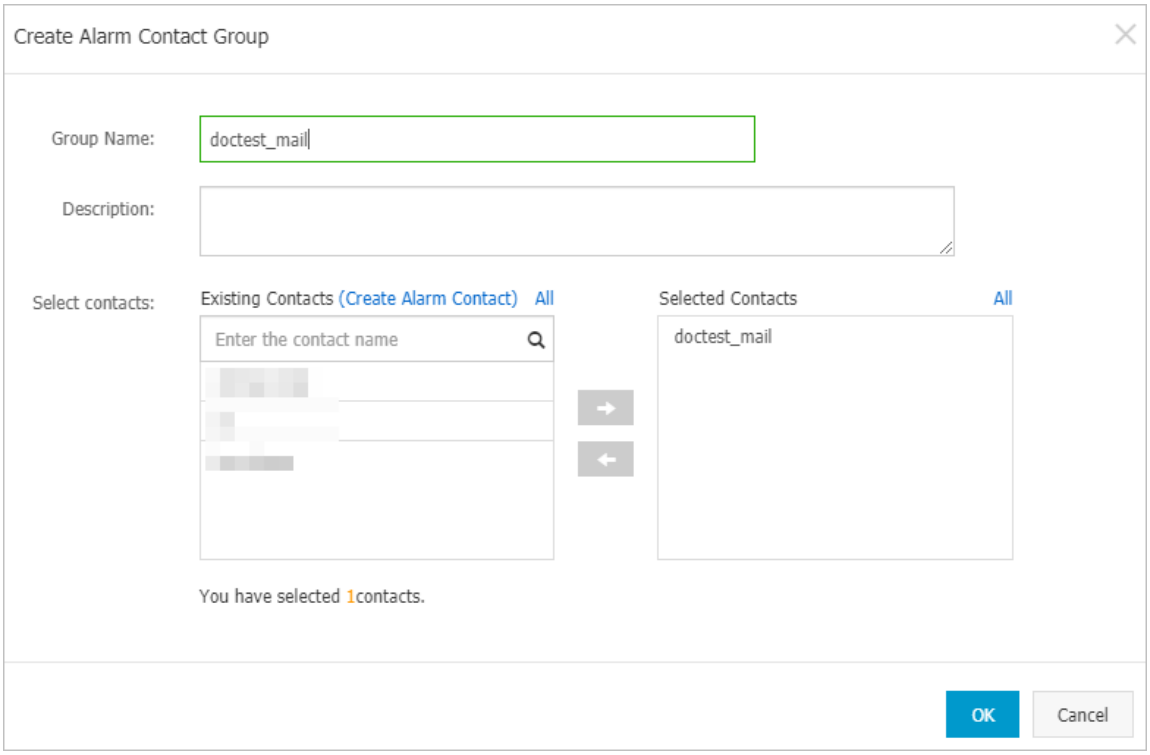
Note:

The recipients of alert notifications must be contact groups. You can add one or more recipients to a contact group.

a) **On the Alarm Contact Group tab, click Create Alarm Contact Group in the upper-right corner.**



b) **In the Create Alarm Contact Group dialog box that appears, enter a group name in the Group Name field. Select recipients from the left-side Existing Contacts list and add them to the right-side Selected Contacts list. Click OK.**

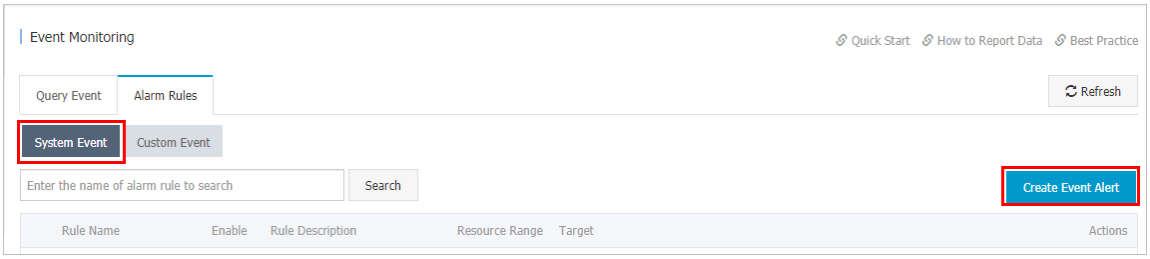


The contact group is created.

4. Create an event alert rule for WAF.

   a) In the left-side navigation pane, click Event Monitoring.

   b) On the Alarm Rules tab, select System Event, and click Create Event Alert.



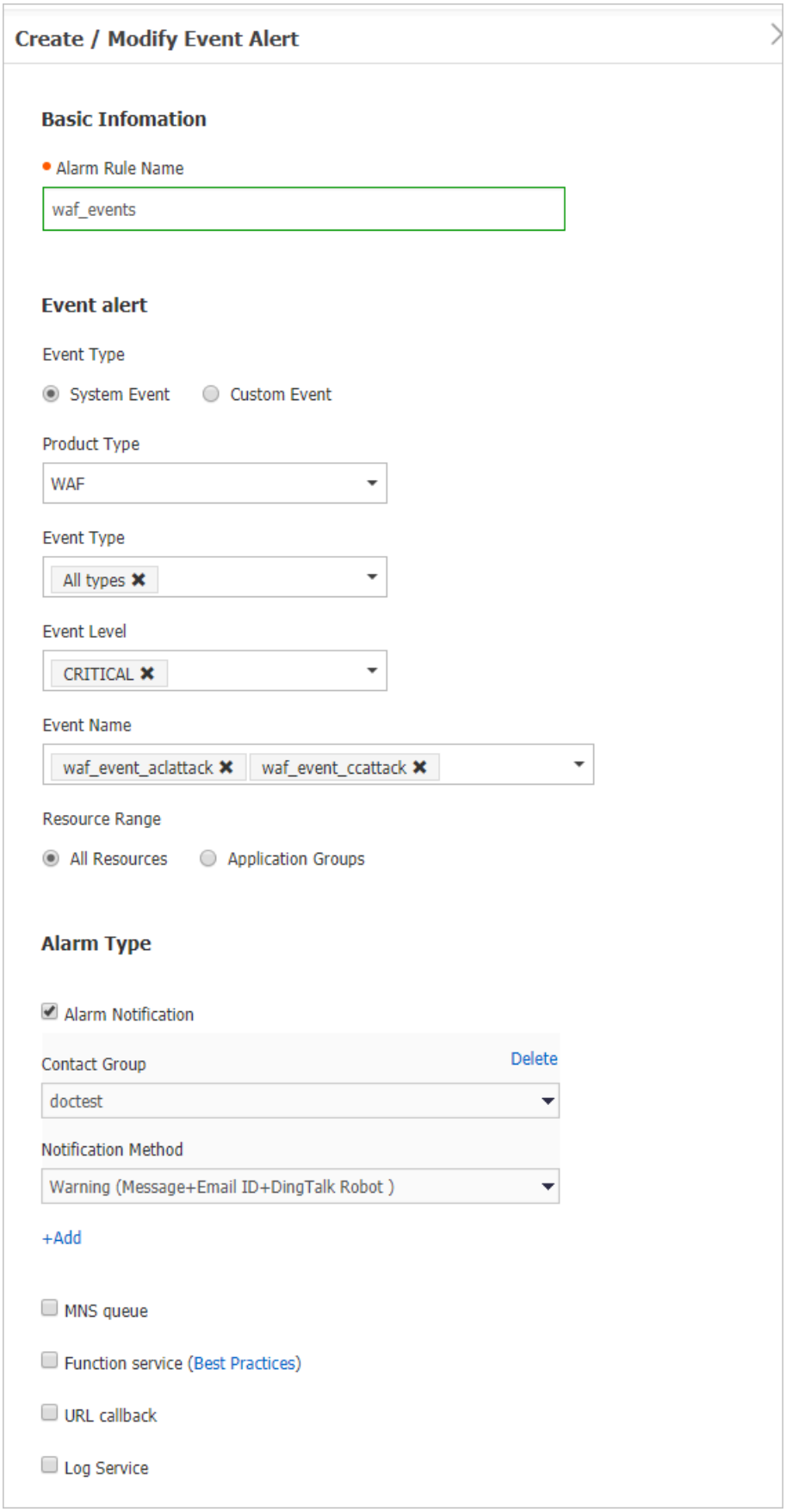
   c) Configure the alert rule in the Create/Modify Event Alert pane and click OK.

   The parameters are described as follows.

Category	Configuration item	Description
Basic information	Alarm Rule Name	Enter the name of the alert rule.
Event alert	Event Type	Select System Event.
	Product Type	Select WAF.
	Event Type	Select WAF attack events.
	Event Level	Select the level of the event. Valid values:CRITICAL, WARN, and INFO. You can select multiple levels but you must select CRITICAL.
	Event Name	Select the type of events which CloudMonitor will alert you about. Valid values: · waf_event_aclattack · waf_event_ccattack · waf_event_webattack · waf_event_webscan You can select multiple event types. The event levels must be CRITICAL.
	Resource Range	Select All Resources.

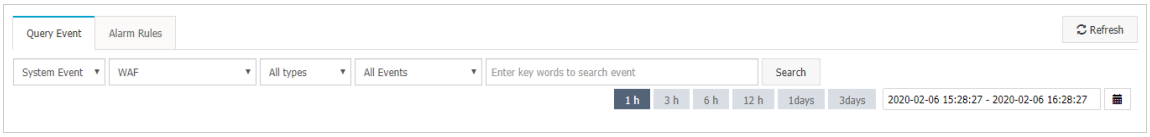
Category	Configuration item	Description
Alarm Type	Alarm Notification	Select Alarm Notification, and configure the Contact Group and Notification Method. · Contact Group: Select an existing contact group. · Notification Method: Select Warning (Message+Email ID+DingTalk Robot) or Info (Email ID+DingTalk Robot). You can click Add to add more contact groups and notification methods.
	MNS queue	This option is not required.
	Function service	This option is not required.
	URL callback	This option is not required.

Category	Configuration item	Description
	Log Service	This option is not required.


Create / Modify Event Alert
Basic Infomation
Alarm Rule Name
waf_events
Event alert
Event Type
System Event
Custom Event
Product Type
WAF
Event Type
All types
Event Level
CRITICAL
Event Name
waf_event_aclattack
waf_event_ccattack
Resource Range
All Resources
Application Groups
Alarm Type
Alarm Notification
Contact Group
Delete
doctest
Notification Method
Warning (Message+Email ID+DingTalk Robot)
+Add
MNS queue
Function service (Best Practices)
URL callback
Log Service

You have created a WAF event alert rule. When a specific event occurs on a domain name that is added to WAF, the contact group specified in the alert rule receives an alert.

5. Optional: Query events. You can also query the recent WAF events in the CloudMonitor console.
   a) Click the Query Event tab of the Event Monitoring page.
   b) Select System Event and WAF, and specify the event type and time period to query the events.



   c) You can click View the Detail in the Operation column to view details of an event.

2.3 Create a monitoring dashboard for Web Application Firewall

This topic describes how to create and customize real-time monitoring dashboards and add charts to the dashboards for Web Application Firewall (WAF) in the CloudMonitor console. By using custom dashboards and charts, you can view detailed information about how your workloads are protected by WAF in a visualized manner.

Context

CloudMonitor is a service that monitors Internet applications and Alibaba Cloud resources. It allows you to view monitoring data in custom dashboards. You can aggregate monitoring data of different products and instances running the same type of workloads by using one dashboard.

You can configure dashboards for WAF in the CloudMonitor console. CloudMonitor supports monitoring the following WAF data metrics.

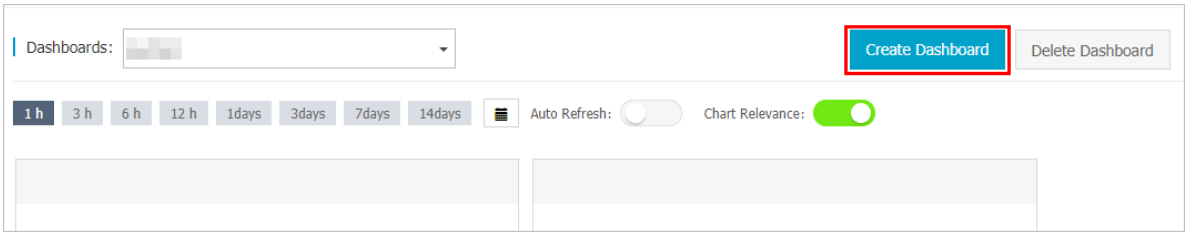
Table 2-3: Monitor metrics for WAF

Monitor metric	Dimens	Unit	Description	Remarks
4XX_ratio	Domain name	%	The percentage of the 4xx HTTP status codes per minute (405 excluded).	The value is displayed in decimal notation in alert notifications.

