

Log Tank Service

User Guide

Issue 01
Date 2025-02-12



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Log Management.....	1
1.1 Overview.....	1
1.2 Managing Log Groups.....	1
1.3 Managing Log Streams.....	4
1.4 Viewing Log Management.....	6
1.5 Managing Tags.....	10
1.6 Setting Multi-Account Log Aggregation.....	12
2 Log Ingestion.....	15
2.1 Ingesting Cloud Service Logs to LTS.....	15
2.1.1 Ingesting AOM Logs to LTS.....	15
2.1.2 Ingesting APIG Logs to LTS.....	15
2.1.3 Ingesting BMS Text Logs to LTS.....	16
2.1.4 Ingesting CBH Logs to LTS.....	22
2.1.5 Ingesting CCE Application Logs to LTS.....	22
2.1.6 Ingesting CFW Logs to LTS.....	36
2.1.7 Ingesting CTS Logs to LTS.....	36
2.1.8 Ingesting GaussDB(DWS) Logs to LTS.....	37
2.1.9 Ingesting ECS Text Logs to LTS.....	37
2.1.10 Ingesting ELB Logs to LTS.....	43
2.1.11 Ingesting Enterprise Router Logs to LTS.....	44
2.1.12 Ingesting FunctionGraph Logs to LTS.....	44
2.1.13 Ingesting ModelArts Logs to LTS.....	45
2.1.14 Ingesting SMN Logs to LTS.....	45
2.1.15 Ingesting SecMaster Logs to LTS.....	45
2.1.16 Ingesting ServiceStage Containerized Application Logs to LTS.....	45
2.1.17 Ingesting ServiceStage Cloud Host Logs to LTS.....	53
2.1.18 Ingesting VPC Logs to LTS.....	58
2.1.19 Ingesting WAF Logs to LTS.....	59
2.2 Using APIs to Ingest Logs to LTS.....	59
2.2.1 Collecting Logs Using APIs.....	59
2.2.2 API for Reporting Logs.....	61
2.2.3 API for Reporting High-Precision Logs.....	65
2.3 Other Ingestion Modes.....	70

2.3.1 Ingesting Logs to LTS Across IAM Accounts.....	70
2.3.2 Ingesting Self-Built Kubernetes Application Logs to LTS.....	74
2.4 Setting ICAgent Structuring Parsing Rules.....	87
3 Host Management.....	103
3.1 Managing Host Groups.....	103
3.2 Managing Hosts.....	109
3.2.1 Installing ICAgent (Intra-Region Hosts).....	109
3.2.2 Installing ICAgent (Extra-Region Hosts).....	113
3.2.3 Managing ICAgent.....	123
4 Log Search and Analysis.....	127
4.1 Overview.....	127
4.2 Setting Cloud Structuring Parsing.....	127
4.2.1 Overview.....	127
4.2.2 Setting Cloud Structuring Parsing.....	128
4.2.3 Setting a Structuring Template.....	134
4.2.4 Setting Structured and Tag Fields.....	137
4.2.5 Setting Custom Log Time.....	138
4.3 Setting Indexes.....	141
4.4 Searching Logs.....	157
4.4.1 Accessing the Log Search Page.....	157
4.4.2 Using LTS Search Syntax.....	161
4.4.3 Creating an LTS Quick Analysis Task.....	172
4.4.4 Saving Conditions for Quick Search.....	173
4.5 Viewing Real-Time Logs.....	175
4.6 Analyzing Logs in LTS.....	176
4.7 SQL Analysis Syntax.....	178
4.7.1 Overview.....	178
4.7.2 SQL Aggregate Functions.....	182
4.7.3 SQL Period-over-Period Functions.....	183
4.7.4 SQL JSON Functions.....	186
4.7.5 SQL IP Functions.....	187
4.7.6 SQL Mathematical Functions.....	189
4.7.7 SQL Time Functions.....	193
4.7.8 SQL Extrema Functions.....	200
4.7.9 SQL String Functions.....	200
4.7.10 SQL SPLIT Functions.....	204
4.7.11 SQL Comparison Operators.....	206
4.7.12 SQL IP Address Functions.....	207
4.7.13 SQL Reduction Functions.....	209
4.7.14 Other SQL Functions.....	210
4.7.15 SQL JOIN Syntax.....	211
4.7.16 SQL Query Example.....	212

5 Log Visualization	214
5.1 Overview	214
5.2 Visualizing Logs in Statistical Charts	214
5.2.1 Statistical Charts	215
5.2.2 Table	216
5.2.3 Bar Chart	218
5.2.4 Line Chart	220
5.2.5 Pie Chart	222
5.2.6 Number Chart	226
5.2.7 Digital Line Chart	227
5.2.8 Map	228
5.2.9 Funnel Chart	230
5.3 Visualizing Logs in Dashboards	231
5.3.1 Creating a Dashboard	231
5.3.2 Adding a Dashboard Filter	236
5.3.3 Dashboard Templates	238
5.3.3.1 APIG Dashboard Templates	238
5.3.3.2 CCE Dashboard Templates	243
5.3.3.3 CDN Dashboard Templates	254
5.3.3.4 CFW Dashboard Templates	258
5.3.3.5 CSE Dashboard Templates	261
5.3.3.6 DCS Dashboard Template	266
5.3.3.7 DDS Dashboard Template	267
5.3.3.8 DMS Dashboard Template	268
5.3.3.9 DSL Dashboard Template	269
5.3.3.10 ER Dashboard Template	270
5.3.3.11 METRIC Dashboard Template	271
5.3.3.12 Nginx Dashboard Templates	273
5.3.3.13 VPC Dashboard Template	278
5.3.3.14 WAF Dashboard Templates	280
6 Log Alarms	286
6.1 Configuring Log Alarm Rules	286
6.2 Configuring Log Alarm Notifications	296
6.2.1 Creating a Message Template on the LTS Console	296
6.2.2 Creating an Alarm Action Rule	301
6.3 Viewing Alarms in LTS	302
7 Log Transfer	304
7.1 Overview	304
7.2 Transferring Logs to OBS	304
8 Log Processing	313
8.1 Processing Logs with FunctionGraph Function Templates	313

9 Configuration Center.....	314
9.1 Setting LTS Log Collection Quota.....	314
9.2 Configuring Log Content Delimiters.....	315
9.3 Setting ICAgent Collection.....	318

1 Log Management

1.1 Overview

LTS manages logs by log group and stream for easy classification. Before using LTS, create a log group and then multiple log streams in the group. By collecting and storing log data in different log streams, you can search, analyze, and transfer log data and set alarm rules by log stream.

To better understand and use LTS, perform the following steps:

1. Create a log group. For details, see [Managing Log Groups](#).
2. Create a log stream. For details, see [Managing Log Streams](#).
3. On the **Log Management** page, view resource statistics, and the **My Favorites**, **My Favorites(Local Cache)**, and **Recently Visited** lists. For details, see [Viewing Log Management](#).

1.2 Managing Log Groups

A log group is the basic unit for LTS to manage logs. It classifies and consists of log streams, but does not store any log data. Up to 100 log groups can be created for an account.

A log group usually corresponds to a project or business in a company. You are advised to sort log streams of various applications or services within a project or business to the same log group. In this way, project staff only need to monitor log streams in the log group corresponding to their project, without being distracted by log streams of other projects.

LTS allows you to add tags to log groups to help O&M personnel manage services.

Prerequisites

You have obtained an account and its password for logging in to the LTS console.

Creating a Log Group

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.
- Step 2** On the **Log Management** page, click **Create Log Group**.
- Step 3** On the displayed page, set log group parameters by referring to [Table 1-1](#).

Table 1-1 Log group parameters

Parameter	Description
Log Group Name	<ul style="list-style-type: none">Enter 1 to 64 characters, including only letters, digits, hyphens (-), underscores (_), and periods (.). Do not start with a period or underscore or end with a period.Collected logs are sent to log streams of the corresponding log groups. If there are too many logs to collect, separate logs into different log groups based on log types, and name log groups in an easily identifiable way.
Enterprise Project Name	Select an enterprise project. You can click View Enterprise Projects to view all enterprise projects. Enterprise projects allow you to manage cloud resources and users by project.
Log Retention (Days)	Specify the log retention duration for the log group, that is, how many days the logs will be stored in LTS after being reported to LTS. By default, logs are retained for 30 days (customizable for 1 to 30 days).
Tag	Tag the log group as required. Click Add Tags , enter a tag key and value, and enable Apply to Log Stream . <ul style="list-style-type: none">To add more tags, repeat this step. A maximum of 20 tags can be added.To delete a tag, click Delete in the Operation column of the tag.A tag key can contain up to 128 characters, and a tag value can contain up to 255 characters.A tag key must be unique.If a tag is used by a transfer task, you need to modify the task configuration after deleting the tag.
Remark	Enter remarks. The value contains up to 1,024 characters.

- Step 4** Click **OK**. The created log group will be displayed in the log group list.
- In the log group list, you can view information such as the log group name, tags, and log streams.
 - Click the log group name to access the log stream details page.

- When multiple log groups are created concurrently, there may be a limit exceeding error.

----End

Deleting a Log Group

You can delete a log group that is no longer needed. Deleting a log group will also delete the log streams and log data in the log group. **This may lead to exceptions in related tasks. In addition, deleted log groups cannot be restored. Exercise caution when performing this operation.**

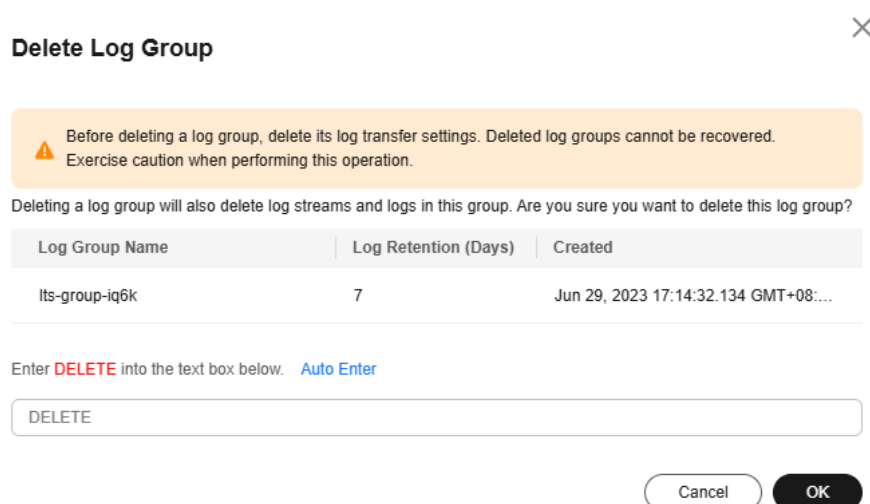
NOTE

If you want to delete a log group that is associated with a log transfer task, delete the task first.