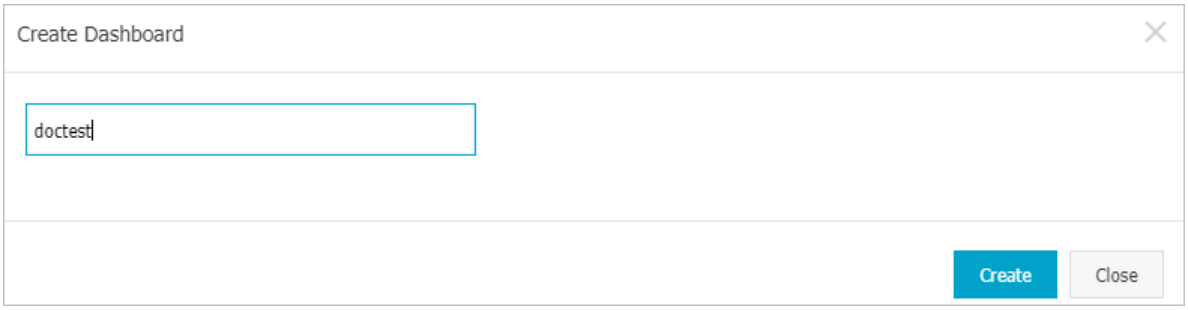
Monitor metric	Dimens	Unit	Description	Remarks
5XX_ratio	Domain name	%	The percentage of the 5xx HTTP status codes per minute.	The value is displayed in decimal notation in alert notifications.
acl_blocks_5m	Domain name	Count	The number of requests blocked by access control within the last five minutes.	None
acl_rate_5m	Domain name	%	The percentage of requests blocked by access control within the last five minutes.	The value is displayed in decimal notation in alert notifications.
cc_blocks_5m	Domain name	Count	The number of requests blocked by HTTP flood protection within the last five minutes.	None
cc_rate_5m	Domain name	%	The percentage of requests blocked by HTTP flood protection within the last five minutes.	The value is displayed in decimal notation in alert notifications.
web_blocks_5m	Domain name	Count	The number of requests blocked by web attack protection within the last five minutes.	None
web_rate_5m	Domain name	%	The percentage of requests blocked by web attack protection within the last five minutes.	The value is displayed in decimal notation in alert notifications.
qps	Domain name	Count	The number of queries per second.	None
qps_ratio	Domain name	%	The growth rate of QPS every minute on the minute.	The value is displayed in percentage notation in alert notifications.
qps_ratio_down	Domain name	%	The decrease rate of QPS every minute on the minute.	The value is displayed in percentage notation in alert notifications.

Procedure

1. Log on to the *CloudMonitor console*.
2. Choose Dashboard > Custom Dashboard, and then click Create Dashboard.

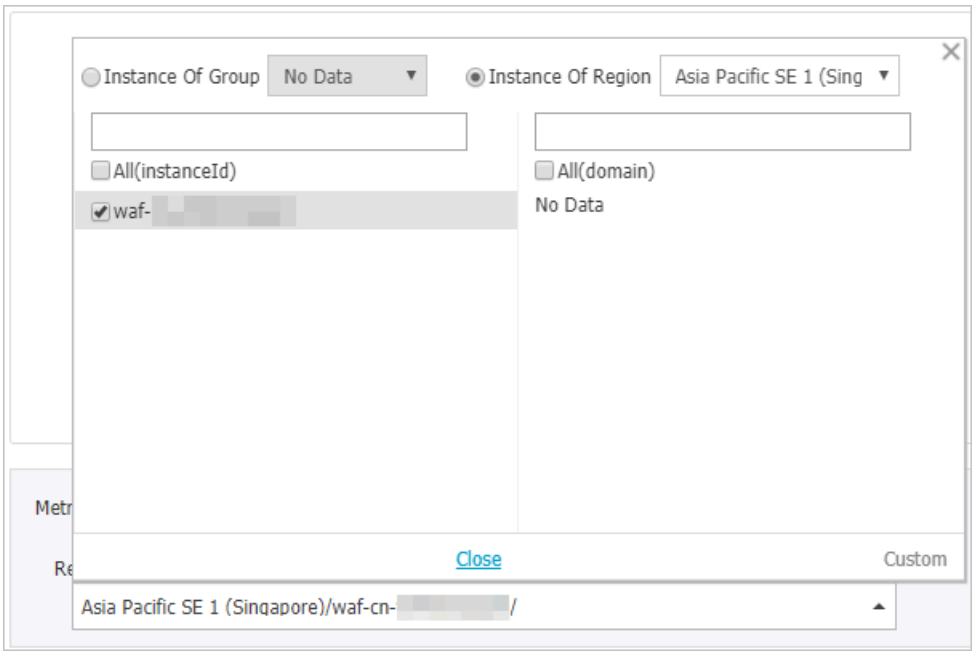


3. In the Create Dashboard dialog box that appears, specify a name for the dashboard, and then click Create.

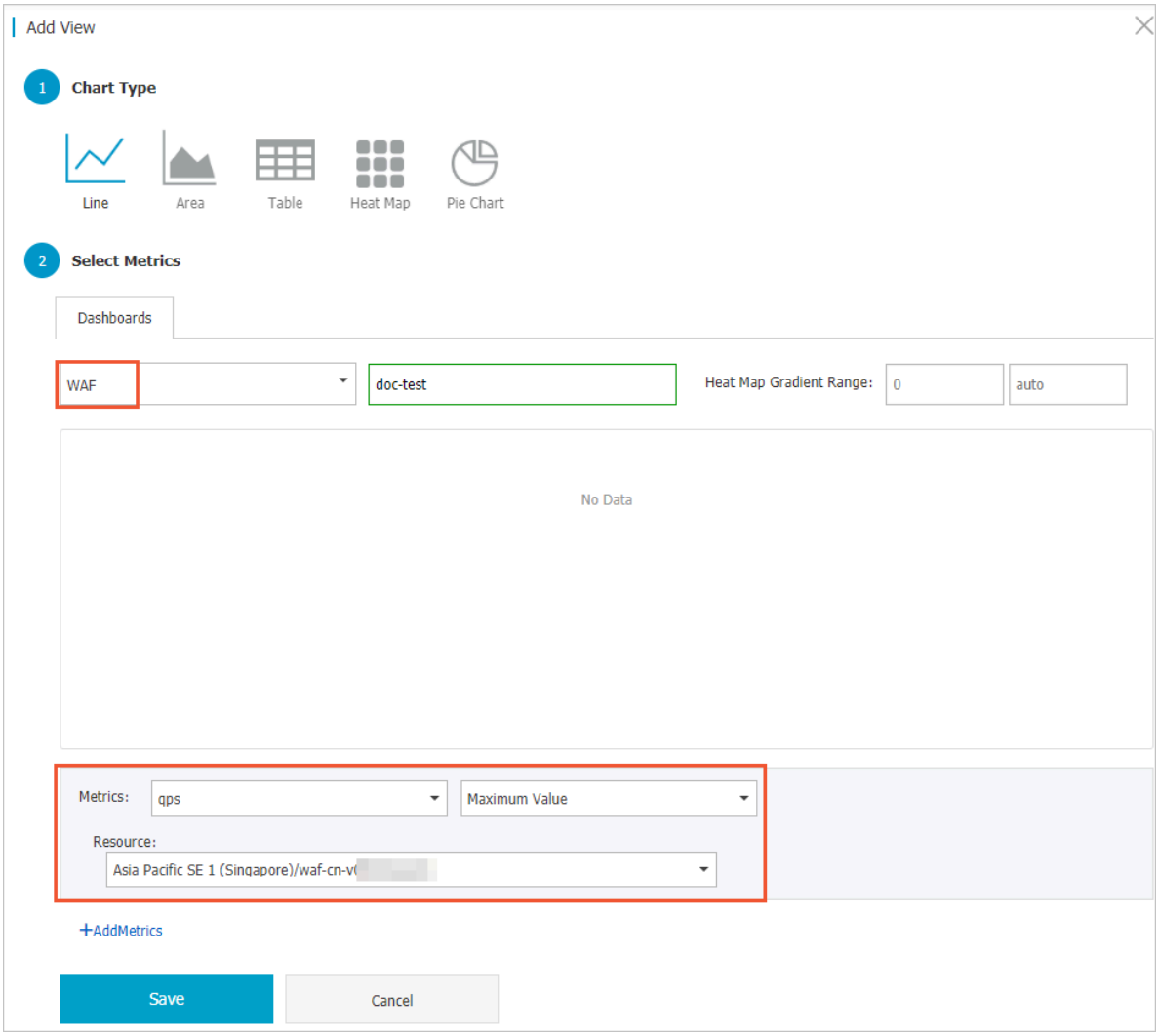


After the dashboard is created, you are redirected to the Dashboards page. You can select a dashboard from the Dashboards drop-down list to view or manage the selected dashboard.

4. Choose Custom Dashboard and click Add View, set the required parameters on the Add View page that appears on the right side.
   a) Select a chart type. Supported chart types include line, area, and pie charts, TopN tables, and heat maps.
   b) Select Metrics. Click the Dashboards tab and select WAF. Select a metric from the Metrics drop-down list and select resources from the Resource drop-down list.
      - Metrics: Select a metric to be monitored. For more information, see *WAF metrics*.
      - Resource: Select the domain names to be monitored.



      Click AddMetrics if you want to add more metrics.
   c) Click Save to create the chart.



You have created a WAF monitoring chart.

5. **To add more charts to the dashboard, repeat step 4. For more information, see *#unique_9* and *#unique_10*.**

3 Best practices for configuring alerts in Log Service

3.1 Overview

In this practice, the alert feature of Alibaba Cloud Log Service is used to configure custom monitoring charts and alerts for domain names that are added to WAF and have Log Service enabled. Enterprise users and individual users can refer to this practice to monitor the traffic and security status of their workloads and configure alerts.

Procedure

This practice contains the following steps.

Step	Description
Step 1: create a WAF log analysis dashboard	**After you use Log Service in WAF to initiate log query and analysis, you can create a dashboard based on the SQL statement. By default, the dashboard contains the charts generated based on the SQL statements.**
Step 2: configure log charts	**After you create a log analysis dashboard, you can edit or delete log charts on the dashboard or create a new log chart by copying an existing chart.**
Step 3: Configure a log alert	**After you create a log analysis dashboard, you can configure log alert on the dashboard. You must associate an alert with an existing log chart and set the alert trigger conditions based on the parameters in the associated chart. You can customize the alert message template.**

Configuration examples

This practice provides 13 examples of log charts and alert configurations, including alerts on an abnormal percentage of 4xx status codes (blocked requests excluded), alerts on an abnormal percentage of 5xx status codes, alerts on an abnormal query rate, alerts on an abrupt increase in query rate, alerts on an abrupt decrease in query rate, alerts on requests blocked by HTTP ACL policy in the last five minutes , alerts on requests blocked by web application protection in the last five minutes,

alerts on requests blocked by HTTP flood protection in the last five minutes, alerts on requests blocked by anti-scan rules in the last five minutes, alerts on the number of attacks from a single source IP address in the last five minutes, alerts on the number of domains attacked by a single IP address in the last five minutes, alerts on average delay in the last five minutes, and alerts on an abrupt decrease in query rate from a user.

We recommend that you learn how to configure a log chart (step 2), configure an alert rule (step 3), and then create chart and configure an alert rule. For more information, see *WAF log charts and alert configuration examples*.

For more information about the metrics used in alert configuration and the recommended thresholds for the metrics, see *Common monitoring metrics*.

For more information about the SQL statements used to query and analyze logs, see *Common SQL statements*.

3.2 Step 1: create a WAF log analysis dashboard

After you use Log Service in WAF to initiate log query and analysis, you can create a dashboard based on the SQL statement. By default, the dashboard contains the chart generated based on the SQL statements.

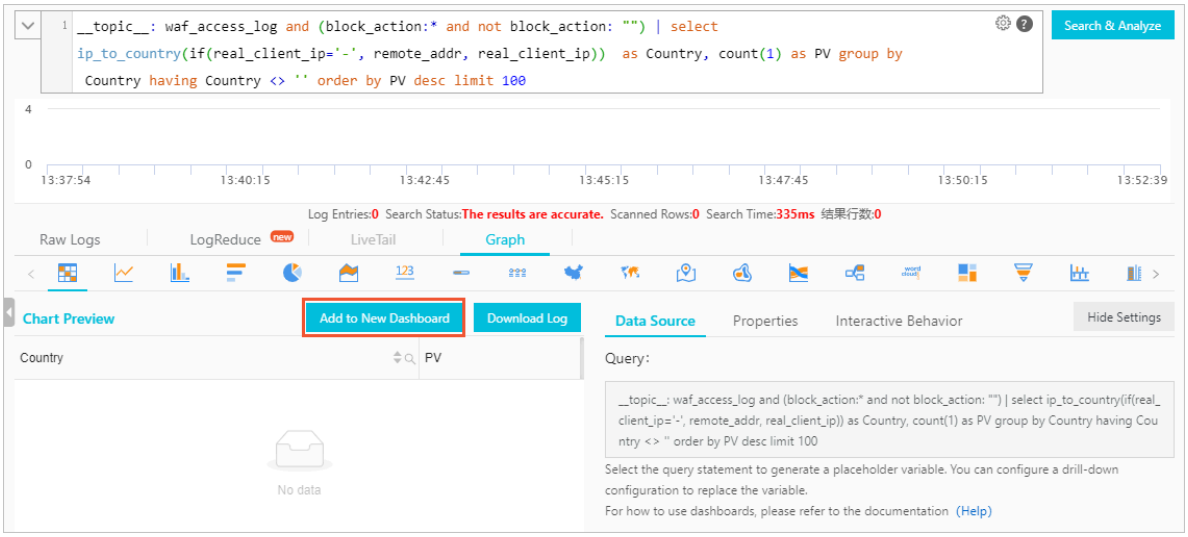
Prerequisites

- You have added your domain name to WAF for protection. For more information, see *#unique_19*.
- You have enabled Log Service for your domain name in the WAF console. For more information, see *#unique_20*.

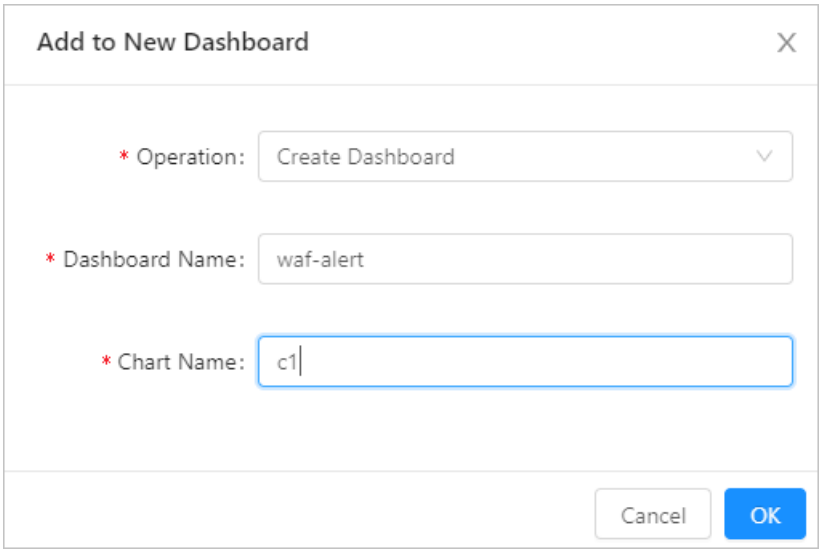
Procedure

1. Log on to the *WAF console*.
2. Go to the advanced management page of Log Service.
   a) In the upper part of the page, set the region to Mainland China or International. In the left-side navigation pane, choose App Market > App Management.
   b) In Real-time Log Query and Analysis Service, click Configure.
   c) In the upper-right corner of the Log Service page, click Advanced Settings.
   d) In the dialog box that appears, click OK.

3. In the project list, find the target log project, and click the project name.
4. Enter SQL statements, and click Search & Analyze.
5. After the query is complete, click Add to New Dashboard on the Graph tab.



6. In the Add to New Dashboard dialog box, set the following parameters, and click OK.



Parameter	Description
Operation	Select Create Dashboard.
Dashboard Name	Enter a dashboard name.
Chart Name	Enter a name for the chart generated based on the SQL statements.

Result

After the dashboard is created, you are redirected to the new dashboard. By default, the dashboard contains the chart generated based on the SQL statements entered in step 4. You can edit the charts or create more charts on the dashboard.

What's next

Step 2: configure log charts

3.3 Step 2: configure log charts

After you create a log analysis dashboard, you can edit or delete log charts on the dashboard or create a new log chart by copying an existing chart.

Prerequisites

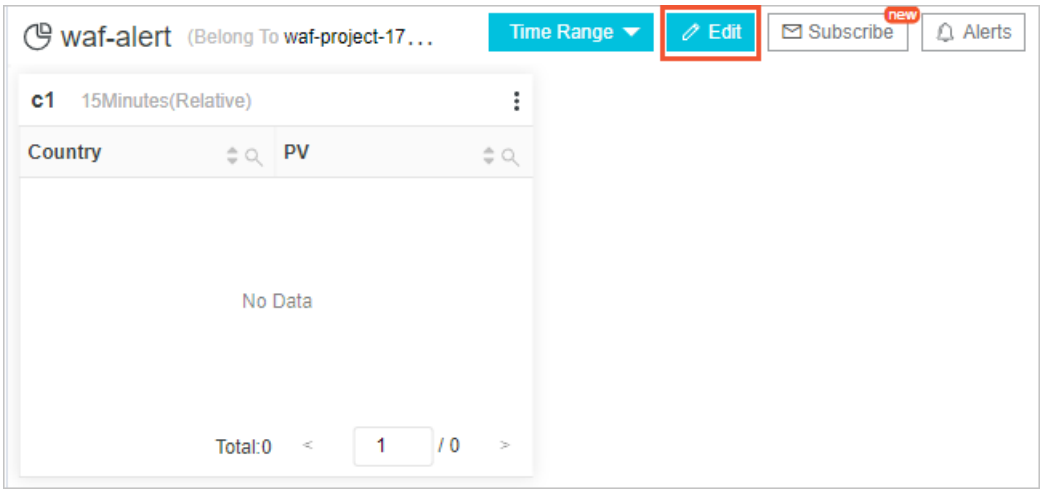
You have created a log analysis dashboard. For more information, see *Step 1: create a WAF log analysis dashboard*.

Context

This practice provides 13 default chart configuration examples. For more information, see *WAF log charts and alert configuration examples*. We recommend that you learn the alert configuration steps before you create a chart based on the examples and configure alerts during the chart creation process. For more information about alert configuration, see *Step 3: Configure a log alert* .

Procedure

1. Enter the customized WAF log analysis dashboard.
2. In the upper-right corner of the dashboard, click Edit.



The dashboard enters the edit mode. In this mode, you can edit or delete the charts on the dashboard or copy a chart to create a new chart.

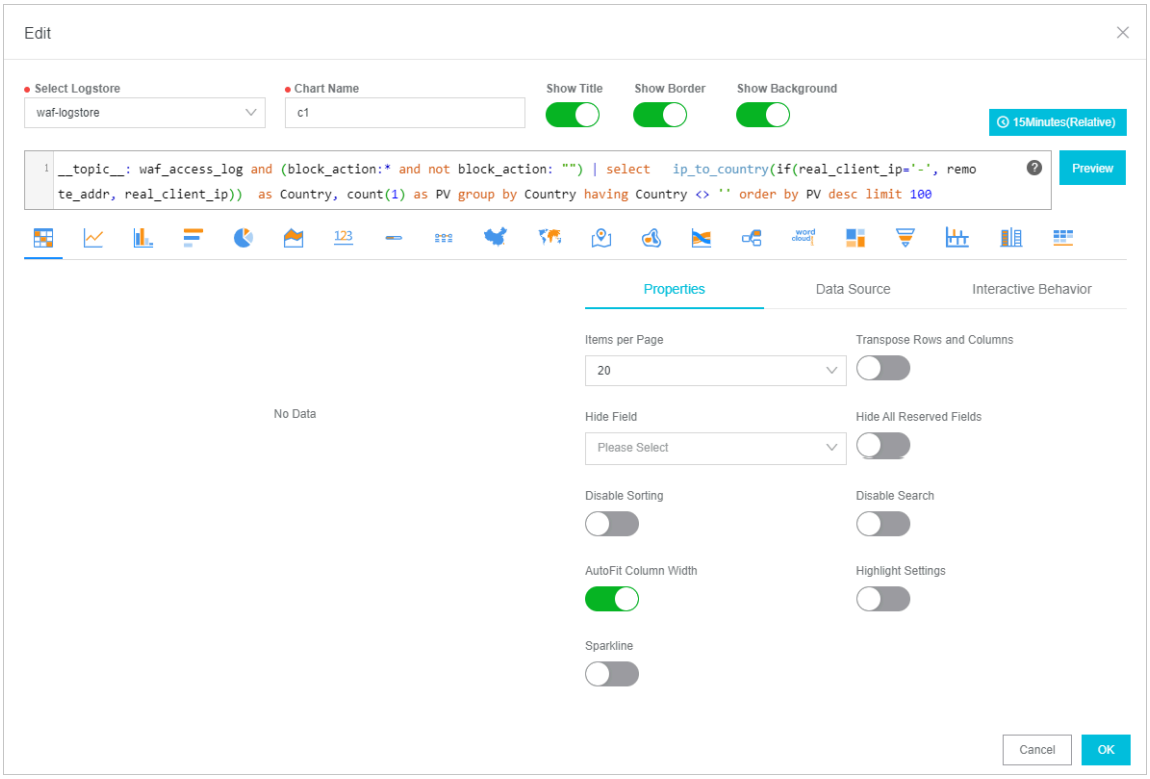
3. Edit a chart.

a) Find the chart to be edited, move the pointer over the icon in the upper-right corner of the chart, and click Edit.

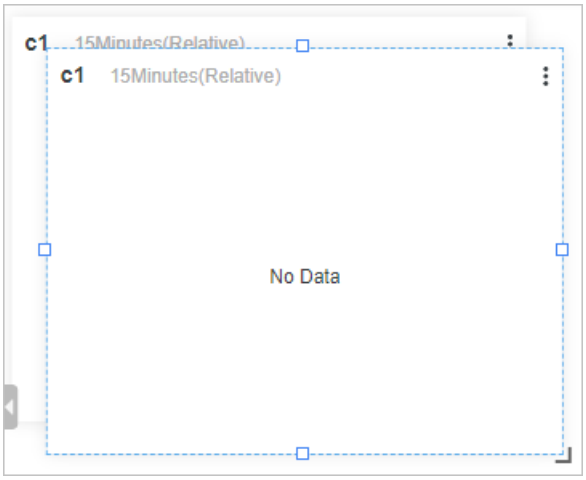
b) On the Edit page, modify the chart configurations, such as Chart Name, SQL statements, relative data collection period, and chart type. Click OK.

Note:
If you have modified the SQL statements, you must click Preview before you click OK. This operation allows the system to check the statement validity. If the SQL statements are invalid, an error message appears, and the OK

button becomes unavailable. You can click OK only after you make sure the statements are valid.



4. **Copy a chart to create a new chart.**
   a) **Find the chart to be copied. Move the pointer over the ⋮ icon in the upper-right corner of the chart, and click Copy.**
   After you copy a chart, an identical chart appears.



   b) **Drag the new chart and drop it at an appropriate position on the dashboard.**
   c) **Modify the information of the new chart, including the chart name and the SQL statements.**

5. Repeat steps 3 and 4 to create more charts for diversified data visualization and alert configurations.

What's next

Step 3: Configure a log alert

3.4 Step 3: Configure a log alert

After you create a log analysis dashboard, you can configure log alerts on the dashboard. You must associate an alert with an existing log chart and set the alert trigger conditions based on the parameters in the associated chart. You can customize the alert message template.

Prerequisites

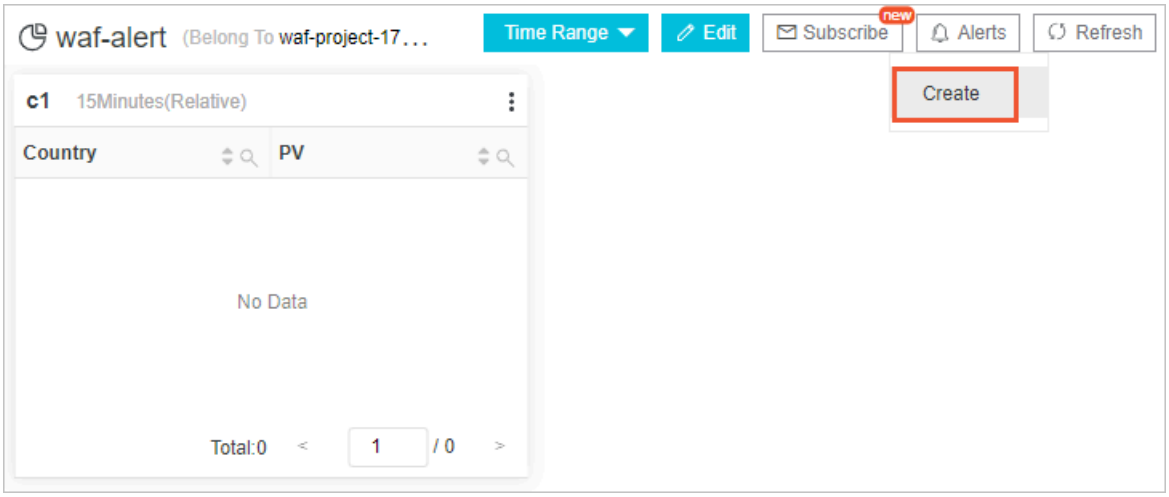
You have created a log analysis dashboard. For more information, see *Step 1: create a WAF log analysis dashboard*.