Step 1 In the log group list on the **Log Management** page, locate the target log group and click **Delete** in the **Operation** column.

Step 2 Enter **DELETE** and click **OK**.

Figure 1-1 Deleting a log group



----End

Searching Log Groups/Streams

In the log group list, you can set the following filter criteria:

- Log group/stream
- Original log group/stream name
- Log group name/ID
- Log stream name/ID
- Log group tag
- Remarks

1.3 Managing Log Streams

LTS manages logs by log stream. Collected logs of different types are classified and stored in different log streams for easier log management. If there are a large number of logs, you can create multiple log streams and name them for quick log search. For example, you can sort operation logs and access logs into different log streams, making it easier to find specific logs when you need them. You can add tags to log streams to help O&M personnel manage services.

A maximum of 100 log streams can be created in a log group. If you cannot create a log stream, delete unnecessary log streams before creating new ones, or create log streams in another log group.

Prerequisites

You have created a log group.

Creating a Log Stream


- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.
- Step 2** Click  on the left of the target log group.
- Step 3** Click **Create Log Stream**. On the displayed page, set log stream parameters by referring to [Table 1-2](#).

Table 1-2 Log stream parameters

Parameter	Description
Log Group Name	The name of the target log group is displayed by default.
Log Stream Name	Enter 1 to 64 characters, including only letters, digits, hyphens (-), underscores (_), and periods (.). Do not start with a period or underscore or end with a period.
Enterprise Project Name	Select the required enterprise project. The default value is default . You can click View Enterprise Projects to view all enterprise projects.
Log Retention (Days)	Specify the log retention duration for the log stream, that is, how many days the logs will be stored in LTS after being reported to LTS. By default, logs are retained for 30 days (customizable for 1 to 30 days). <ul style="list-style-type: none"> If you enable Log Retention (Days) for the log stream, logs are retained for the duration set for the log stream. If you disable Log Retention (Days) for the log stream, logs are retained for the duration set for the log group.

Parameter	Description
Tag	<p>You can tag log streams as required. Click Add Tags and enter a tag key and tag value.</p> <p>NOTE</p> <ul style="list-style-type: none"> • To add more tags, repeat this step. A maximum of 20 tags can be added. • To delete a tag, click Delete in the Operation column of the tag. • A tag key can contain up to 128 characters, and a tag value can contain up to 255 characters. • A tag key must be unique. • If a tag is used by a transfer task, you need to modify the task configuration after deleting the tag.
Remark	Enter remarks. The value contains up to 1,024 characters.

Step 4 Click **OK**. In the log stream list, you can view information such as the log stream name and operations.

----End

Deleting a Log Stream

You can delete a log stream that is no longer needed. Deleting a log stream will also delete the log data in the log stream. **This may lead to exceptions in related tasks. In addition, deleted log streams cannot be restored. Exercise caution when performing this operation.**

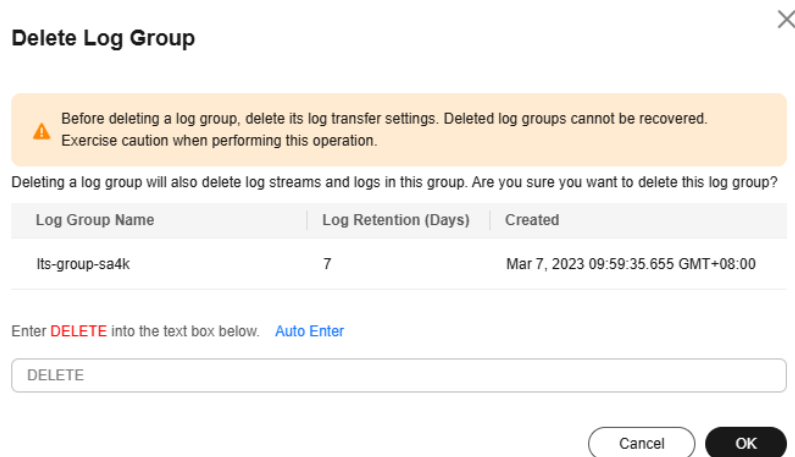
NOTE

- Before deleting a log stream, check whether any log collection task is configured for it. If there is a log collection task, deleting the log stream may affect log reporting.
- If you want to delete a log stream that is associated with a log transfer task, delete the task first.

Step 1 In the log stream list, locate the target log stream and click **Delete** in the **Operation** column.

Step 2 Enter **DELETE** and click **OK**.

Figure 1-2 Deleting a log stream



----End

Other Operations

- Adding a log stream to favorites
Click **More** > **Edit** in the **Operation** column of a log stream. On the displayed dialog box, enable **My Favorites** and/or **My Favorites(Local Cache)**. The stream is then displayed in the **My Favorites/My Favorites(Local Cache)** list.
- Viewing details
Click **More** > **Details** in the **Operation** column of a log stream to view its details, including its name, ID, and creation time.

1.4 Viewing Log Management

The log management page displays resource statistics, your favorite log streams/ favorite log streams (local cache), alarm statistics, latest alarms, and recently viewed log streams.

Resource Statistics

The **Statistics** area shows resource statistics and details by category in charts. The statistics are for reference only.

- Step 1** Log in to the management console and choose **Management & Deployment** > **Log Tank Service**. The **Log Management** page is displayed by default.
- Step 2** Under **Overview**, click **Details** in the upper right corner of the **Statistics** area to access the resource statistics details page.
- Step 3** Select a time range as required. By default, resource statistics display log resource data of one week (from now).

There are three types of time range: relative time from now, relative time from last, and specified time.

 NOTE


- **From now:** queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- **From last:** queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified:** queries log data that is generated in a specified time range.

Step 4 View the resource statistics.

- **Read/Write:** LTS charges for the amount of compressed log data read from and written to LTS. Generally, the log compression ratio is 5:1.
- **Index Traffic - Standard:** Raw logs are full-text indexed (delimited) by default for log search. When logs are written to databases, traffic fees are charged at a time.
- **Standard Storage Volume:** Space used for storing compressed logs, indexes, and copies is billed. The space is roughly the size of the raw logs.
- **Raw Log Traffic:** size of raw logs.

Step 5 View the resource statistics of **Log Groups (Top 100)** and **Log Streams (Top 100)**.

You can select a time range and view the daily standard storage volume (GB), daily index traffic - standard (GB), and daily read/write traffic (GB) of this period in tables or bar charts.

- For a new log group or log stream, resource statistics will be collected in at least one hour.
- Click the name of one of the top 100 log groups to query its log stream resource statistics.
- Click  to download the resource statistics of the log groups and streams.


 NOTE

The downloaded files are in **.CSV** format.

----End

Alarm Statistics and Latest Alarms

In the lower part of **Overview**, you can view alarm statistics and latest alarms.

- The **Alarms** area displays the total number of LTS alarms and the number of alarms of each severity (**Critical**, **Major**, **Minor**, and **Warning**). You can view alarm statistics of the last 30 minutes, last 1 hour, last 6 hours, last 1 day, or last 1 week.
- The **Latest Alarms** area displays a maximum of three latest alarm rules in the last 30 minutes. To view more alarms or add alarm rules, click .

Log Applications

LTS provides multiple log applications under **Overview** on the **Log Management** page. These log applications are out-of-the-box dashboard templates for quick log analysis.

- **ELB Log Center:** Elastic Load Balance (ELB) logs can be ingested to LTS and displayed in the ELB dashboard templates.
- **APIG Log Center:** API Gateway (APIG) logs can be ingested to LTS and displayed in the APIG dashboard templates.
- **VPC Flow Log Center:** Virtual Private Cloud (VPC) logs can be ingested to LTS and displayed in the VPC dashboard templates.
- **CFW Log Center:** Cloud Firewall (CFW) logs can be ingested to LTS and displayed in the CFW dashboard templates. For details, see [CFW Dashboard Templates](#).
- **CTS Log Center:** Cloud Trace Service (CTS) logs can be ingested to LTS but cannot be displayed in the CTS dashboard templates. For details, see [Ingesting CTS Logs to LTS](#).
- **Multi-Account Log Center:** Log streams of multiple accounts can be copied to a specified account for central storage and analysis of logs from those accounts. For details, see [Setting Multi-Account Log Aggregation](#).

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Log Groups

Log groups and log streams are listed in **Log Groups**. For more information, see [Managing Log Groups](#) and [Managing Log Streams](#).

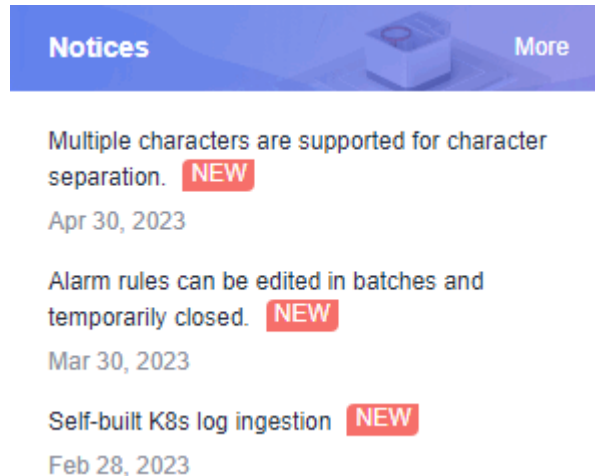
Notices

This area displays the new functions and news of LTS.

To view more function descriptions, click [More](#).

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.



My Favorites/My Favorites (Local Cache)


This area displays the log streams you have added to favorites, including **My Favorites** and **My Favorites(Local Cache)**.

- **My Favorites**: Save log streams to the database. This function is disabled by default. If your account has the write permission, **My Favorites** and **My Favorites(Local Cache)** are displayed.
- **My Favorites(Local Cache)**: Save log streams to the local cache of the browser. This function is disabled by default. This parameter is displayed for both writable and read-only users.

If your account has the write permission, at least one of **My Favorites** and **My Favorites(Local Cache)** is enabled. Otherwise, log streams cannot be added to favorites.

Adding frequently used log streams to your favorites helps you quickly locate them.

The following example shows how to add a log stream of log group **lts-test** to favorites:

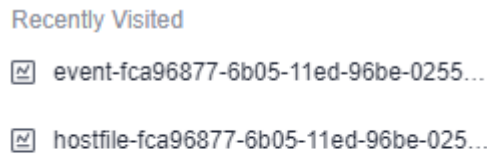
- Step 1** In the **Log Groups** list, click  on the left of log group **lts-test**.
- Step 2** Click **More > Edit** in the **Operation** column of the target log stream. On the displayed dialog box, enable **My Favorites** and/or **My Favorites(Local Cache)** and click **OK**.
- Step 3** After the log stream is added to favorites, it is displayed in **My Favorites/My Favorites(Local Cache)** on the right.

----End

Recently Visited

This area displays a maximum of three log streams that are recently visited.

Figure 1-3 Recently visited



FAQ


This area displays frequently asked questions.