Context

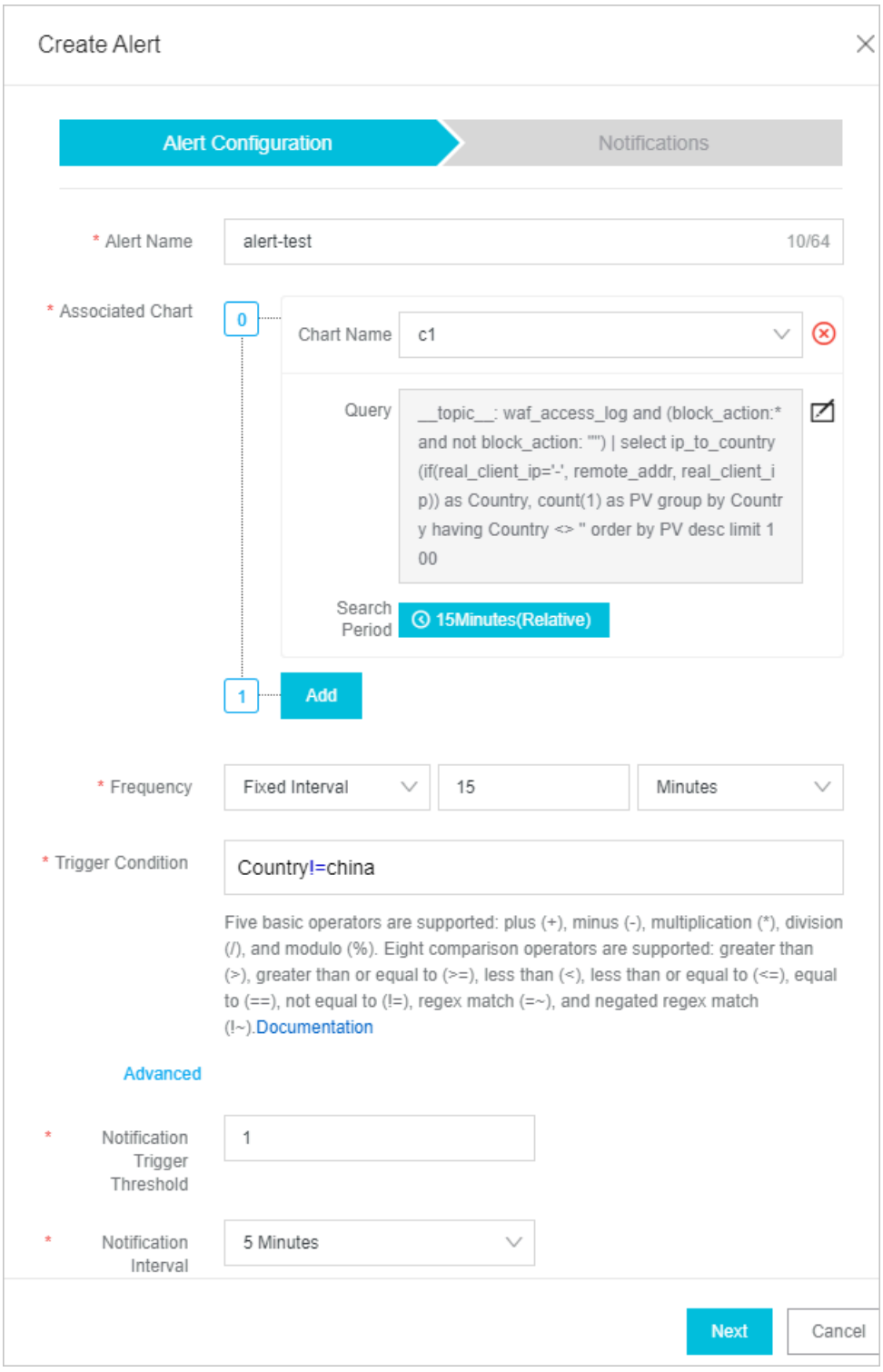
This practice provides 13 default alert configuration examples. For more information, see *WAF log charts and alert configuration examples*. We recommend that you learn the chart configuration steps before you create a chart based on the examples and configure alerts and notification methods during chart creation. For more information about chart configuration, see *Step 2: configure log charts*.

Procedure

1. Enter the customized WAF log analysis dashboard.
2. In the upper-right corner of the dashboard, choose Alerts > Create.



3. In the Create Alert pane, set the parameters in Alert Configuration, and click Next.



Parameter	Description
Alert Name	The name of the alert. The name must be 1 to 64 characters in length.

Parameter	Description
Associated Chart	The chart with which the alert is associated. The Search Period parameter specifies the time range of log data that the server reads for running a data query task. You can select either a relative time period or a time frame. For example, if you set Search Period to 15 minutes (relative) and start the query at 14:30:06, the server reads the log data that was written from 14:15:06 to 14:30:06. If you set Search Period to 15 minutes (time frame) and start the query at 14:30:06, the server reads the log data that was written from 14:15:00 to 14:30:00. To associate the alert with multiple charts, click Add and configure new charts. You can add up to three charts. The number before the chart name is the sequence number of the chart in alert configuration. You can use the sequence number to associate a chart with a conditional expression in the trigger condition.
Frequency	The time interval at which the server checks log data according to the alert configuration. Note: Currently, the server samples and checks only the first 100 data entries each time the specified time interval arrives.

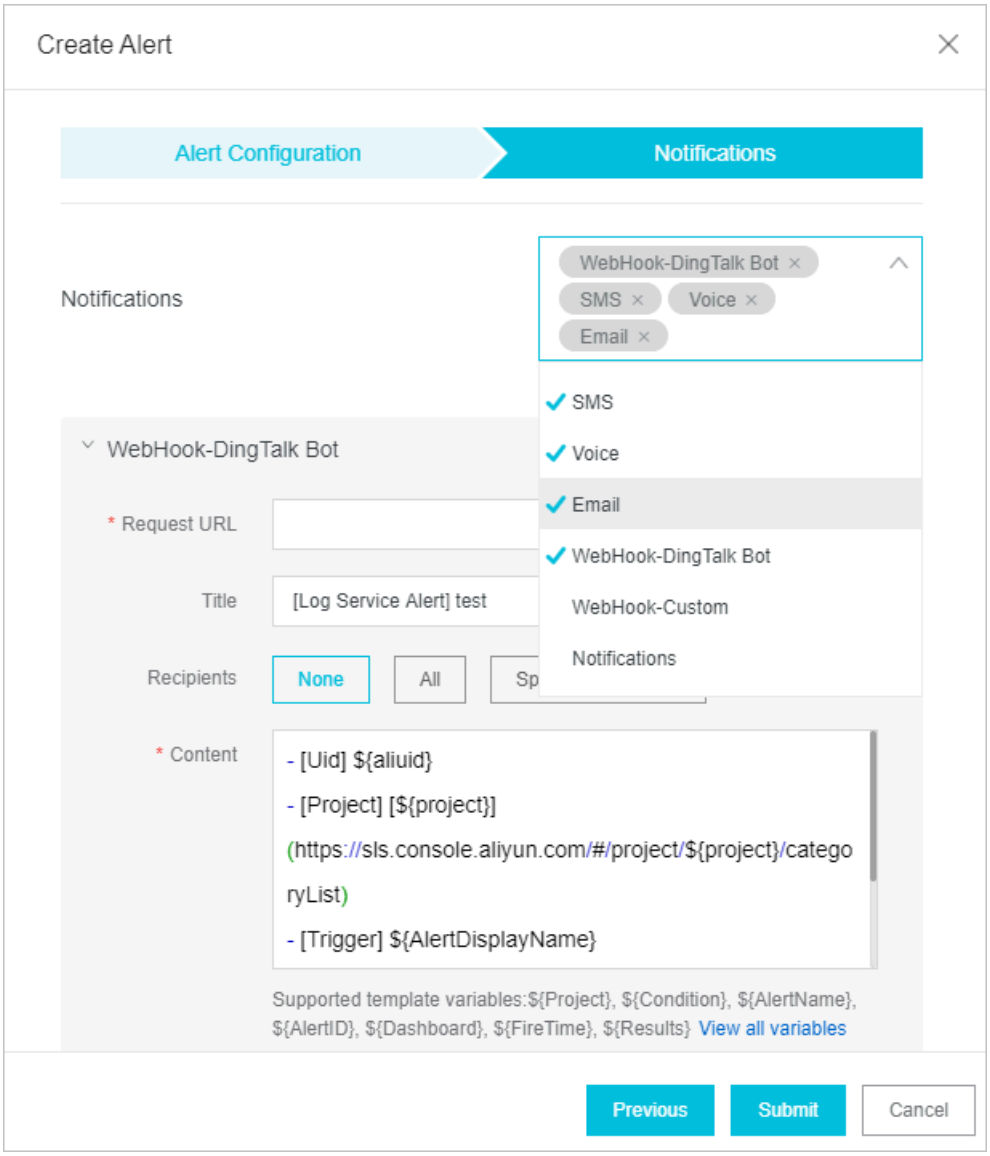
Parameter	Description
Trigger Condition	The conditional expression that determines whether the alert is triggered. When the condition is met, the server sends an alert notification based on the specified Frequency and Notification Interval. By default, the charts are numbered from 0. In a trigger condition, you can use `$0` to indicate the first chart. For example, you can set a trigger condition to `$0.domainnum >=10`, which indicates that an alert is triggered if the `domainnum` parameter in the first chart is greater than or equal to 10. If two conditions are jointed with two consecutive ampersands (`&&`), both the conditions must be met to trigger the alert. If two conditions are jointed with two consecutive vertical bars (`||`), either of the condition can trigger the alert. **Note:** For more syntaxes of conditional expressions, see *#unique_21*.
Advanced	

Parameter	Description
Notification Trigger Threshold	The threshold for sending an alert notification based on the specified notification interval when the cumulative number of times that the trigger condition is met exceeds this threshold. If the trigger condition is not met, the overall count does not change. The default value of Notification Trigger Threshold is 1. That is, each time the specified trigger condition is met, the server checks whether the specified notification interval arrives. You can also specify this parameter to enable the server to send an alert notification after the trigger condition is met multiple times. For example, if you set this parameter to 100, the server checks whether the specified notification interval arrives only after the trigger condition is met 100 times. If the specified notification trigger threshold is reached and the specified notification interval arrives, the server sends an alert notification. Then, the overall count is reset. If the server fails to check log data due to exceptions such as a network failure, the overall count does not change.
Notification Interval	The time interval at which the server sends an alert notification. If the trigger condition is met several times that exceed the specified notification trigger threshold and the specified notification interval arrives, the server sends an alert notification. If you set this parameter to 5 minutes, you can receive up to one alert notification every 5 minutes for the alert. The default value is No Interval. **Note:** By setting Notification Trigger Threshold and Notification Interval, you can control the number of alert notifications that you receive.

Note:

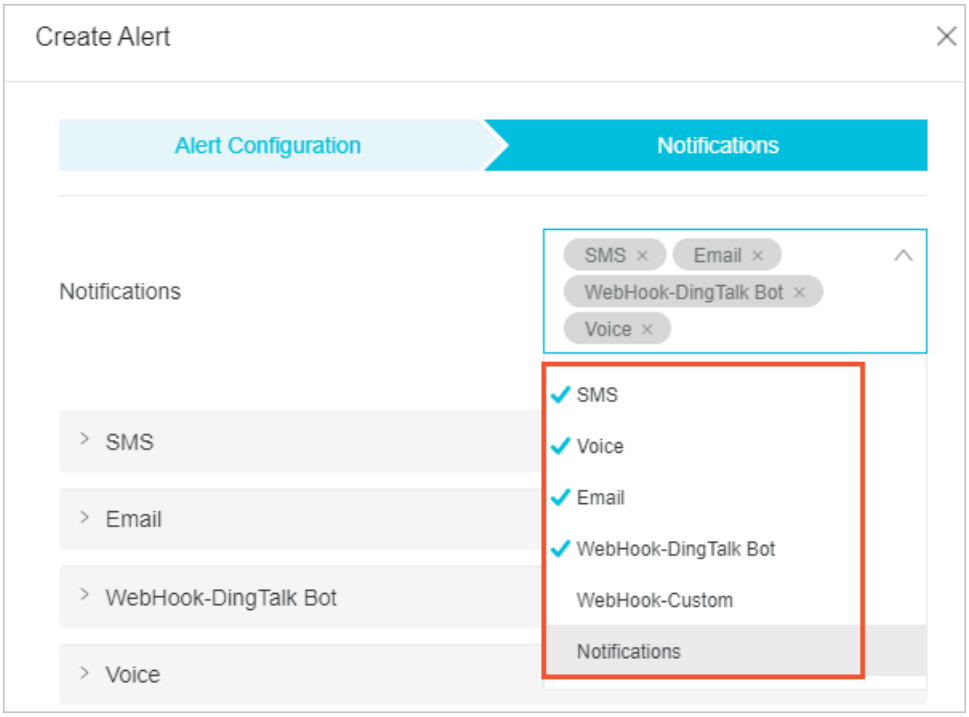
After you specify Notification Trigger Threshold, Notification Interval, and Frequency, the system checks whether the trigger conditions are met at the specified frequency and sends notifications if Notification Trigger Threshold is exceeded within a Notification Interval.