1.5 Managing Tags

You can tag log groups, log streams, host groups, and log ingestion configurations.

Tagging a Log Group

Users can add, delete, modify, and query tags on the log group page.

1. Log in to the management console and choose **Management & Deployment** > **Log Tank Service**. The **Log Management** page is displayed by default.
2. Move the cursor to the **Tags** column of the target log group and click .
3. On the **Edit** page that is displayed, click **Add Tags** and enter a tag key and value. If you enable **Apply to Log Stream**, the tag will be synchronized to all log streams in the log group.



NOTE

- A tag key can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`. Do not start with an underscore (`_`).
 - A tag value can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`.
 - To add multiple tags, repeat this step.
 - To delete a tag, click **Delete** in the **Operation** column of the tag.
 - A tag key can contain up to 128 characters, and a tag value can contain up to 255 characters.
 - A tag key must be unique.
 - If a tag is used by a transfer task, you need to modify the task configuration after deleting the tag.
4. Click **OK**.

On the **Log Management** page, you can view the added tags in the **Tags** column of the log group.

Tagging a Log Stream

You can add, delete, modify, and view tags on the log stream list page. When you manage the tags of a single log stream, the changes will not be synchronized to other streams.

1. Click  in front of the name of the target log group.
2. Move the cursor to the **Tags** column of the target log stream and click .
3. On the **Edit** page that is displayed, click **Add Tags** and enter a tag key and value.

NOTE

- A tag key can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`. Do not start with an underscore (`_`).
 - A tag value can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`.
 - To add multiple tags, repeat this step.
 - To delete a tag, click **Delete** in the **Operation** column of the tag.
 - A tag key can contain up to 128 characters, and a tag value can contain up to 255 characters.
 - A tag key must be unique.
 - If a tag is used by a transfer task, you need to modify the task configuration after deleting the tag.
4. Click **OK**.

In the log stream list, you can view the system tags and added custom tags in the **Tags** column of the log stream.

Tagging a Host Group

You can add, delete, modify, and view tags of host groups. When you manage the tags of a single host group, the changes will not be synchronized to other groups.

1. Choose **Host Management > Host Groups** in the navigation pane.
2. Locate the target host group and click **Configure Tag** in the **Operation** column.
3. On the **Edit** page that is displayed, click **Add Tags** and enter a tag key and value.

NOTE

- A tag key can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`. Do not start with an underscore (`_`).
 - A tag value can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`.
 - To add multiple tags, repeat this step.
 - To delete a tag, click **Delete** in the **Operation** column of the tag.
 - A tag key can contain up to 128 characters, and a tag value can contain up to 255 characters.
 - A tag key must be unique.
4. Click **OK**. You can view the added tags in the **Tags** column of the host group.

Tagging a Log Ingestion Configuration

You can add, delete, modify, and view tags of log ingestion configurations. When you manage the tags of a single log ingestion configuration, the changes will not be synchronized to other configurations.

1. In the navigation pane, choose **Log Ingestion > Ingestion Management**.
2. Locate the target log ingestion configuration and click **Configure Tag** in the **Operation** column.
3. On the **Edit** page that is displayed, click **Add Tags** and enter a tag key and value.

NOTE

- A tag key can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`. Do not start with an underscore (`_`).
 - A tag value can contain only UTF-8 letters, digits, spaces, and the following special characters `_:/=+-@`.
 - To add multiple tags, repeat this step.
 - To delete a tag, click **Delete** next to the tag on the tag management dialog box.
 - A tag key can contain up to 128 characters, and a tag value can contain up to 255 characters.
 - A tag key must be unique.
4. Click **OK**. You can view the added tags in the **Tags** column of the log ingestion configuration.

1.6 Setting Multi-Account Log Aggregation

The multi-account log center allows you to copy log streams of multiple accounts to a specified account to centrally store and analyze logs across accounts. This meets the requirements in scenarios such as security compliance and centralized analysis.

Background

- Corporate groups often use Landing Zone, a multi-account solution that allows business departments to use different accounts, to enhance security by isolating permissions and resources.
- Their security compliance departments expect to collect logs centrally, and aggregate key logs of different accounts of each business department into one log account for centralized storage and analysis to meet the security compliance audit requirements of different countries and regions.
- To meet operations analysis requirements, their operations departments may also expect to aggregate key logs of different accounts of each business department into one log account for centralized storage and analysis, to facilitate unified big data processing and visual display.

Solution

- The Organizations service helps you govern multiple accounts within your organization. It enables you to consolidate multiple accounts into an organization that you create and centrally manage these accounts. You can

use Service Control Policies (SCPs) to control the maximum available permissions for all accounts in your organization. This helps you better meet the service security and compliance requirements of your business.

- LTS provides the multi-account log center based on Organizations. You can log in to Organizations using the management account and specify a member account in your organization as the delegated administrator account of the trusted service LTS, and then go to the LTS console to copy the log streams of another member account to the management or delegated administrator account. In this way, logs of multiple accounts are aggregated into one account.
- The target log groups or streams are generated by copying the source log groups or streams from a member account to the management or delegated administrator account. Therefore, they do not interfere with the source log groups or streams of the member account. You can configure transfer, consumption, and processing tasks for them separately.

Prerequisites

- An organization has been created.
- LTS has been set as a trusted service.
- The management or delegated administrator account has been granted the read-only permission on the Organizations service.

Setting a Multi-Account Log Aggregation Task

Step 1 Log in to the console using the management or delegated administrator account and choose **Log Tank Service** in the navigation pane. The **Log Management** page is displayed.

Step 2 Click **Multi-Account Log Center** under **Log Applications**.

Step 3 On the displayed page, enable **Log receiving status**, select a member account on the left, and select its source log groups or streams on the right. You can also customize the names of the target log groups or streams.

NOTE

- To stop using multi-account log aggregation, simply disable **Log receiving status**. In this case, all aggregation settings become invalid, and logs in the source log streams stop being aggregated to the target log streams.
- The number of log streams to be created cannot exceed the limit.

Step 4 Click **OK**. It takes about 5 minutes to create the aggregation settings for the account. Refresh the page later to view the settings.

 **NOTE**

- By default, a target log stream is initialized using the index and structuring configurations of the log stream of the source account. After the aggregation is configured, subsequent index and structuring configuration changes of the source will not be synchronized to the target.
- Deleting the log groups or streams of the source account does not affect their target equivalents.
- After you deselect a log group or stream in **Aggregation Settings** and click **OK**, its logs will not be aggregated to the target log group or stream.

----End

2 Log Ingestion

2.1 Ingesting Cloud Service Logs to LTS

2.1.1 Ingesting AOM Logs to LTS

LTS can collect logs from AOM.

For details, see [Accessing LTS](#).

 NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.2 Ingesting APIG Logs to LTS

LTS can collect logs from APIG.

 NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Prerequisites

You have created and used an API gateway.

Configuring APIG Log Ingestion in LTS

Perform the following operations to configure APIG log ingestion:

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **APIG (API Gateway)**.
- Step 3** Select a log stream.

1. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
2. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
3. Click **Next: Configure APIG**.

Step 4 Click **Configure APIG** to configure APIG on the APIG console.

Step 5 Click **Next: Configure Log Stream**.

Table 2-1 Log stream parameters

Parameter	Description
Auto Structure and Index	If this function is enabled, the structuring for the log stream is based on the APIG system template, and the indexing enables quick analysis for all parsed APIG fields.

Step 6 Click **Submit**.

----End

2.1.3 Ingesting BMS Text Logs to LTS

A Bare Metal Server (BMS) features both the scalability of VMs and high performance of physical servers. It provides dedicated servers on the cloud, delivering the performance and security required by core databases, critical applications, high-performance computing (HPC), and big data.

After you configure BMS log ingestion, ICAgent collects logs from BMS based on your specified rules, and sends the logs to LTS by log stream. You can view and analyze these logs on the LTS console for improving host running stability and information security.

Perform the following steps to complete the ingestion configuration:

1. [Step 1: Select a Log Stream](#)
2. [Step 2: \(Optional\) Select a Host Group](#)
3. [Step 3: Configure the Collection](#)
4. [Step 4: Configure Indexing](#)
5. [Step 5: Complete the Ingestion Configuration](#)


To collect logs from multiple scenarios, [set multiple ingestion configurations in a batch](#).

Prerequisites

ICAgent has been [installed](#) and [added](#) to the host group.

Step 1: Select a Log Stream

1. Log in to the management console and choose **Management & Deployment > Log Tank Service**.

2. Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **BMS (Bare Metal Server)**.
Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > BMS (Bare Metal Server)** on the displayed page.
Alternatively, choose **Log Management** in the navigation pane and click the target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and click **Ingest Log**. In the displayed dialog box, click **BMS (Bare Metal Server)**.
3. Select a log group from the **Log Group** drop-down list. If there is no desired log group, click **Create Log Group**. For details, see [Managing Log Groups](#).
4. Select a log stream from the **Log Stream** drop-down list. If there is no desired log stream, click **Create Log Stream**. For details, see [Managing Log Streams](#).
5. Click **Next: (Optional) Select Host Group**.

Step 2: (Optional) Select a Host Group

1. Select one or more host groups from which you want to collect logs. If there are no desired host groups, click **Create** above the host group list to create one. For details, see [Managing Host Groups](#).

NOTE

You can also skip this step, but the collection configuration will not take effect. You are advised to select a host group during the first ingestion configuration. If you skip this step, follow either of the following ways to configure host groups after the ingestion configuration is complete:

- Choose **Host Management > Host Groups** in the navigation pane and associate host groups with ingestion configurations.
 - Choose **Log Ingestion > Ingestion Management** in the navigation pane. In the ingestion configuration list, click **Modify** in the **Operation** column. On the page displayed, select required host groups.
2. Click **Next: Configurations**.

Step 3: Configure the Collection

After selecting host groups, configure the collection as follows:

NOTE

- Ensure that sensitive information is not collected.
 - If a collection path of a host has been configured in AOM, do not configure the path in LTS.
 - If log files were last modified more than 12 hours earlier than the time when the path is added, the files are not collected.
1. **Collection Configuration Name:** Enter 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Do not start with a period or underscore, or end with a period.
If you want to reuse existing collection configurations, click **Import Configuration** next to the text box. On the **Import Configuration** page, select a configuration and click **OK**.

 NOTE

Import Old-Edition Configuration: Import the host ingestion configuration of the old version to the log ingestion of the new version.

- If LTS is newly installed and **Import Old-Edition Configuration** is not displayed, you can directly create a configuration without importing the old one.
- If LTS is upgraded, **Import Old-Edition Configuration** is displayed. Import the old configuration or create one as required.

2. **Collection Paths:** Add one or more host paths. LTS will collect logs from these paths. The rules for setting collection paths are as follows:

- Logs can be collected recursively. A double asterisk (**) can represent up to 5 directory levels in a path.

For example, **/var/logs/**/a.log** will match the following logs:

```
/var/logs/a.log
/var/logs/1/a.log
/var/logs/1/2/a.log
/var/logs/1/2/3/a.log
/var/logs/1/2/3/4/a.log
/var/logs/1/2/3/4/5/a.log
```

 NOTE

- **/1/2/3/4/5/** indicates the 5 levels of directories under the **/var/logs** directory. All the **a.log** files found in all these levels of directories will be collected.
 - Only one double asterisk (**) can be contained in a collection path. For example, **/var/logs/**/a.log** is acceptable but **/opt/test/**/log/**** is not.
 - A collection path cannot begin with a double asterisk (**), such as ****/test**, to avoid collecting system files.
- You can use an asterisk (*) as a wildcard for fuzzy match. The wildcard (*) can represent one or more characters of a directory or file name.

 NOTE

If a log collection path is similar to **C:\windows\system32** but logs cannot be collected, enable Web Application Firewall (WAF) and configure the path again.

- Example 1: **/var/logs/*/a.log** will match all **a.log** files found in all directories under the **/var/logs/** directory:


```
/var/logs/1/a.log
/var/logs/2/a.log
```
- Example 2: **/var/logs/service-*/a.log** will match files as follows:


```
/var/logs/service-1/a.log
/var/logs/service-2/a.log
```
- Example 3: **/var/logs/service/a*.log** will match files as follows:


```
/var/logs/service/a1.log
/var/logs/service/a2.log
```
- If the collection path is set to a file name, the corresponding file is collected. Only text files can be collected.
- **Add Custom Wrapping Rule:** ICAgent determines whether a file is wrapped based on the file name rule. If your wrapping rule does not comply with the built-in rules, you can add a custom wrap rule to prevent log loss during repeated collection and wrapping.

LTS enables combined parsing, allowing you to create different structuring parsing rules for each collection configuration of a log stream.

If you have configured cloud structuring parsing, delete its configurations before configuring ICAgent structuring parsing.

7. Configure the log format and time by referring to [Table 2-3](#).

Table 2-3 Log collection settings

Parameter	Description
Log Format	<ul style="list-style-type: none"> ● Single-line: Each log line is displayed as a single log event. ● Multi-line: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.
Log Time	<p>System time: log collection time by default. It is displayed at the beginning of each log event.</p> <p>NOTE</p> <ul style="list-style-type: none"> ● Log collection time is the time when logs are collected and sent by ICAgent to LTS. ● Log printing time is the time when logs are printed. ICAgent collects and sends logs to LTS with an interval of 1 second. ● Restriction on log collection time: Logs are collected within 24 hours before and after the system time. <p>Time wildcard: You can set a time wildcard so that ICAgent will look for the log printing time as the beginning of a log event.</p> <ul style="list-style-type: none"> ● If the time format in a log event is 2019-01-01 23:59:59.011, the time wildcard should be set to YYYY-MM-DD hh:mm:ss.SSS. ● If the time format in a log event is 19-1-1 23:59:59.011, the time wildcard should be set to YY-M-D hh:mm:ss.SSS. <p>NOTE</p> <p>If a log event does not contain year information, ICAgent regards it as printed in the current year.</p> <p>Example:</p> <pre> YY - year (19) YYYY - year (2019) M - month (1) MM - month (01) D - day (1) DD - day (01) hh - hours (23) mm - minutes (59) ss - seconds (59) SSS - millisecond (999) hpm - hours (03PM) h:mmpm - hours:minutes (03:04PM) h:mm:sspm - hours:minutes:seconds (03:04:05PM) hh:mm:ss ZZZZ (16:05:06 +0100) hh:mm:ss ZZZ (16:05:06 CET) hh:mm:ss ZZ (16:05:06 +01:00) </pre>

Parameter	Description
Log Segmentation	This parameter needs to be specified if the Log Format is set to Multi-line . By generation time indicates that a time wildcard is used to detect log boundaries, whereas By regular expression indicates that a regular expression is used.
By regular expression	You can set a regular expression to look for a specific pattern to indicate the beginning of a log event. This parameter needs to be specified when you select Multi-line for Log Format and By regular expression for Log Segmentation .

 NOTE

The time wildcard and regular expression will look for the specified pattern right from the beginning of each log line. If no match is found, the system time, which may be different from the time in the log event, is used. In general cases, you are advised to select **Single-line** for **Log Format** and **System time** for **Log Time**.

Step 4: Configure Indexing

1. (Optional) Configure indexing. For details, see [Setting Indexes](#).
2. Click **Submit**.

Step 5: Complete the Ingestion Configuration

The created ingestion configuration will be displayed.

- Click its name to view its details.
- Click **Modify** in the **Operation** column to modify the ingestion configuration.
- Click **Configure Tag** in the **Operation** column to add a tag.
- Click **More > Copy** in the **Operation** column to copy the ingestion configuration.
- Click **More > Delete** in the **Operation** column to delete the ingestion configuration.

 NOTE

Deleting an ingestion configuration may lead to log collection failures, potentially resulting in service exceptions related to user logs. In addition, the deleted ingestion configuration cannot be restored. Exercise caution when performing this operation.

Setting Multiple Ingestion Configurations in a Batch

You can set multiple ingestion configurations for multiple scenarios in a batch, avoiding repetitive setups.

- Step 1** On the **Ingestion Management** page, click **Batch Ingest** to go to the details page. For details, see [Table 2-4](#).

Table 2-4 Adding configurations in batches

Type	Parameter	Description
Basic Settings	Ingestion Type	Choose BMS (Bare Metal Server) .
	Configurations to Add	Enter the number of ingestion configurations in the text box and click Add . A maximum of 100 ingestion configurations can be added, including the one already exists under Ingestion Settings by default. Therefore, you can add up to 99 more.
Ingestion Settings	Configuration List	<ol style="list-style-type: none">1. The ingestion configurations are displayed on the left. You can add up to 99 more configurations.2. The ingestion configuration details are displayed on the right. Set them by referring to Step 3: Configure the Collection.3. After an ingestion configuration is complete, you can click Apply to Other Configurations to copy its settings to other configurations.

Step 2 Click **Check Parameters**. After the check is successful, click **Submit**.

The added ingestion configurations will be displayed on the **Ingestion Management** page after the batch creation is successful.

Step 3 (Optional) Perform the following operations on ingestion configurations:

- Select multiple existing ingestion configurations and click **Edit**. On the displayed page, select an ingestion type to modify the corresponding ingestion configurations.
- Select multiple existing ingestion configurations and click **Enable** or **Disable**. If you toggle off the switch in the **Status** column of an ingestion configuration, logs will not be collected for this configuration.
- Select multiple existing ingestion configurations and click **Delete**.

----End

2.1.4 Ingesting CBH Logs to LTS

LTS can collect logs from CBH.

For details, see [Configuring LTS](#).

2.1.5 Ingesting CCE Application Logs to LTS

CCE provides highly scalable, high-performance, enterprise-class Kubernetes clusters. With CCE, you can easily deploy, manage, and scale containerized applications.

After ingesting CCE logs to LTS, you can centrally manage and analyze them and visualize log reports on the LTS console. This helps you better monitor and

manage containerized application running, promptly detect container issues, and improve container performance and reliability.

Perform the following steps to complete the ingestion configuration:

1. [Step 1: Select a Log Stream](#)
2. [Step 2: Check Dependencies](#)
3. [Step 3: \(Optional\) Select a Host Group](#)
4. [Step 4: Configure the Collection](#)
5. [Step 5: Configure Indexing](#)
6. [Step 6: Complete the Ingestion Configuration](#)

To collect logs from multiple scenarios, [set multiple ingestion configurations in a batch](#).

Prerequisites

- ICAgent has been installed in the CCE cluster and a host group with custom identifiers has been created for related nodes. The system will automatically check these configurations and make necessary corrections when CCE logs are ingested to LTS.

NOTE

- On the **Hosts** page, select **CCE Clusters**, select the target cluster in the search box, and click **Upgrade ICAgent**. For details, see [Upgrading ICAgent](#).
- You have disabled **Output to AOM**.

Constraints


- Currently, ServiceStage hosting is not supported.
- CCE cluster nodes whose container engine is Docker are supported. For details, see [Node Overview](#).
- CCE cluster nodes whose container engine is Containerd are supported. You must be using ICAgent 5.12.130 or later.
- To collect container log directories mounted to host directories to LTS, you must configure the node file path.
- Constraints on the Docker storage driver: Currently, container file log collection supports only the overlay2 storage driver. devicemapper cannot be used as the storage driver. Run the following command to check the storage driver type:

```
docker info | grep "Storage Driver"
```
- If you select **Fixed log stream** for log ingestion, ensure that you have created a CCE cluster.

Step 1: Select a Log Stream

1. Log in to the management console and choose **Management & Deployment > Log Tank Service**.
2. Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **CCE (Cloud Container Engine)**.

Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > CCE (Cloud Container Engine)**.

Alternatively, choose **Log Management** in the navigation pane and click the target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and click **Ingest Log**. In the displayed dialog box, click **CCE (Cloud Container Engine)**.

3. Choose a collection mode between **Fixed log stream** and **Custom log stream**.
 - If you set **Collect** to **Fixed log stream**, perform the following steps:

Logs will be collected to a fixed log stream. The default log streams for a CCE cluster are **stdout-*{ClusterID}*** for standard output/errors, **hostfile-*{ClusterID}*** for node files, **event-*{ClusterID}*** for Kubernetes events, and **containerfile-*{ClusterID}*** for container files. Log streams are automatically named with a cluster ID. For example, if the cluster ID is **Cluster01**, the standard output/error log stream is **stdout-Cluster01**.

Log streams that can be created for a CCE cluster are **stdout-*{ClusterID}*** for standard output/errors, **hostfile-*{ClusterID}*** for node files, **event-*{ClusterID}*** for Kubernetes events, and **containerfile-*{ClusterID}*** for container files. If one of them has been created in a log group, the log stream will no longer be created in the same log group or other log groups.

 - i. Select a cluster from the **CCE Cluster** drop-down list.
 - ii. The default log group is **k8s-log-*ClusterID***. For example, if the cluster ID is **c7f3f4a5-bcb8-11ed-a4ec-0255ac100b07**, the default log group will be **k8s-log-c7f3f4a5-bcb8-11ed-a4ec-0255ac100b07**.

NOTE

If there is no such group, the system displays the following message: This log group does not exist and will be automatically created to start collecting logs.

- iii. Click **Next: Check Dependencies**.
- If you set **Collect** to **Custom log stream**, perform the following steps:
 - i. Select a cluster from the **CCE Cluster** drop-down list.
 - ii. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
 - iii. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
 - iv. Click **Next: Check Dependencies**.