4. **In the Create Alert pane, complete the settings for Notifications, and click Submit.**

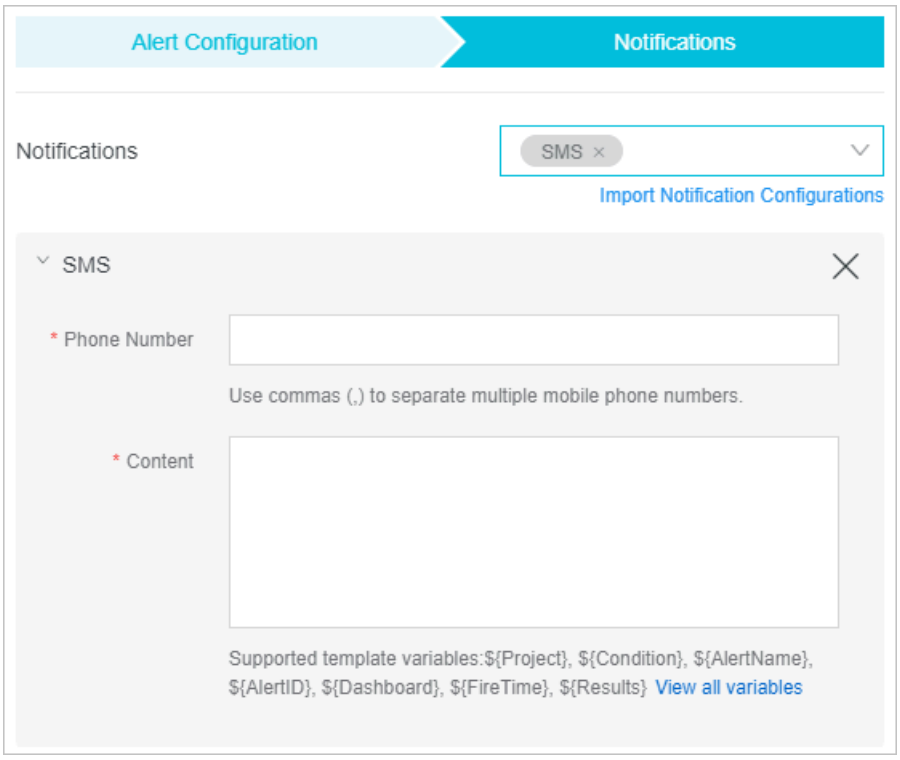


Log Service supports multiple common alert notification methods, such as SMS, Voice, Email, and WebHook-DingTalk Bot. You must select a notification method

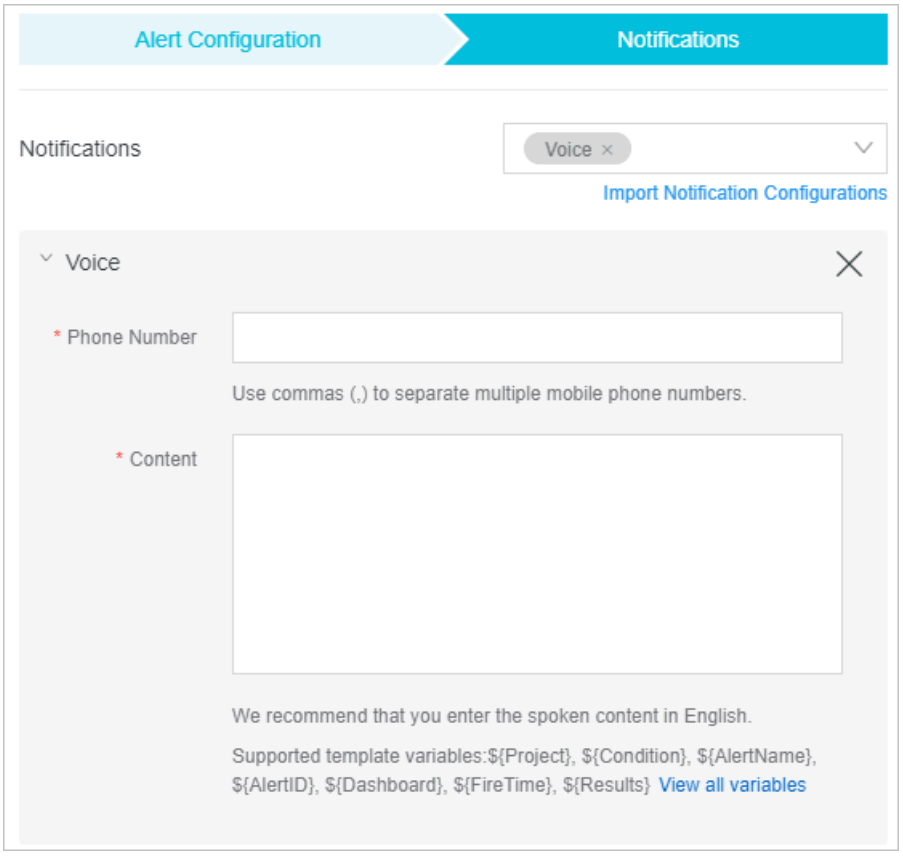
on the right of Notifications and complete the configuration. You can select and configure multiple notification methods.



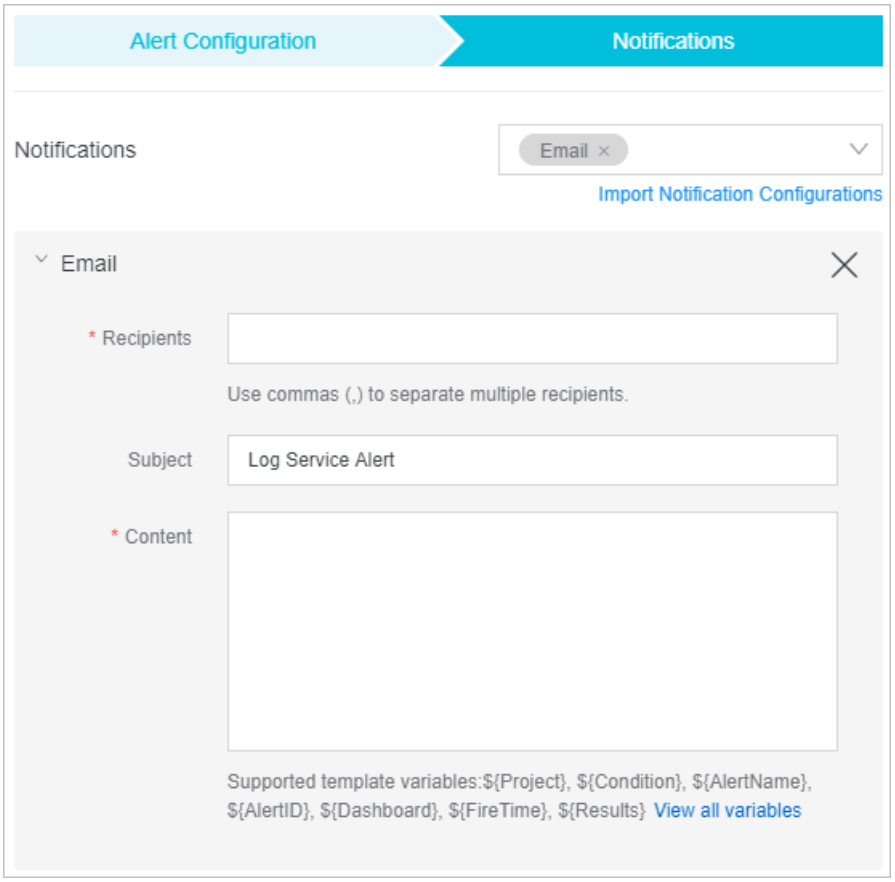
- SMS: Set Phone Number to receive alerts and Content of the notification. You can specify variables to be included in the content. Click View all variables to view the meaning of each variable.



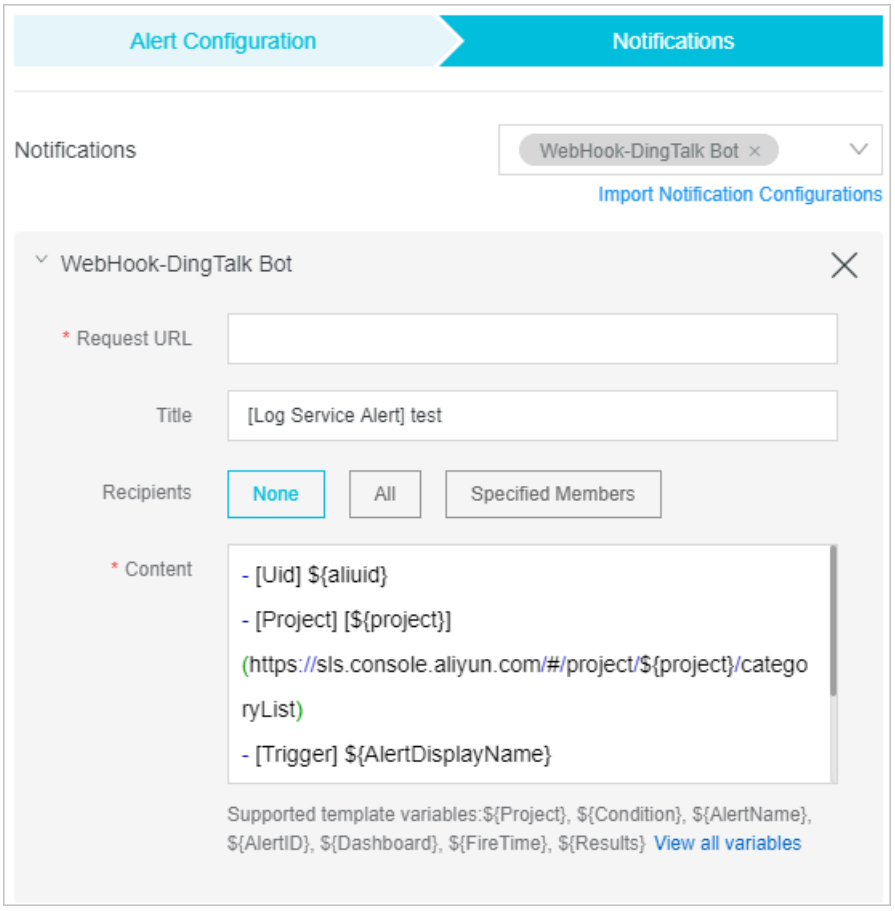
- Voice: Set Phone Number to receive alerts and Content of the notification.



- **Email: Set Recipients email addresses, Subject, and Content.**



- **WebHook-DingTalk Bot: Set Request URL to the webhook URL of the DingTalk bot to receive alerts, and specify Content.**



5. **Repeat steps 2 to 4 to create and configure more alerts.**

3.5 WAF log charts and alert configuration examples

This practice provides 13 examples of alert configuration based on log query and analysis in WAF. You can refer to the SQL statement templates in this topic to configure charts on a WAF log dashboard and configure alerts based on the suggested alert parameters.

Instructions

To configure alerts based on the examples, you must create a WAF log dashboard. For more information, see *Step 1: create a WAF log analysis dashboard*.

- **For more information about how to configure charts on a dashboard, see** *Step 2: configure log charts*.
- **For more information about how to configure alerts on a dashboard, see** *Step 3: Configure a log alert* .

This topic provides the following 13 alert configuration examples.