Step 2: Check Dependencies

The system automatically checks the following items:

1. ICAgent has been installed (version 5.12.130 or later).
2. There is a host group with the custom identifier **k8s-log-*ClusterID***.
3. There is a log group named **k8s-log-*ClusterID***. If **Fixed log stream** is selected, this item is checked.
4. The recommended log stream exists. If **Fixed log stream** is selected, this item is checked.

You need to meet all the requirements before moving on. If not, click **Auto Correct**.

 **NOTE**

- **Auto Correct:** Check the previous settings with one click.
- **Check Again:** Recheck dependencies.

Step 3: (Optional) Select a Host Group

1. In the host group list, select one or more host groups from which you want to collect logs.

 **NOTE**

- The host group to which the cluster belongs is selected by default. You can also select host groups as required.
 - You can also skip this step, but the collection configuration will not take effect. You are advised to select a host group during the first ingestion configuration. If you skip this step, follow either of the following ways to configure host groups after the ingestion configuration is complete:
 - Choose **Host Management > Host Groups** in the navigation pane and associate host groups with ingestion configurations.
 - Choose **Log Ingestion > Ingestion Management** in the navigation pane. In the ingestion configuration list, click **Modify** in the **Operation** column. On the page displayed, select required host groups.
2. Click **Next: Configurations**.

Step 4: Configure the Collection

When CCE is used to ingest logs, the configuration details are as follows:

1. **Collection Configuration Name:** Enter 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Do not start with a period or underscore, or end with a period.
2. **Data Source:** Select a data source type and configure it. For details, see [Table 2-5](#).

Table 2-5 Data source parameters

Parameter	Description
Container standard output	<p>Collects stderr and stdout logs of a specified container in the cluster.</p> <p>NOTE The standard output of the matched container is collected to the specified log stream. Standard output to AOM stops.</p> <ul style="list-style-type: none"> • Output to AOM: ICAgent has been installed on hosts in the cluster and collects container standard output to AOM only. This function is enabled by default. To collect container standard output to LTS, disable this function. • Either Container Standard Output (stdout) or Container Standard Error (stderr) must be enabled. • If you enable Container Standard Error (stderr), select your collection destination path: Collect standard output and standard error to different files (stdout.log and stderr.log) or Collect standard output and standard error to the same file (stdout.log). • Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams.

Parameter	Description
Container file	<p>Collects file logs of a specified container in the cluster.</p> <ul style="list-style-type: none"> Add Collection Path: Add one or more host paths. LTS will collect logs from these paths. For more examples, see Collection Paths. <p>NOTE If a container mount path has been configured for the CCE cluster workload, the paths added for this field are invalid. The collection paths take effect only after the mount path is deleted.</p> <ul style="list-style-type: none"> Add Custom Wrapping Rule: ICAgent determines whether a file is wrapped based on the file name rule. If your wrapping rule does not comply with the built-in rules, you can add a custom wrap rule to prevent log loss during repeated collection and wrapping. The built-in rules are <i>{basename}{connector}{wrapping identifier}.{suffix}</i> and <i>{basename}.{suffix}{connector}{wrapping identifier}</i>. Connectors can be hyphens (-), periods (.), or underscores (_), wrapping identifiers can contain only non-letter characters, and the suffix can contain only letters. A custom wrapping rule consists of <i>{basename}</i> and the feature regular expression of the wrapped file. Example: If your log file name is test.out.log and the names after wrapping are test.2024-01-01.0.out.log and test.2024-01-01.1.out.log, configure the collection path to /opt/*.log, and add a custom wrapping rule: <i>{basename}\.\d{4}-\d{2}-\d{2}\.\d{1}.out.log</i>. Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams. Set Collection Filters: Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.

Parameter	Description
Node file	<p>Collects files of a specified node in a cluster.</p> <ul style="list-style-type: none"> ● Add Collection Path: Add one or more host paths. LTS will collect logs from these paths. For more examples, see Collection Paths. ● Add Custom Wrapping Rule: ICAgent determines whether a file is wrapped based on the file name rule. If your wrapping rule does not comply with the built-in rules, you can add a custom wrap rule to prevent log loss during repeated collection and wrapping. The built-in rules are <i>{basename}{connector}{wrapping identifier}.{suffix}</i> and <i>{basename}.{suffix}{connector}{wrapping identifier}</i>. Connectors can be hyphens (-), periods (.), or underscores (_), wrapping identifiers can contain only non-letter characters, and the suffix can contain only letters. A custom wrapping rule consists of <i>{basename}</i> and the feature regular expression of the wrapped file. Example: If your log file name is test.out.log and the names after wrapping are test.2024-01-01.0.out.log and test.2024-01-01.1.out.log, configure the collection path to /opt/*.log, and add a custom wrapping rule: <i>{basename}\.\d{4}-\d{2}-\d{2}\.\d{1}.out.log</i>. ● Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams. ● Set Collection Filters: Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.
Kubernetes event	<p>Collects event logs of the Kubernetes cluster. Kubernetes events of a Kubernetes cluster can be collected to only one log stream.</p>

3. (Optional) **Kubernetes Matching Rules:** Set these parameters only when the data source type is set to **Container standard output** or **Container file**.

 **NOTE**

After entering a regular expression, click **Verify** to verify it.

Table 2-6 Kubernetes matching rules

Parameter	Description
Namespace Name Regular Expression	<p>Specifies the container whose logs are to be collected based on the namespace name. Regular expression matching is supported.</p> <p>NOTE LTS will collect logs of the namespaces with names matching this expression. To collect logs of all namespaces, leave this field empty.</p>
Pod Name Regular Expression	<p>Specifies the container whose logs are to be collected based on the pod name. Regular expression matching is supported.</p> <p>NOTE LTS will collect logs of the pods with names matching this expression. To collect logs of all pods, leave this field empty.</p>
Container Name Regular Expression	<p>Specifies the container whose logs are to be collected based on the container name (the Kubernetes container name is defined in spec.containers). Regular expression matching is supported.</p> <p>NOTE LTS will collect logs of the containers with names matching this expression. To collect logs of all containers, leave this field empty.</p>
Label Whitelist	<p>Specifies the containers whose logs are to be collected. If you want to set a Kubernetes label whitelist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple whitelists, you can select the And or Or relationship. This means a container will be matched when it satisfies all or any of the whitelists.</p> <p>NOTE If Label Value is empty, LTS will match all containers whose Kubernetes label contains a specified Label Key. If Label Value is not empty, only containers whose Kubernetes label contains a specified Label Key that is equal to its Label Value are matched. Label Key requires full matching while Label Value supports regular matching.</p>
Label Blacklist	<p>Specifies the containers whose logs are not to be collected. If you want to set a Kubernetes label blacklist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple blacklists, you can select the And or Or relationship. This means a container will be excluded when it satisfies all or any of the blacklists.</p> <p>NOTE If Label Value is empty, LTS will exclude all containers whose Kubernetes label contains a specified Label Key. If Label Value is not empty, only containers whose Kubernetes label contains a specified Label Key that is equal to its Label Value will be excluded. Label Key requires full matching while Label Value supports regular matching.</p>

Parameter	Description
Kubernetes Label	<p>After the Kubernetes Label is set, LTS adds related fields to logs.</p> <p>NOTE LTS adds the specified fields to the log when each Label Key has a corresponding Label Value. For example, if you enter "app" as the key and "app_alias" as the value, when the container label contains "app=lhs", "{app_alias: lhs}" will be added to the log.</p>
Container Label Whitelist	<p>Specifies the containers whose logs are to be collected. If you want to set a container label whitelist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple whitelists, you can select the And or Or relationship. This means a container will be matched when it satisfies all or any of the whitelists.</p> <p>NOTE If Label Value is empty, LTS will match all containers whose container label contains a specified Label Key. If Label Value is not empty, only containers whose container label contains a specified Label Key that is equal to its Label Value are matched. Label Key requires full matching while Label Value supports regular matching.</p>
Container Label Blacklist	<p>Specifies the containers whose logs are not to be collected. If you want to set a container label blacklist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple blacklists, you can select the And or Or relationship. This means a container will be excluded when it satisfies all or any of the blacklists.</p> <p>NOTE If Label Value is empty, LTS will exclude all containers whose container label contains a specified Label Key. If Label Value is not empty, only containers whose container label contains a specified Label Key that is equal to its Label Value will be excluded. Label Key requires full matching while Label Value supports regular matching.</p>
Container Label	<p>After the Container Label is set, LTS adds related fields to logs.</p> <p>NOTE LTS adds the specified fields to the log when each Label Key has a corresponding Label Value. For example, if you enter "app" as the key and "app_alias" as the value, when the container label contains "app=lhs", "{app_alias: lhs}" will be added to the log.</p>

Parameter	Description
Environment Variable Whitelist	<p>Specifies the containers whose logs are to be collected. If you want to set an environment variable whitelist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple whitelists, you can select the And or Or relationship. This means a container will be matched when it satisfies all or any of the whitelists.</p> <p>NOTE If Environment Variable Value is empty, LTS will match all containers whose environment variable contains a specified Environment Variable Key. If Environment Variable Value is not empty, only containers whose environment variable contains a specified Environment Variable Key that is equal to its Environment Variable Value are matched. Label Key requires full matching while Label Value supports regular matching.</p>
Environment Variable Blacklist	<p>Specifies the containers whose logs are not to be collected. If you want to set an environment variable blacklist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple blacklists, you can select the And or Or relationship. This means a container will be excluded when it satisfies all or any of the blacklists.</p> <p>NOTE If Environment Variable Value is empty, LTS will exclude all containers whose environment variable contains a specified Environment Variable Key. If Environment Variable Value is not empty, only containers whose environment variable contains a specified Environment Variable Key that is equal to its Environment Variable Value will be excluded. Label Key requires full matching while Label Value supports regular matching.</p>
Environment Variable Label	<p>After the environment variable label is set, the log service adds related fields to the log.</p> <p>NOTE LTS adds the specified fields to the log when each Environment Variable Key has a corresponding Environment Variable Value. For example, if you enter "app" as the key and "app_alias" as the value, when the Kubernetes environment variable contains "app=lhs", "{app_alias: lhs}" will be added to the log.</p>

4. Enable structuring parsing. For details, see [Setting ICAgent Structuring Parsing Rules](#).

LTS enables combined parsing, allowing you to create different structuring parsing rules for each collection configuration of a log stream.

If you have configured cloud structuring parsing, delete its configurations before configuring ICAgent structuring parsing.