No.	Alert
1	*Abnormal percentage of 4xx status codes*
2	*Abnormal percentage of 5xx status codes*
3	*Abnormal query rate*
4	*Abrupt increase in query rate*
5	*Abrupt decrease in query rate*
6	*Requests blocked by HTTP ACL policy in the last five minutes*
7	*Requests blocked by web application protection in the last five minutes*
8	*Requests blocked by HTTP flood protection in the last five minutes*
9	*Requests blocked by anti-scan rules in the last five minutes*
10	*Attacks from a single IP address*
11	*Number of domains attacked by a single IP address*
12	*Average delay in the last five minutes*
13	*Abrupt decrease in query rate from a single user*

Abnormal percentage of 4xx status codes

SQL statement template

```
user_id:11111111110000 and not
real_client_ip:1.1.1.1|select user_id,host as "Domain",Rate_2XX as
"2xx codes percentage",Rate_3XX as "3xx codes percentage",Rate_4XX as
 "4xx codes percentage",Rate_5XX
as "5xx codes percentage",countall as
"aveQPS",status_2XX,status_3XX,status_4XX,status_5XX,countall
from(select user_id,host,round(round(status_2XX*1.0000/countall,4)*100
,2) as
Rate_2XX,round(round(status_3XX*1.0000/countall,4)*100,2) as Rate_3XX,
round(round

(status_4XX*1.0000/countall,4)*100,2) as
Rate_4XX,round(round(status_5XX*1.0000/countall,4)*100,2) as Rate_5XX,
status_2XX,status_3XX,status_4XX,status_5XX,countall
from(select user_id,

host,count_if(status>=200 and status<300) as
status_2XX,count_if(status>=300 and status<400) as
status_3XX,count_if(status>=400 and status<500 and status<>444 and
status<>405 ) as status_4XX,count_if(status>=500 and

status<600) as
```

```
status_5XX,COUNT(*) as countall group by host,user_id)) where
countall>120 order by Rate_4XX DESC  limit 5
```

Suggested parameter configuration for the alert

The chart contains the following parameters: `aveQPS` (request rate of the domain), `2xx codes percentage`, `3xx codes percentage`, `4xx codes percentage`, and `5xx codes percentage`. To show status codes changes caused by system workloads instead of external reasons, 444 and 405 codes triggered by HTTP flood attacks and web attacks blocked by WAF are not included as `4xx codes`. You can select one or more of these parameters to configure alerts. For example, `aveQPS>10 && 2xx codes percentage<60` indicates that the request rate of the specified domain name is higher than 10 QPS and the percentage of 2xx status codes is less than 60% during the specified period. The suggested parameters are as follows:

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition: `$0.countall>3000&& $0.4xx codes percentage>80`**
- **Notification Trigger Threshold: 2**
- **Notification Interval: 10 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0]. Domain}
- Product:WAF
- Total number of requests in the last five minutes:${Results[0].
RawResults[0].countall}
- 2xx codes percentage:${Results[0].RawResults[0].2xx codes
percentage} %
- 3xx codes percentage:${Results[0].RawResults[0].3xx codes
percentage} %
- 4xx codes percentage:${Results[0].RawResults[0].4xx codes
percentage} %
- 5xx codes percentage:${Results[0].RawResults[0].5xx codes
percentage} %
```

Abnormal percentage of 5xx status codes

SQL statement template

```
user_id:11111111110000 and not
real_client_ip:1.1.1.1|select user_id,host as "Domain",Rate_2XX as
"2xx codes percentage",Rate_3XX as "3xx codes percentage",Rate_4XX as
 "4xx codes percentage",Rate_5XX
as "5xx codes percentage",countall as "Requests in specified relative
time period",status_2XX,status_3XX,status_4XX,status_5XX,countall
from(select user_id,host,round(round(status_2XX*1.0000/countall,4)*100
,2) as
```

```
Rate_2XX,round(round(status_3XX*1.0000/countall,4)*100,2) as Rate_3XX,
round(round

(status_4XX*1.0000/countall,4)*100,2) as
Rate_4XX,round(round(status_5XX*1.0000/countall,4)*100,2) as
Rate_5XX,status_2XX,status_3XX,status_4XX,status_5XX,countall from(
select
user_id,

host,count_if(status>=200 and status<300) as
status_2XX,count_if(status>=300 and status<400) as
status_3XX,count_if(status>=400 and status<500) as
status_4XX,count_if(status>=500 and

status<600) as
status_5XX,COUNT(*) as countall group by host,user_id)) where
countall>120 order by Rate_5XX DESC  limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition:** `$0.countall>3000&& $0.5xx codes percentage>80`
- **Notification Trigger Threshold: 2**
- **Notification Interval: 10 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0]. Domain}
- Product:WAF
- Total number of requests in the last five minutes:${Results[0].
RawResults[0].countall}
- 2xx codes percentage:${Results[0].RawResults[0].2xx codes
percentage%
- 3xx codes percentage:${Results[0].RawResults[0].3xx codes
percentage} %
- 4xx codes percentage:${Results[0].RawResults[0].4xx codes
percentage} %
- 5xx codes percentage:${Results[0].RawResults[0].5xx codes
percentage} %
```

Abnormal query rate

SQL statement template

```
user_id: 11111111110000 and not
real_client_ip:1.1.1.1|select
user_id,host,Rate_2XX,Rate_3XX,Rate_4XX,Rate_5XX,countall/60 as
"aveQPS",status_2XX,status_3XX,status_4XX,status_5XX,countall
from(select user_id,host,round(round(status_2XX*1.0000/countall,4)*100
,2) as Rate_2XX,round(round(status_3XX*1.0000/countall,4)*100,2)
as Rate_3XX, round(round

(status_4XX*1.0000/countall,4)*100,2) as
Rate_4XX,round(round(status_5XX*1.0000/countall,4)*100,2) as
```

```
Rate_5XX,status_2XX,status_3XX,status_4XX,status_5XX,countall from(
select
user_id,

host,count_if(status>=200 and status<300) as
status_2XX,count_if(status>=300 and status<400) as
status_3XX,count_if(status>=400 and status<500 and status<>444 and
status<>405 ) as status_4XX,count_if(status>=500 and

status<600) as
status_5XX,COUNT(*) as countall group by host,user_id)) where
countall>120 order by aveQPS DESC  limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 1 minute**
- **Frequency: 1 minute**
- **Trigger Condition:** `$0.aveQPS>=50`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF
- Average query rate in the past 1 minute:${Results[0].RawResults[0
].aveQPS}
- Status code 2xx percentage:${Results[0].RawResults[0].Rate_2XX}%
- Status code 3xx percentage:${Results[0].RawResults[0].Rate_3XX}%
- Status code 4xx percentage:${Results[0].RawResults[0].Rate_4XX}%
- Status code 5xx percentage:${Results[0].RawResults[0].Rate_5XX}%
```

Abrupt increase in query rate

SQL statement template

```
user_id: 11111111110000 |select
t1.user_id,t1.now1mQPS,t1.past1mQPS,in_ratio,t1.host,t2.Rate_2XX,
Rate_3XX,Rate_4XX,Rate_5XX,aveQPS
from (

 (

 SELECT
user_id,round(c[1]/60,0) as now1mQPS,round(c[2]/60,0) as past1mQPS,
round(round(c[1]/60,0)/round(c[2]/60,0)*100-100,0) as in_ratio ,host
from

       (SELECT
compare(t, 60) as c,host, user_id from

           (SELECT
COUNT(*) as t,host,user_id from log GROUP by host, user_id )  GROUP by
 host, user_id) where c[3] >1.1
and (c[1]>180 or c[2]>180
```

```
        )

  )t1

           join

  (select
user_id,host,Rate_2XX,Rate_3XX,Rate_4XX,Rate_5XX,countall/60 as
"aveQPS",status_2XX,status_3XX,status_4XX,status_5XX,countall from



     (select
user_id,host,round(round(status_2XX*1.0000/countall,4)*100,2) as
Rate_2XX,round(round(status_3XX*1.0000/countall,4)*100,2) as Rate_3XX,
round(round(status_4XX*1.0000/countall,4)*100,2) as
Rate_4XX,round(round(status_5XX*1.0000/countall,4)*100,2) as
Rate_5XX,status_2XX,status_3XX,status_4XX,status_5XX,countall from

        (select
user_id, host,count_if(status>=200 and status<300) as
status_2XX,count_if(status>=300 and status<400) as
status_3XX,count_if(status>=400 and status<500 and status<>444 and
status<>405 ) as status_4XX,count_if(status>=500 and status<600) as
status_5XX,COUNT(*) as countall from log group by host,user_id)

     ) where  countall>1

   )t2

     on t1.host=t2.host) order by in_ratio DESC
limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 1 minute**
- **Frequency: 1 minute**
- **Trigger Condition:** `$0.now1mqps>50&& $0.in_ratio>300`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF
- Average query rate in the past 1 minute:${Results[0].RawResults[0
].now1mqps}
- Abrupt increase ratio of query rate:${Results[0].RawResults[0].
in_ratio}%
- Status code 2xx percentage:${Results[0].RawResults[0].rate_2xx}%
- Status code 3xx percentage:${Results[0].RawResults[0].Rate_3XX}%
- Status code 4xx percentage:${Results[0].RawResults[0].Rate_4XX}%
```

```
- Status code 5xx percentage:${Results[0].RawResults[0].Rate_5XX}%
```

Abrupt decrease in query rate

SQL statement template

```
user_id: 11111111110000 |select
t1.user_id,t1.now1mQPS,t1.past1mQPS,de_ratio,t1.host,t2.Rate_2XX,
Rate_3XX,Rate_4XX,Rate_5XX,aveQPS
from (

 (

 SELECT
user_id,round(c[1]/60,0) as now1mQPS,round(c[2]/60,0) as past1mQPS,
round(100-round(c[1]/60,0)/round(c[2]/60,0)*100,2) as de_ratio,host
from

(SELECT compare(t, 60) as c,host, user_id from

    (SELECT
COUNT(*) as t,host,user_id from log GROUP by host, user_id )  GROUP by
 host, user_id ) where c[3] <0.9
and (c[1]>180 or c[2]>180

        )

  )t1

           join

  (select
user_id,host,Rate_2XX,Rate_3XX,Rate_4XX,Rate_5XX,countall/60 as
"aveQPS",status_2XX,status_3XX,status_4XX,status_5XX,countall from

     (select
user_id,host,round(round(status_2XX*1.0000/countall,4)*100,2) as
Rate_2XX,round(round(status_3XX*1.0000/countall,4)*100,2) as

Rate_3XX,
round(round(status_4XX*1.0000/countall,4)*100,2) as
Rate_4XX,round(round(status_5XX*1.0000/countall,4)*100,2) as

Rate_5XX,status_2XX,status_3XX,status_4XX,status_5XX,countall
from

        (select
user_id, host,count_if(status>=200 and status<300) as
status_2XX,count_if(status>=300 and status<400) as status_3XX,count_if

(status>=400 and status<500 and status<>444
and status<>405 ) as status_4XX,count_if(status>=500 and
status<600) as status_5XX,COUNT(*) as countall from log group by host,
user_id)

     ) where  countall>1

)t2 on
t1.host=t2.host) order by de_ratio DESC limit 5
```

Suggested parameter configuration for the alert

The chart contains the following parameters: `now1mpqs` (average query rate of the current minute), `past1mqps` (average query rate of the last minute), `de_ratio` (decrease ratio of query rate), and `host`. You can select these parameters to configure alerts.

- **Search Period: 1 minute**
- **Frequency: 1 minute**
- **Trigger Condition: `$0.now1mqps>10&& $0.de_ratio>50`**
- **Notification Trigger Threshold: 2**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF (International)
- Average query rate in the past 1 minute:${Results[0].RawResults[0
].now1mqps}
- Abrupt decrease ratio of query rate:${Results[0].RawResults[0].
de_ratio}%
- Status code 2xx percentage:${Results[0].RawResults[0].rate_2xx}%
- Status code 3xx percentage:${Results[0].RawResults[0].Rate_3XX}%
- Status code 4xx percentage:${Results[0].RawResults[0].Rate_4XX}%
- Status code 5xx percentage:${Results[0].RawResults[0].Rate_5XX}%
```

Requests blocked by HTTP ACL policy in the last five minutes

SQL statement template

```
User_id:
11111111110000 |select user_id,host,count_if(block_action='antiscan')
as "Requests blocked by anti-scan rules",count_if(block_action='acl')
as "Requests blocked by HTTP ACL policy",count_if(aliwaf_action='block
')
as "Requests blocked by web application protection",count_if(cc_action
='close') as
"Requests blocked by HTTP flood protection",count_if(block_action='acl
' or
aliwaf_action='block' or cc_action='close' or block_action='antiscan
') as
totalblock  group by host,user_id having
("Requests blocked by HTTP ACL policy" >=0 and "Requests blocked by
web application protection" >=0 and "Requests blocked by HTTP flood
protection">=0
and totalblock>10) order by "Requests blocked by HTTP ACL policy"
DESC limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**

- **Trigger Condition:**`$0.totalblock>=500&&($0.Requests blocked by HTTP ACL policy>=500)`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF
- Total requests blocked in the last five minutes:${Results[0].
RawResults[0].totalblock}
- Requests blocked by HTTP ACL policy:${Results[0].RawResults[0].
Requests blocked by HTTP ACL policy}
- Requests blocked by web application protection:${Results[0].
RawResults[0].Requests blocked by web application protection}
- Requests blocked by HTTP flood protection:${Results[0].RawResults[
0].Requests blocked by HTTP flood protection}
- Requests blocked by anti-scan rules:${Results[0].RawResults[0].
Requests blocked by anti-scan rules}
```

Requests blocked by web application protection in the last five minutes

SQL statement template

```
user_id:11111111110000
|select user_id,host,count_if(block_action='antiscan') as "Requests
blocked by anti-scan rules",count_if(block_action='acl')
as "Requests blocked by HTTP ACL policy",count_if(aliwaf_action='block
')
as "Requests blocked by web application protection",count_if(cc_action
='close') as
"Requests blocked by HTTP flood protection",count_if(block_action='acl
' or
aliwaf_action='block' or cc_action='close' or block_action='antiscan
') as
totalblock  group by host,user_id having
("Requests blocked by HTTP ACL policy" >=0 and "Requests blocked by
web application protection" >=0 and "Requests blocked by HTTP flood
protection">=0
and totalblock>10) order by "Requests blocked by web application
protection"  DESC limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition:** `$0.totalblock>=500&&($0.Requests blocked by web application protection>=500)`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**

- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF
- Total requests blocked in the last 5 minutes:${Results[0].
RawResults[0].totalblock}
- Requests blocked by HTTP ACL policy:${Results[0].RawResults[0].
Requests blocked by HTTP ACL policy}
- Requests blocked by web application protection:${Results[0].
RawResults[0].Requests blocked by web application protection}
- Requests blocked by HTTP flood protection:${Results[0].RawResults[
0].Requests blocked by HTTP flood protection}
- Requests blocked by anti-scan rules:${Results[0].RawResults[0].
Requests blocked by anti-scan rules}
```

Requests blocked by HTTP flood protection in the last five minutes

SQL statement template

```
user_id:
11111111110000 |select user_id,host,count_if(block_action='antiscan')
as "Requests blocked by anti-scan rules",count_if(block_action='acl')
as "Requests blocked by HTTP ACL policy",count_if(aliwaf_action='block
')
as "Requests blocked by web application protection",count_if(cc_action
='close') as
"Requests blocked by HTTP flood protection",count_if(block_action='acl
' or
aliwaf_action='block' or cc_action='close' or block_action='antiscan
') as
totalblock  group by host,user_id having
("Requests blocked by HTTP ACL policy" >=0 and "Requests blocked by
web application protection" >=0 and "Requests blocked by HTTP flood
protection">=0
and totalblock>10) order by "Requests blocked by HTTP flood protection
"  DESC limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition:** `$0.totalblock>=500&&($0.Requests blocked by HTTP flood protection>=500)`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF
```

```
- Total requests blocked in the last five minutes:${Results[0].
RawResults[0].totalblock}
- Requests blocked by HTTP ACL policy:${Results[0].RawResults[0].
Requests blocked by HTTP ACL policy}
- Requests blocked by web application protection:${Results[0].
RawResults[0].Requests blocked by web application protection}
- Requests blocked by HTTP flood protection:${Results[0].RawResults[
0].Requests blocked by HTTP flood protection}
- Requests blocked by anti-scan rules:${Results[0].RawResults[0].
Requests blocked by anti-scan rules}
```

Requests blocked by anti-scan rules in the last five minutes

SQL statement template

```
user_id:
11111111110000 |select user_id,host,count_if(block_action='antiscan')
as "Requests blocked by anti-scan rules",count_if(block_action='acl')
as "Requests blocked by HTTP ACL policy",count_if(aliwaf_action='block
')
as "Requests blocked by web application protection",count_if(cc_action
='close') as
"Requests blocked by HTTP flood protection",count_if(block_action='acl
' or
aliwaf_action='block' or cc_action='close' or block_action='antiscan
') as
totalblock  group by host,user_id having
("Requests blocked by HTTP ACL policy" >=0 and "Requests blocked by
web application protection" >=0 and "Requests blocked by HTTP flood
protection">=0
and totalblock>10) order by "Requests blocked by anti-scan rules"
DESC limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition:** `$0.totalblock>=500&&($0. Requests blocked by anti-scan rules>=500)`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF (International)
- Total requests blocked in the last 5 minutes:${Results[0].
RawResults[0].totalblock}
- Requests blocked by HTTP ACL policy:${Results[0].RawResults[0].
Requests blocked by HTTP ACL policy}
- Requests blocked by web application protection:${Results[0].
RawResults[0].Requests blocked by web application protection}
- Requests blocked by HTTP flood protection:${Results[0].RawResults[
0].Requests blocked by HTTP flood protection}
```

```
- Requests blocked by anti-scan rules:${Results[0].RawResults[0].
Requests blocked by anti-scan rules}
```

Attacks from a single IP address

SQL statement template

```
user_id:
11111111110000 |select user_id,real_client_ip,concat('Requests blocked
 by HTTP ACL policy:',cast(aclblock as
varchar(10)),'  ','Requests blocked by web application protection:',
cast(wafblock as varchar(10)),'
','Requests blocked by HTTP flood protection:',cast(aclblock as
varchar(10))) as
blockNum,totalblock,allRequest from (select user_id,real_client_ip,
count_if(block_action='acl')
as aclblock,count_if(aliwaf_action='block') as
wafblock,count_if(cc_action='close') as ccblock,count_if(block_action
='acl' or
aliwaf_action='block' or cc_action='close') as totalblock,COUNT(*) as
allRequest from log group by user_id,real_client_ip having totalblock>
1
order by totalblock DESC  limit 5)
```

Suggested parameter configuration for the alert

The chart contains the following parameters: `real_client_ip`, `blockNum` **(including** `Requests blocked by HTTP ACL policy`, `Requests blocked by web application protection`, **and** `Requests blocked by HTTP flood protection`, `totalblock` **(total number of blocked requests), and** `allRequest` **(total number of requests). You can select the parameters to configure alerts.**

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition:** `$0.totalblock >=500`
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Product:WAF
- Top 3 attack source IP addresses in the last 5 minutes:
- ${Results[0].RawResults[0].real_client_ip}  (${Results[0].
RawResults[0].blockNum})
- ${Results[0].RawResults[1].real_client_ip}  (${Results[0].
RawResults[1].blockNum})
```

```
-${Results[0].RawResults[2].real_client_ip}  (${Results[0].
RawResults[2].blockNum})
```

Number of domains attacked by a single IP address

SQL statement template

```
user_id:
11111111110000 and not
upstream_status:504 and not upstream_addr:- and request_time_msec <
5000 and
upstream_status:200 and not ua_browser:bot |SELECT user_id,host,
upstream_time,request_time,ssl_handshake,requestnum
from (select user_id,host,round(avg(upstream_response_time),2)*1000 as
upstream_time,round(avg(request_time_msec),2) as
request_time,round(avg(ssl_handshake_time)*1000,2) as ssl_handshake,
COUNT(*) as
requestnum from log group by host,user_id) where requestnum>30 order
by
request_time DESC limit 5
```

Suggested parameter configuration for the alert

The chart contains the following parameters: `real_client_ip` (attacker IP address), `totalblock` (total number of blocked requests), and `domainnum` (number of domains attacked by this IP address). You can select one or more of these parameters to configure alerts. For example, `totalblock>500&& domainnum>5` indicates that the total number of attacks launched by an IP address reaches 500 and the number of attacked domains exceeds 5.

- **Search Period: 5 minutes**
- **Frequency: 1 minute**
- **Trigger Condition: `$0.domainnum>=10`**
- **Notification Trigger Threshold: 1**
- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Product:WAF
- Attacker IP:${Results[0].RawResults[0].real_client_ip}
- Number of attacked domains:${Results[0].RawResults[0].domainnum}
- Total requests blocked in the last 5 minutes:${Results[0].
RawResults[0].totalblock}
```

```
- Please handle the alert in a timely manner.
```

Average delay in the last five minutes

SQL statement template

```
user_id:
11111111110000 and and not upstream_status:504 and not upstream_addr
:- and
request_time_msec < 5000 and upstream_status:200 and not ua_browser:
bot|SELECT
user_id,host,upstream_time,request_time,ssl_handshake,requestnum from
 (select user_id,host,round(avg(upstream_response_time),2)*1000
as upstream_time,round(avg(request_time_msec),2) as
request_time,round(avg(ssl_handshake_time)*1000,2) as ssl_handshake,
COUNT(*) as
requestnum from log group by host,user_id) where requestnum>30 order
by
request_time DESC limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 5 minutes**
- **Frequency: 5 minutes**
- **Trigger Condition:** `$0.request_time>1000&& $0.requestnum>30`
- **Notification Trigger Threshold: 2**
- **Notification Interval: 10 minutes**
- **Content**

```
- [Time]:${FireTime}
- [Uid]:${Results[0].RawResults[0].user_id}
- Domain:${Results[0].RawResults[0].host}
- Product:WAF (International)
- [Trigger condition]:${condition}
- Top 3 domains with the longest delay in the last 5 minutes (unit:
millisecond)
- Host1:${Results[0].RawResults[0].host} Delay_time:${Results[0].
RawResults[0].upstream_time}
- Host2:${Results[0].RawResults[1].host} Delay_time:${Results[0].
RawResults[1].upstream_time}
- Host3:${Results[0].RawResults[2].host} Delay_time:${Results[0].
RawResults[2].upstream_time}
```

Abrupt decrease in query rate from a single user

SQL statement template

```
user_id: 11111111110000 |select
t1.user_id,t1.now1mQPS,t1.past1mQPS,de_ratio,t2.Rate_2XX,Rate_3XX,
Rate_4XX,Rate_5XX,aveQPS
from (

 (

 SELECT
```

```
user_id,round(c[1]/60,0) as now1mQPS,round(c[2]/60,0) as past1mQPS,
round(100-round(c[1]/60,0)/round(c[2]/60,0)*100,2) as de_ratio from

(SELECT compare(t, 60) as c, user_id from

    (SELECT
COUNT(*) as t,user_id from log GROUP by user_id )  GROUP by user_id )
where c[3] <0.9 and
(c[1]>180 or c[2]>180

        )

  )t1

           join

  (select
user_id,Rate_2XX,Rate_3XX,Rate_4XX,Rate_5XX,countall/60 as
"aveQPS",status_2XX,status_3XX,status_4XX,status_5XX,countall from



     (select
user_id,round(round(status_2XX*1.0000/countall,4)*100,2) as
Rate_2XX,round(round(status_3XX*1.0000/countall,4)*100,2) as



Rate_3XX,
round(round(status_4XX*1.0000/countall,4)*100,2) as
Rate_4XX,round(round(status_5XX*1.0000/countall,4)*100,2) as



Rate_5XX,status_2XX,status_3XX,status_4XX,status_5XX,countall
from

        (select
user_id,count_if(status>=200 and status<300) as
status_2XX,count_if(status>=300 and status<400) as status_3XX,count_if



(status>=400 and status<500 and status<>444
and status<>405 ) as status_4XX,count_if(status>=500 and
status<600) as status_5XX,COUNT(*) as countall from log group by
user_id)

     ) where  countall>0

)t2 on
t1.user_id=t2.user_id) order by de_ratio DESC limit 5
```

Suggested parameter configuration for the alert

- **Search Period: 1 minute**
- **Frequency: 1 minute**
- **Trigger Condition:** `$0.de_ratio>50&& $0.now1mqps>20`
- **Notification Trigger Threshold: 1**

- **Notification Interval: 5 minutes**
- **Content**