5. Set other configurations.

Table 2-7 Other configurations

Parameter	Description
Max Directory Depth	<p>The maximum directory depth is 20 levels.</p> <p>Collection paths can use double asterisks (**) for multi-layer fuzzy match. Specify the maximum directory depth in the text box. For example, if your log path is /var/logs/department/app/a.log and your collection path is /var/logs/**/a.log, logs will not be collected when this parameter is set to 1, but will be collected when this parameter is set to 2 or a larger number.</p>
Split Logs	<ul style="list-style-type: none"> • If log splitting is enabled, logs exceeding the specified size will be split into multiple logs for collection. Specify the size in the range from 500 KB to 1,024 KB. For example, if you set the size to 500 KB, a 600 KB log will be split into a 500 KB log and a 100 KB log. This restriction is applicable to single-line logs only, not multi-line logs. • If log splitting is disabled, when a log exceeds 500 KB, the extra part will be truncated and discarded.
Collect Binary Files	<p>LTS can collect binary files.</p> <p>Run the file -i File_name command to view the file type. charset=binary indicates that a log file is a binary file.</p> <p>If this option is enabled, binary log files will be collected, but only UTF-8 strings are supported. Other strings will be garbled on the LTS console.</p> <p>If this option is disabled, binary log files will not be collected.</p>
Log File Code	<p>The log file encoding format can be UTF-8 or GBK (not available to Windows).</p> <p>UTF-8 encoding is a variable-length encoding mode and represents Unicode character sets. GBK, an acronym for Chinese Internal Code Extension Specification, is a Chinese character encoding standard that extends both the ASCII and GB2312 encoding systems.</p>
Collection Policy	<p>Select incremental or full collection.</p> <ul style="list-style-type: none"> • Incremental: When collecting a new file, ICAgent reads the file from the end of the file. • All: When collecting a new file, ICAgent reads the file from the beginning of the file.

Parameter	Description
Custom Metadata	<ul style="list-style-type: none"> If this option is disabled, ICAgent will report logs to LTS based on the default system fields. You do not need to and cannot configure the fields. If this option is enabled, ICAgent will report logs based on your selected built-in fields and fields created with custom key-value pairs. Built-in Fields: Select built-in fields as required. Custom Key-Value Pairs: Click Add and set a key and value.

6. Configure the log format and time by referring to [Table 2-8](#).

Table 2-8 Log collection settings

Parameter	Description
Log Format	<ul style="list-style-type: none"> Single-line: Each log line is displayed as a single log event. Multi-line: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.
Log Time	<p>System time: log collection time by default. It is displayed at the beginning of each log event.</p> <p>NOTE</p> <ul style="list-style-type: none"> Log collection time is the time when logs are collected and sent by ICAgent to LTS. Log printing time is the time when logs are printed. ICAgent collects and sends logs to LTS with an interval of 1 second. Restriction on log collection time: Logs are collected within 24 hours before and after the system time.

Parameter	Description
	<p>Time wildcard: You can set a time wildcard so that ICAgent will look for the log printing time as the beginning of a log event.</p> <ul style="list-style-type: none"> • If the time format in a log event is 2019-01-01 23:59:59.011, the time wildcard should be set to YYYY-MM-DD hh:mm:ss.SSS. • If the time format in a log event is 19-1-1 23:59:59.011, the time wildcard should be set to YY-M-D hh:mm:ss.SSS. <p>NOTE If a log event does not contain year information, ICAgent regards it as printed in the current year.</p> <p>Example:</p> <pre> YY - year (19) YYYY - year (2019) M - month (1) MM - month (01) D - day (1) DD - day (01) hh - hours (23) mm - minutes (59) ss - seconds (59) SSS - millisecond (999) hpm - hours (03PM) h:mmpm - hours:minutes (03:04PM) h:mm:sspm - hours:minutes:seconds (03:04:05PM) hh:mm:ss ZZZZ (16:05:06 +0100) hh:mm:ss ZZZ (16:05:06 CET) hh:mm:ss ZZ (16:05:06 +01:00) </pre>
Log Segmentation	<p>This parameter needs to be specified if the Log Format is set to Multi-line. By generation time indicates that a time wildcard is used to detect log boundaries, whereas By regular expression indicates that a regular expression is used.</p>
By regular expression	<p>You can set a regular expression to look for a specific pattern to indicate the beginning of a log event. This parameter needs to be specified when you select Multi-line for Log Format and By regular expression for Log Segmentation.</p>

 **NOTE**

The time wildcard and regular expression will look for the specified pattern right from the beginning of each log line. If no match is found, the system time, which may be different from the time in the log event, is used. In general cases, you are advised to select **Single-line** for **Log Format** and **System time** for **Log Time**.

Step 5: Configure Indexing

1. (Optional) Configure indexing. For details, see [Setting Indexes](#).

2. Click **Submit**.

Step 6: Complete the Ingestion Configuration

The created ingestion configuration will be displayed.

- Click its name to view its details.
- Click **Modify** in the **Operation** column to modify the ingestion configuration.
- Click **Configure Tag** in the **Operation** column to add a tag.
- Click **More > Copy** in the **Operation** column to copy the ingestion configuration.
- Click **More > Delete** in the **Operation** column to delete the ingestion configuration.

NOTE

Deleting an ingestion configuration may lead to log collection failures, potentially resulting in service exceptions related to user logs. In addition, the deleted ingestion configuration cannot be restored. Exercise caution when performing this operation.

Setting Multiple Ingestion Configurations in a Batch

You can set multiple ingestion configurations for multiple scenarios in a batch, avoiding repetitive setups.

- Step 1** On the **Ingestion Management** page, click **Batch Ingest** to go to the details page. For details, see [Table 2-9](#).

Table 2-9 Adding configurations in batches

Type	Parameter	Description
Basic Settings	Ingestion Type	Select CCE (Cloud Container Engine) .
	Configurations to Add	Enter the number of ingestion configurations in the text box and click Add . A maximum of 100 ingestion configurations can be added, including the one already exists under Ingestion Settings by default. Therefore, you can add up to 99 more.
Ingestion Settings	Configuration List	<ol style="list-style-type: none"> 1. The ingestion configurations are displayed on the left. You can add up to 99 more configurations. 2. The ingestion configuration details are displayed on the right. Set them by referring to Step 4: Configure the Collection. 3. After an ingestion configuration is complete, you can click Apply to Other Configurations to copy its settings to other configurations.

- Step 2** Click **Check Parameters**. After the check is successful, click **Submit**.

The added ingestion configurations will be displayed on the **Ingestion Management** page after the batch creation is successful.

Step 3 (Optional) Perform the following operations on ingestion configurations:

- Select multiple existing ingestion configurations and click **Edit**. On the displayed page, select an ingestion type to modify the corresponding ingestion configurations.
- Select multiple existing ingestion configurations and click **Enable** or **Disable**. If you toggle off the switch in the **Status** column of an ingestion configuration, logs will not be collected for this configuration.
- Select multiple existing ingestion configurations and click **Delete**.

----End

2.1.6 Ingesting CFW Logs to LTS

LTS can collect logs from CFW. It also enables real-time, efficient, and secure analysis and processing of a vast volume of collected logs.

For details, see [Log Settings](#).

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.7 Ingesting CTS Logs to LTS

LTS can collect logs from CTS.

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Configuring CTS Log Ingestion in LTS

Perform the following operations to configure CTS log ingestion:

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**.

Step 2 Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **CTS (Cloud Trace Service)**.

Step 3 Select a log stream.

1. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
2. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
3. Click **Next: Configure CTS**.

Step 4 Click **Configure CTS**.

Step 5 Click **Next: Configure Log Stream**.

Table 2-10 Log stream parameters

Parameter	Description
Auto Structure and Index	If this function is enabled, the structuring for the log stream is based on the CTS system template, and the indexing enables quick analysis for all parsed CTS fields.

Step 6 Click **Submit**.

----End

2.1.8 Ingesting GaussDB(DWS) Logs to LTS

LTS can collect logs from GaussDB(DWS).

For details, see [Cluster Log Management](#).

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.9 Ingesting ECS Text Logs to LTS

Elastic Cloud Server (ECS) provides scalable, on-demand cloud servers to build secure, flexible, and efficient environment for your applications.

After you configure ECS log ingestion, ICAgent collects logs from ECSs (hosts) based on your specified rules, and sends the logs to LTS by log stream. You can view and analyze these logs on the LTS console for improving host running stability and information security.

Perform the following steps to complete the ingestion configuration:

1. [Step 1: Select a Log Stream](#)
2. [Step 2: \(Optional\) Select a Host Group](#)
3. [Step 3: Configure the Collection](#)
4. [Step 4: Configure Indexing](#)
5. [Step 5: Complete the Ingestion Configuration](#)


To collect logs from multiple scenarios, [set multiple ingestion configurations in a batch](#).

Prerequisites

ICAgent has been [installed](#) and [added](#) to the host group.

Step 1: Select a Log Stream

1. Log in to the management console and choose **Management & Deployment > Log Tank Service**.

2. Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **ECS (Elastic Cloud Server)**.
Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > ECS (Elastic Cloud Server)**.
Alternatively, choose **Log Management** in the navigation pane and click the target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and click **Ingest Log**. In the displayed dialog box, click **ECS (Elastic Cloud Server)**.
3. Select a log group from the **Log Group** drop-down list. If there is no desired log group, click **Create Log Group**. For details, see [Managing Log Groups](#).
4. Select a log stream from the **Log Stream** drop-down list. If there is no desired log stream, click **Create Log Stream**. For details, see [Managing Log Streams](#).
5. Click **Next: (Optional) Select Host Group**.

Step 2: (Optional) Select a Host Group

1. Select one or more host groups from which you want to collect logs. If there are no desired host groups, click **Create** above the host group list to create one. For details, see [Managing Host Groups](#).

NOTE

You can also skip this step, but the collection configuration will not take effect. You are advised to select a host group during the first ingestion configuration. If you skip this step, follow either of the following ways to configure host groups after the ingestion configuration is complete:

- Choose **Host Management > Host Groups** in the navigation pane and associate host groups with ingestion configurations.
 - Choose **Log Ingestion > Ingestion Management** in the navigation pane. In the ingestion configuration list, click **Modify** in the **Operation** column. On the page displayed, select required host groups.
2. Click **Next: Configurations**.

Step 3: Configure the Collection

After selecting host groups, configure the collection as follows:

NOTE

- Ensure that sensitive information is not collected.
 - If a collection path of a host has been configured in AOM, do not configure the path in LTS.
 - If log files were last modified more than 12 hours earlier than the time when the path is added, the files are not collected.
1. **Collection Configuration Name:** Enter 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Do not start with a period or underscore, or end with a period.
If you want to reuse existing collection configurations, click **Import Configuration** next to the text box. On the **Import Configuration** page, select a configuration and click **OK**.

 NOTE

Import Old-Edition Configuration: Import the host ingestion configuration of the old version to the log ingestion of the new version.

- If LTS is newly installed and **Import Old-Edition Configuration** is not displayed, you can directly create a configuration without importing the old one.
- If LTS is upgraded, **Import Old-Edition Configuration** is displayed. Import the old configuration or create one as required.

2. **Collection Paths:** Add one or more host paths. LTS will collect logs from these paths. The rules for setting collection paths are as follows:

- Logs can be collected recursively. A double asterisk (**) can represent up to 5 directory levels in a path.