```
- [Time]:${FireTime}
- [UID]:${Results[0].RawResults[0].user_id}
- Product:WAF
- Average query rate in the past 1 minute:${Results[0].RawResults[0
].now1mqps}
- [Trigger condition (abrupt decrease ratio of query rate & query
rate)]:${condition}
- Abrupt decrease ratio of query rate:${Results[0].RawResults[0].
de_ratio}%
- Status code 2xx percentage:${Results[0].RawResults[0].rate_2xx}%
- Status code 3xx percentage:${Results[0].RawResults[0].Rate_3XX}%
- Status code 4xx percentage :${Results[0].RawResults[0].Rate_4XX}%
- Status code 5xx percentage:${Results[0].RawResults[0].Rate_5XX}%
```

3.6 Common monitoring metrics

This topic describes the common metrics that are used to query and analyze logs collected by Log Service in WAF. You can use these metrics to configure alerts so that you can monitor exceptions in your workload as needed. This topic also provides the recommended alert thresholds of metrics and suggestions on handling metric exceptions.

Metric	Description	Recommended threshold	Suggestion
200	The server has processed the request and returned the requested data.	Before you initialize your workloads, set the alert threshold to 90% for status code 200. You can adjust the threshold as needed.	If the percentage of code 200 is lower than the specified threshold, identify the reason. For example, this metric may have decreased because the percentage of another status code has increased.

Metric	Description	Recommended threshold	Suggestion
request_time_msec	Time period between the time when the client sends a request and the time when the client receives a response.	Set the alert thresholds based on the time required for actual service requests.	If it takes a long time to receive responses from a domain name, check the network connectivity between the client and WAF and that between WAF and the origin servers, and make sure that the origin servers respond properly.
upstream_response_time	The time period between the time when WAF sends data to the origin server and the time when WAF receives a response from the origin server.		
ssl_handshake_time	The time required for an SSL handshake between the client and WAF during an HTTPS request.		
status:302 and block_action:captcha and block_action:200	The status code indicates whether CAPTCHA is triggered. Code 302 indicates that CAPTCHA is triggered, and code 200 indicates that CAPTCHA is not triggered and the user needs to customize HTTP flood protection.	When you initialize your workloads, we recommend that you set the alert thresholds for status code percentage to a value from 5% to 10%. You can adjust the thresholds based on the traffic blocked by WAF.	· If the alert threshold is reached, find out whether the domain is under HTTP flood attacks and customize rules to block the attacks. · Check for server exceptions, for example, a large number of 5xx status codes or 4xx status codes.
200 and block_action:bot_detection	A request is blocked by data risk control.		Test the alert rule before you apply it. If you receive this alert frequently, contact us to adjust the alert threshold.

Metric	Description	Recommended threshold	Suggestion
status:404	The server cannot find the requested resources.		Query the source IP addresses that trigger the alert. · If only one IP address triggers the alert, a malicious user may have started a path traversal on your server. · If multiple IP addresses trigger the alert, check whether the server works properly and whether any files are missing.
status:405	A request is blocked by either web application protection rules or HTTP ACL policy rules.		Use the log search feature to analyze the blocked request and the rule used to block the request, and find out whether this is a false positive.
status:444	A request is blocked by custom HTTP flood protection rules.		· If the alert threshold is reached, find out whether the domain is under HTTP flood attacks and customize rules to block the attacks. · If the blocked request is not an attack but an API call, you can adjust the threshold or allow API calls on specified servers.

Metric	Description	Recommended threshold	Suggestion
status:499	After a client sends a request, the server does not return data. After the maximum wait time of the client is reached, the client disconnects, and the server returns this status code.		· Check for exceptions on the origin server, for example, slow responses and a large number of slow queries on the database. · Check whether attacks have consumed all resources on the origin server.
status:500	A request cannot be processed due to the 500 Internal Server Error.		We recommend that you check the loads and database status of the origin server.
status:502	The server is used as a gateway or a proxy and receives invalid responses from the upstream server due to a 502 Bad Gateway error. The origin server does not respond due to low quality performance of the back-to-origin network or the fact that back-to-origin requests are blocked by access control policies configured for the origin server.		· Check the back-to-origin network quality, the access control policies on the origin server, and the loads and database status of the origin server. · Check whether the origin server has blocked the back-to-origin IP address of WAF.
status:503	The service is unavailable due to overloads or maintenance needs.		Check for exceptions on the origin server.

Metric	Description	Recommended threshold	Suggestion
status:504	The server serves as a gateway or proxy and fails to receive requests from the upstream server in time. The 504 Gateway Timeout error occurred.		Possible causes include: · The server fails to respond due to overload. · The origin server does not reset after it discards requests. · The protocol-based communication fails.

3.7 Common SQL statements

This topic describes the SQL statements used to query and analyze monitoring metrics by using Log Service in WAF.

The following list describes the metrics that are commonly used to query and analyze logs in Log Service. You can click a metric to view the corresponding SQL statements. For more information about the metrics, see *Common monitoring metrics*.

- *request_time_msec*
- *upstream_response_time*
- *ssl_handshake_time*
- *200*
- *status:302 and block_action:tmd/status:200 and block_action:tmd*
- *200 and block_action:‘antifraud’*
- *status:404*
- *status:405 and aliwaf_action='block'*
- *status:405 and aliwaf_action='acl'*
- *status:444*
- *status:499*
- *status:500*
- *status:502*
- *status:503*
- *status:504*

request_time_msec

Time period between the time when the client sends a request and the time when the client receives a response.

```
* |SELECT user_id,host,round(round(request_time_cnt*1.0000/countall,4
)*100,2)
as percent FROM  (select user_id,host,count_if(request_time_msec>500)
AS request_time_cnt ,COUNT(*)  as countall from log group by user_id,
host)
group by user_id,host,percent
```

upstream_response_time

The time period between the time when WAF sends data to the origin server and the time when WAF receives a response from the origin server.

```
* |SELECT
user_id,host,round(round(upstream_response_time_cnt*1.0000/countall,4
)*100,2)
as percent FROM  (select
user_id,host,count_if(upstream_response_time>500) AS
upstream_response_time_cnt ,COUNT(*)  as countall from log group by
user_id,host) group by user_id,host,percent
```

ssl_handshake_time

The time required for an SSL handshake between the client and WAF during an HTTPS request.

```
* |SELECT
user_id,host,round(round(ssl_handshake_time_cnt*1.0000/countall,4)*100
,2) as
percent FROM  (select user_id,host,count_if(ssl_handshake_time>10) AS
ssl_handshake_time_cnt ,COUNT(*)  as countall from log group by
user_id,host) group by user_id,host,percent
```

200

The server has processed the request and returned the requested data.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round
```

```
(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as Rate_502,
round(round(status_503*1.0000/countall,4)*100,2)
as Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=444) as
status_444,count_if(status=499) as status_499,count_if(status=500) as
status_500,count_if(status=502)
as status_502,count_if(status=503) as status_503,count_if(status=504)
as
status_504,COUNT(*)  as countall from log group by user_id,host))
where  countall>120 order by Rate_200 DESC  limit 5
```

status:302 and block_action:tmd/status:200 and block_action:tmd

The status code indicates whether CAPTCHA is triggered. Code 302 indicates that CAPTCHA is triggered, and code 200 indicates that CAPTCHA is not triggered and the user needs to customize HTTP flood protection.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as Rate_444,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as Rate_503,
round(round(status_504*1.0000/countall,4)*100,2)
as
```

```
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200 and

block_action:tmd

) as status_200,count_if(status=302 and

block_action:tmd

) as
status_302,count_if(status=404) as status_404,count_if(status=405) as
status_405,count_if(status=444) as status_444,count_if(status=499) as
status_499,count_if(status=500) as status_500,count_if(status=502) as
status_502,count_if(status=503) as status_503,count_if(status=504) as
status_504,COUNT(*)  as countall from log group by user_id,host))
where  countall>120 order by Rate_200 DESC  limit 5
```

200 and block_action: ‘antifraud’

A request is blocked by data risk control.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302
, round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200 and block_action:‘
antifraud’) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=444) as
status_444,count_if(status=499) as status_499,count_if(status=500) as
status_500,count_if(status=502) as status_502,count_if(status=503) as
status_503,count_if(status=504) as status_504,COUNT(*)  as countall
from
log group by user_id,host)) where  countall>120 order by Rate_200
```

```
DESC  limit 5
```

status:404

The server cannot find the requested resources.

```
*|select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_500 as "Status code 500
percentage",Rate_502 as "Status code 502 percentage",Rate_503
as "Status code 503 percentage",Rate_504 as "Status code 504
percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_500,
status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_405,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_500,
status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=499) as
status_499,count_if(status=500) as status_500,count_if(status=502) as
status_502,count_if(status=503) as status_503,count_if(status=504) as
status_504,COUNT(*)  as countall from log group by user_id,host))
where  countall>120 order by Rate_404 DESC  limit 5
```

status:405 and aliwaf_action='block'

A request is blocked by web application protection rules.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as "aveQPS",status_200
,status_302,status_404,status_405,status_444,status_499,status_500,
status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
```

```
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as Rate_504
,status_200,status_302,status_404,status_405,status_444,status_499,
status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405 and aliwaf_action='block' ) as
status_405,count_if(status=444) as status_444,count_if(status=499) as
status_499,count_if(status=500) as status_500,count_if(status=502) as
status_502,count_if(status=503) as status_503,count_if(status=504) as
status_504,COUNT(*)
as countall from log group by user_id,host)) where  countall>120 order
by Rate_405 DESC  limit 5
```

status:405 and aliwaf_action='acl'

A request is blocked by HTTP ACL policy rules.

```
user_id: 1111111111111 |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2)
as Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405 and aliwaf_action='acl') as
status_405,count_if(status=444) as status_444,count_if(status=499) as
status_499,count_if(status=500) as status_500,count_if(status=502) as
status_502,count_if(status=503) as status_503,count_if(status=504) as
```

```
status_504,COUNT(*)  as countall from log group by user_id,host))
where
countall>120 order by Rate_405 DESC  limit 5
```

status:444

A request is blocked by custom HTTP flood protection rules.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as Rate_503,
round(round(status_504*1.0000/countall,4)*100,2)
as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as status_200,count_if(
status=302)
as status_302,count_if(status=404) as status_404,count_if(status=405)
as
status_405,count_if(status=444) as status_444,count_if(status=499) as
status_499,count_if(status=500) as status_500,count_if(status=502) as
status_502,count_if(status=503) as status_503,count_if(status=504) as
status_504,COUNT(*)  as countall from log group by user_id,host))
where  countall>120 order by Rate_444 DESC  limit 5
```

status:499

After a client sends a request, the server does not return data. After the maximum wait time of the client is reached, the client disconnects, and the server returns this status code.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
```

```
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=444) as
status_444,count_if(status=499) as status_499,count_if(status=500) as
status_500,count_if(status=502) as status_502,count_if(status=503) as
status_503,count_if(status=504) as status_504,COUNT(*)  as countall
from
log group by user_id,host)) where  countall>120 order by Rate_499
DESC  limit 5
```

status:500

A request cannot be processed due to the 500 Internal Server Error.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round
```

```
(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=444) as
status_444,count_if(status=499) as status_499,count_if(status=500) as
status_500,count_if(status=502) as status_502,count_if(status=503) as
status_503,count_if(status=504) as status_504,COUNT(*)  as countall
from
log group by user_id,host)) where  countall>120 order by Rate_500
DESC  limit 5
```

status:502

The server is used as a gateway or a proxy and receives invalid responses from the upstream server due to a 502 Bad Gateway error. The origin server does not respond due to low quality performance of the back-to-origin network or the fact that back-to-origin requests are blocked by access control policies configured for the origin server.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as Rate_502,
round(round(status_503*1.0000/countall,4)*100,2)
as Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
```

```
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=444) as
status_444,count_if(status=499) as status_499,count_if(status=500) as
status_500,count_if(status=502)
as status_502,count_if(status=503) as status_503,count_if(status=504)
as
status_504,COUNT(*)  as countall from log group by user_id,host))
where  countall>120 order by Rate_502 DESC  limit 5
```

status:503

The service is unavailable due to overloads or maintenance needs.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405)
as status_405,count_if(status=444) as status_444,count_if(status=499)
as
status_499,count_if(status=500) as status_500,count_if(status=502) as
status_502,count_if(status=503) as status_503,count_if(status=504) as
status_504,COUNT(*)
as countall from log group by user_id,host)) where  countall>120 order
```

```
by Rate_503 DESC  limit 5
```

status:504

The server serves as a gateway or proxy and fails to receive requests from the upstream server in time. The 504 Gateway Timeout error occurred.

```
* |select user_id,host as "Domain",Rate_200 as
"Status code 200 percentage",Rate_302 as "Status code 302 percentage",
Rate_404 as "Status code 404 percentage",Rate_405
as "Status code 405 percentage",Rate_444 as "Status code 444
percentage",Rate_499 as "Status code 499 percentage",Rate_500
as "Status code 500 percentage",Rate_502 as "Status code 502
percentage",Rate_503 as "Status code 503 percentage",Rate_504
as "Status code 504 percentage",countall/60 as
"aveQPS",status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from(SELECT user_id,host,round(round(status_200*1.0000/countall,4)*100
,2) as
Rate_200,round(round(status_302*1.0000/countall,4)*100,2) as Rate_302,
round(round

(status_404*1.0000/countall,4)*100,2) as
Rate_404,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_405,round(round

(status_405*1.0000/countall,4)*100,2) as
Rate_444,round(round

(status_405*1.0000/countall,4)*100,2)
as Rate_499,round(round(status_500*1.0000/countall,4)*100,2) as
Rate_500,round(round(status_502*1.0000/countall,4)*100,2) as
Rate_502,round(round(status_503*1.0000/countall,4)*100,2) as
Rate_503,round(round(status_504*1.0000/countall,4)*100,2) as
Rate_504,status_200,status_302,status_404,status_405,status_444,
status_499,status_500,status_502,status_503,status_504,countall
from (select user_id,host,count_if(status=200) as
status_200,count_if(status=302) as status_302,count_if(status=404) as
status_404,count_if(status=405) as status_405,count_if(status=444) as
status_444,count_if(status=499) as status_499,count_if(status=500) as
status_500,count_if(status=502) as status_502,count_if(status=503) as
status_503,count_if(status=504) as status_504,COUNT(*)  as countall
from
log group by user_id,host)) where  countall>120 order by Rate_504 DESC

limit 5
```