For example, **/var/logs/**/a.log** will match the following logs:

```
/var/logs/a.log
/var/logs/1/a.log
/var/logs/1/2/a.log
/var/logs/1/2/3/a.log
/var/logs/1/2/3/4/a.log
/var/logs/1/2/3/4/5/a.log
```

 NOTE

- **/1/2/3/4/5/** indicates the 5 levels of directories under the **/var/logs** directory. All the **a.log** files found in all these levels of directories will be collected.
 - Only one double asterisk (**) can be contained in a collection path. For example, **/var/logs/**/a.log** is acceptable but **/opt/test/**/log/**** is not.
 - A collection path cannot begin with a double asterisk (**), such as ****/test**, to avoid collecting system files.
- You can use an asterisk (*) as a wildcard for fuzzy match. The wildcard (*) can represent one or more characters of a directory or file name.

 NOTE

If a log collection path is similar to **C:\windows\system32** but logs cannot be collected, enable Web Application Firewall (WAF) and configure the path again.

- Example 1: **/var/logs/*/a.log** will match all **a.log** files found in all directories under the **/var/logs/** directory:


```
/var/logs/1/a.log
/var/logs/2/a.log
```
- Example 2: **/var/logs/service-*/a.log** will match files as follows:


```
/var/logs/service-1/a.log
/var/logs/service-2/a.log
```
- Example 3: **/var/logs/service/a*.log** will match files as follows:


```
/var/logs/service/a1.log
/var/logs/service/a2.log
```
- If the collection path is set to a file name, the corresponding file is collected. Only text files can be collected.
- **Add Custom Wrapping Rule:** ICAgent determines whether a file is wrapped based on the file name rule. If your wrapping rule does not comply with the built-in rules, you can add a custom wrap rule to prevent log loss during repeated collection and wrapping.

The built-in rules are $\{baseline\}\{connector\}\{wrapping\ identifier\}.\{suffix\}$ and $\{baseline\}.\{suffix\}\{connector\}\{wrapping\ identifier\}$. Connectors can be hyphens (-), periods (.), or underscores (_), wrapping identifiers can contain only non-letter characters, and the suffix can contain only letters.

A custom wrapping rule consists of $\{baseline\}$ and the feature regular expression of the wrapped file. Example: If your log file name is **test.out.log** and the names after wrapping are **test.2024-01-01.0.out.log** and **test.2024-01-01.1.out.log**, configure the collection path to **/opt/* \cdot log**, and add a custom wrapping rule: $\{baseline\}\.\d{4}\-\d{2}\-\d{2}\.\d{1}\.out.log$.

3. **Allow Repeated File Collection** (not available to Windows)

After you enable this function, one host log file can be collected to multiple log streams.

After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams.

4. **Set Collection Filters:** Blacklisted directories or files will not be collected.

Blacklist filters can be exact matches or wildcard pattern matches. For details, see [Collection Paths](#).

 **NOTE**

- If you blacklist a file or directory that has been set as a collection path in the previous step, the blacklist settings will be used and the file or files in the directory will be filtered out.
 - If a log has been added to the blacklist, it cannot be collected even if you create a log ingestion task. You can collect it again only after you delete the collection path from the blacklist.
 - If you specify a directory, all files in the directory are filtered out, but log files in the folders in the directory cannot be filtered out.
5. **Collect Windows Event Logs:** To collect logs from Windows hosts, enable this option and set the following parameters.

Table 2-11 Parameters for collecting windows event logs

Parameter	Description
Log Type	Log types include System, Application, Security, and Startup .
First Collection Time Offset	If you set this parameter to 7 , logs generated within the 7 days before the collection start time are collected. This offset takes effect only for the first collection to ensure that the logs are not repeatedly collected. The maximum value is 7 days.
Event Level	You can filter and collect Windows events based on their severity (information, warning, error, critical, and verbose). This function is available only to Windows Vista or later.

6. Enable structuring parsing. For details, see [Setting ICAgent Structuring Parsing Rules](#).

LTS enables combined parsing, allowing you to create different structuring parsing rules for each collection configuration of a log stream.

If you have configured cloud structuring parsing, delete its configurations before configuring ICAgent structuring parsing.

7. Configure the log format and time by referring to [Table 2-12](#).

Table 2-12 Log collection settings

Parameter	Description
Log Format	<ul style="list-style-type: none"> ● Single-line: Each log line is displayed as a single log event. ● Multi-line: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.
Log Time	<p>System time: log collection time by default. It is displayed at the beginning of each log event.</p> <p>NOTE</p> <ul style="list-style-type: none"> ● Log collection time is the time when logs are collected and sent by ICAgent to LTS. ● Log printing time is the time when logs are printed. ICAgent collects and sends logs to LTS with an interval of 1 second. ● Restriction on log collection time: Logs are collected within 24 hours before and after the system time. <p>Time wildcard: You can set a time wildcard so that ICAgent will look for the log printing time as the beginning of a log event.</p> <ul style="list-style-type: none"> ● If the time format in a log event is 2019-01-01 23:59:59.011, the time wildcard should be set to YYYY-MM-DD hh:mm:ss.SSS. ● If the time format in a log event is 19-1-1 23:59:59.011, the time wildcard should be set to YY-M-D hh:mm:ss.SSS. <p>NOTE</p> <p>If a log event does not contain year information, ICAgent regards it as printed in the current year.</p> <p>Example:</p> <pre> YY - year (19) YYYY - year (2019) M - month (1) MM - month (01) D - day (1) DD - day (01) hh - hours (23) mm - minutes (59) ss - seconds (59) SSS - millisecond (999) hpm - hours (03PM) h:mmpm - hours:minutes (03:04PM) h:mm:sspm - hours:minutes:seconds (03:04:05PM) hh:mm:ss ZZZZ (16:05:06 +0100) hh:mm:ss ZZZ (16:05:06 CET) hh:mm:ss ZZ (16:05:06 +01:00) </pre>

Parameter	Description
Log Segmentation	This parameter needs to be specified if the Log Format is set to Multi-line . By generation time indicates that a time wildcard is used to detect log boundaries, whereas By regular expression indicates that a regular expression is used.
By regular expression	You can set a regular expression to look for a specific pattern to indicate the beginning of a log event. This parameter needs to be specified when you select Multi-line for Log Format and By regular expression for Log Segmentation .

 **NOTE**

The time wildcard and regular expression will look for the specified pattern right from the beginning of each log line. If no match is found, the system time, which may be different from the time in the log event, is used. In general cases, you are advised to select **Single-line** for **Log Format** and **System time** for **Log Time**.

Step 4: Configure Indexing

1. (Optional) Configure indexing. For details, see [Setting Indexes](#).
2. Click **Submit**.

Step 5: Complete the Ingestion Configuration

The created ingestion configuration will be displayed.

- Click its name to view its details.
- Click **Modify** in the **Operation** column to modify the ingestion configuration.
- Click **Configure Tag** in the **Operation** column to add a tag.
- Click **More > Copy** in the **Operation** column to copy the ingestion configuration.
- Click **More > Delete** in the **Operation** column to delete the ingestion configuration.

 **NOTE**

Deleting an ingestion configuration may lead to log collection failures, potentially resulting in service exceptions related to user logs. In addition, the deleted ingestion configuration cannot be restored. Exercise caution when performing this operation.

Setting Multiple Ingestion Configurations in a Batch

You can set multiple ingestion configurations for multiple scenarios in a batch, avoiding repetitive setups.

- Step 1** On the **Ingestion Management** page, click **Batch Ingest** to go to the details page. For details, see [Table 2-13](#).

Table 2-13 Adding configurations in batches

Type	Parameter	Description
Basic Settings	Ingestion Type	Select ECS (Elastic Cloud Server) .
	Configurations to Add	Enter the number of ingestion configurations in the text box and click Add . A maximum of 100 ingestion configurations can be added, including the one already exists under Ingestion Settings by default. Therefore, you can add up to 99 more.
Ingestion Settings	Configuration List	<ol style="list-style-type: none"> The ingestion configurations are displayed on the left. You can add up to 99 more configurations. The ingestion configuration details are displayed on the right. Set them by referring to Step 3: Configure the Collection. After an ingestion configuration is complete, you can click Apply to Other Configurations to copy its settings to other configurations.

Step 2 Click **Check Parameters**. After the check is successful, click **Submit**.

The added ingestion configurations will be displayed on the **Ingestion Management** page after the batch creation is successful.

Step 3 (Optional) Perform the following operations on ingestion configurations:

- Select multiple existing ingestion configurations and click **Edit**. On the displayed page, select an ingestion type to modify the corresponding ingestion configurations.
- Select multiple existing ingestion configurations and click **Enable** or **Disable**. If you toggle off the switch in the **Status** column of an ingestion configuration, logs will not be collected for this configuration.
- Select multiple existing ingestion configurations and click **Delete**.

----End

2.1.10 Ingesting ELB Logs to LTS

LTS can collect logs from ELB.

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Prerequisites

You have created and used a load balancer.

Constraints

ELB access logs only record requests sent to layer 7 dedicated and shared load balancers. Layer 4 shared load balancing is not logged.

Configuring ELB Log Ingestion

Perform the following operations to configure ELB log ingestion:

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **ELB (Elastic Load Balance)**.
- Step 3** Select a log stream.
 1. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
 2. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
 3. Click **Next: Configure ELB**.
- Step 4** Click **Configure ELB**.
- Step 5** Configure a log stream.

Table 2-14 Log stream parameters

Parameter	Description
Auto Structure and Index	If this function is enabled, the structuring for the log stream is based on the ELB system template, and the indexing enables quick analysis for all parsed ELB fields.

- Step 6** Click **Submit**.

----End

2.1.11 Ingesting Enterprise Router Logs to LTS

LTS can collect logs from Enterprise Router.

For details, see [Creating a Flow Log](#).

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.12 Ingesting FunctionGraph Logs to LTS

LTS can collect logs from FunctionGraph.

For details, see [Managing Function Logs](#).

 NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.13 Ingesting ModelArts Logs to LTS

LTS can collect logs from the AI development platform ModelArts.

For details, see [Deploying as a Real-Time Service](#).

 NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.14 Ingesting SMN Logs to LTS

LTS can collect logs from SMN.

For details, see [Logs](#).

 NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.15 Ingesting SecMaster Logs to LTS

LTS can collect logs from SecMaster.

For details, see [Creating a Data Delivery](#).

 NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.1.16 Ingesting ServiceStage Containerized Application Logs to LTS

LTS collects log data from containerized applications of ServiceStage. By processing a massive number of logs efficiently, securely, and in real time, LTS provides useful insights for you to optimize the availability and performance of cloud services and applications. It also helps you efficiently perform real-time decision-making, device O&M management, and service trend analysis.

Perform the following steps to complete the ingestion configuration:

1. [Step 1: Select a Log Stream](#)
2. [Step 2: Check Dependencies](#)
3. [Step 3: \(Optional\) Select a Host Group](#)
4. [Step 4: Configure the Collection](#)
5. [Step 5: Configure Indexing](#)
6. [Step 6: Complete the Ingestion Configuration](#)

To collect logs from multiple scenarios, [set multiple ingestion configurations in a batch](#).

 **NOTE**

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Prerequisites


- ICAgent has been installed and added to the host group.
- A ServiceStage application has been created.
- A ServiceStage environment has been created.
- A ServiceStage component has been created.

Constraints

- CCE cluster nodes whose container engine is Docker are supported.
- CCE cluster nodes whose container engine is Containerd are supported. You must be using ICAgent 5.12.130 or later.
- To collect container log directories mounted to host directories to LTS, you must configure the node file path.
- Constraints on the Docker storage driver: Currently, container file log collection supports only the overlay2 storage driver. devicemapper cannot be used as the storage driver. Run the following command to check the storage driver type:

```
docker info | grep "Storage Driver"
```

Step 1: Select a Log Stream

1. Log in to the management console and choose **Management & Deployment > Log Tank Service**.
2. Choose **Log Ingestion > Ingestion Center** in the navigation pane. Then, click **ServiceStage - Containerized Application Logs** under **Self-Built software**.
Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > ServiceStage - Containerized Application Logs**.
Alternatively, choose **Log Management** in the navigation pane and click the target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and click **Ingest Log**. In the displayed dialog box, click **ServiceStage - Containerized Application Logs** under **Self-Built software**.
3. In the **Select Log Stream** step, set the following parameters:
 - a. Select a ServiceStage application and ServiceStage environment.
 - b. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
 - c. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
4. Click **Next: Check Dependencies**.

Step 2: Check Dependencies

1. The system automatically checks whether there is a host group with the custom identifier **k8s-log-Application ID**.

There is a log group named **k8s-log-Application ID**.

If not, click **Auto Correct**.

NOTE

- **Auto Correct:** Configure dependencies with one click.
 - **Check Again:** Recheck dependencies.
2. Click **Next: (Optional) Select Host Group**.

Step 3: (Optional) Select a Host Group

1. In the host group list, the host group to which the cluster belongs is selected by default. You can also select host groups as required.
2. Click **Next: Configurations**.

Step 4: Configure the Collection

When you configure ServiceStage containerized application log ingestion, the collection configuration details are as follows.

1. **Collection Configuration Name:** Enter 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Do not start with a period or underscore, or end with a period.
2. **Data Source:** Select a data source type and configure it. The following data source types are supported: container standard output, container file path, node file path, and K8s event.

Table 2-15 Collection configuration parameters

Type	Description
Container standard output	<p>Collects stderr and stdout logs of a specified container in the cluster. Either Container Standard Output (stdout) or Container Standard Error (stderr) must be enabled.</p> <ul style="list-style-type: none"> • If you enable Container Standard Error (stderr), select your collection destination path: Collect standard output and standard error to different files (stdout.log and stderr.log) or Collect standard output and standard error to the same file (stdout.log). • The standard output of the matched container is collected to the specified log stream. Standard output to AOM stops. • Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams.

Type	Description
Container file	<p>Collects file logs of a specified container in the cluster.</p> <ul style="list-style-type: none"> ● Collection Paths: Add one or more host paths. LTS will collect logs from these paths. For details, see 2. <p>NOTE If a container mount path has been configured for the CCE cluster workload, the paths added for this field are invalid. The collection paths take effect only after the mount path is deleted.</p> <ul style="list-style-type: none"> ● Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams. ● Set Collection Filters: Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.
Node file	<p>Collects files of a specified node in a cluster.</p> <ul style="list-style-type: none"> ● Collection Paths: Add one or more host paths. LTS will collect logs from these paths. ● Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams. ● Set Collection Filters: Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.
Kubernetes event	<p>Collects event logs of the Kubernetes cluster. You do not need to set parameters. Only ICAgent 5.12.150 or later is supported.</p> <p>NOTE Kubernetes events of a Kubernetes cluster can be collected to only one log stream.</p>

3. If you select **Container standard output** or **Container file** as the data source type, set the ServiceStage matching rule by selecting the corresponding component from the drop-down list.
4. Enable structuring parsing. For details, see [Setting ICAgent Structuring Parsing Rules](#).
LTS enables combined parsing, allowing you to create different structuring parsing rules for each collection configuration of a log stream.
If you have configured cloud structuring parsing, delete its configurations before configuring ICAgent structuring parsing.
5. Set other configurations.

Table 2-16 Other configurations

Parameter	Description
Max Directory Depth	<p>The maximum directory depth is 20 levels.</p> <p>Collection paths can use double asterisks (**) for multi-layer fuzzy match. Specify the maximum directory depth in the text box. For example, if your log path is /var/logs/department/app/a.log and your collection path is /var/logs/**/a.log, logs will not be collected when this parameter is set to 1, but will be collected when this parameter is set to 2 or a larger number.</p>
Split Logs	<ul style="list-style-type: none"> • If log splitting is enabled, logs exceeding the specified size will be split into multiple logs for collection. Specify the size in the range from 500 KB to 1,024 KB. For example, if you set the size to 500 KB, a 600 KB log will be split into a 500 KB log and a 100 KB log. This restriction is applicable to single-line logs only, not multi-line logs. • If log splitting is disabled, when a log exceeds 500 KB, the extra part will be truncated and discarded.
Collect Binary Files	<p>LTS can collect binary files.</p> <p>Run the file -i File_name command to view the file type. charset=binary indicates that a log file is a binary file.</p> <p>If this option is enabled, binary log files will be collected, but only UTF-8 strings are supported. Other strings will be garbled on the LTS console.</p> <p>If this option is disabled, binary log files will not be collected.</p>
Log File Code	<p>The log file encoding format can be UTF-8 or GBK (not available to Windows).</p> <p>UTF-8 encoding is a variable-length encoding mode and represents Unicode character sets. GBK, an acronym for Chinese Internal Code Extension Specification, is a Chinese character encoding standard that extends both the ASCII and GB2312 encoding systems.</p>
Collection Policy	<p>Select incremental or full collection.</p> <ul style="list-style-type: none"> • Incremental: When collecting a new file, ICAgent reads the file from the end of the file. • All: When collecting a new file, ICAgent reads the file from the beginning of the file.

Parameter	Description
Custom Metadata	<ul style="list-style-type: none"> If this option is disabled, ICAgent will report logs to LTS based on the default system fields. You do not need to and cannot configure the fields. If this option is enabled, ICAgent will report logs based on your selected built-in fields and fields created with custom key-value pairs. Built-in Fields: Select built-in fields as required. Custom Key-Value Pairs: Click Add and set a key and value.

6. Configure the log format and time by referring to [Table 2-17](#).

Table 2-17 Log collection settings

Parameter	Description
Log Format	<ul style="list-style-type: none"> Single-line: Each log line is displayed as a single log event. Multi-line: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.
Log Time	<p>System time: log collection time by default. It is displayed at the beginning of each log event.</p> <p>NOTE</p> <ul style="list-style-type: none"> Log collection time is the time when logs are collected and sent by ICAgent to LTS. Log printing time is the time when logs are printed. ICAgent collects and sends logs to LTS with an interval of 1 second. Restriction on log collection time: Logs are collected within 24 hours before and after the system time.

Parameter	Description
	<p>Time wildcard: You can set a time wildcard so that ICAgent will look for the log printing time as the beginning of a log event.</p> <ul style="list-style-type: none"> • If the time format in a log event is 2019-01-01 23:59:59.011, the time wildcard should be set to YYYY-MM-DD hh:mm:ss.SSS. • If the time format in a log event is 19-1-1 23:59:59.011, the time wildcard should be set to YY-M-D hh:mm:ss.SSS. <p>NOTE If a log event does not contain year information, ICAgent regards it as printed in the current year.</p> <p>Example:</p> <pre> YY - year (19) YYYY - year (2019) M - month (1) MM - month (01) D - day (1) DD - day (01) hh - hours (23) mm - minutes (59) ss - seconds (59) SSS - millisecond (999) hpm - hours (03PM) h:mmpm - hours:minutes (03:04PM) h:mm:sspm - hours:minutes:seconds (03:04:05PM) hh:mm:ss ZZZZ (16:05:06 +0100) hh:mm:ss ZZZ (16:05:06 CET) hh:mm:ss ZZ (16:05:06 +01:00) </pre>
Log Segmentation	<p>This parameter needs to be specified if the Log Format is set to Multi-line. By generation time indicates that a time wildcard is used to detect log boundaries, whereas By regular expression indicates that a regular expression is used.</p>
By regular expression	<p>You can set a regular expression to look for a specific pattern to indicate the beginning of a log event. This parameter needs to be specified when you select Multi-line for Log Format and By regular expression for Log Segmentation.</p>

Step 5: Configure Indexing

1. (Optional) Configure indexing. For details, see [Setting Indexes](#).
2. Click **Submit**.

Step 6: Complete the Ingestion Configuration

The created ingestion configuration will be displayed.

- Click its name to view its details.
- Click **Modify** in the **Operation** column to modify the ingestion configuration.

- Click **Configure Tag** in the **Operation** column to add a tag.
- Click **More > Copy** in the **Operation** column to copy the ingestion configuration.
- Click **More > Delete** in the **Operation** column to delete the ingestion configuration.

 **NOTE**

Deleting an ingestion configuration may lead to log collection failures, potentially resulting in service exceptions related to user logs. In addition, the deleted ingestion configuration cannot be restored. Exercise caution when performing this operation.

Setting Multiple Ingestion Configurations in a Batch

You can set multiple ingestion configurations for multiple scenarios in a batch, avoiding repetitive setups.

- Step 1** On the **Ingestion Management** page, click **Batch Ingest** to go to the details page. For details, see [Table 2-18](#).

Table 2-18 Adding configurations in batches

Type	Parameter	Description
Basic Settings	Ingestion Type	Select ServiceStage - Containerized Application Logs .
	Configurations to Add	Enter the number of ingestion configurations in the text box and click Add . A maximum of 100 ingestion configurations can be added, including the one already exists under Ingestion Settings by default. Therefore, you can add up to 99 more.
Ingestion Settings	Configuration List	<ol style="list-style-type: none"> 1. The ingestion configurations are displayed on the left. You can add up to 99 more configurations. 2. The ingestion configuration details are displayed on the right. Set them by referring to Step 4: Configure the Collection. 3. After an ingestion configuration is complete, you can click Apply to Other Configurations to copy its settings to other configurations.

- Step 2** Click **Check Parameters**. After the check is successful, click **Submit**.

The added ingestion configurations will be displayed on the **Ingestion Management** page after the batch creation is successful.

- Step 3** (Optional) Perform the following operations on ingestion configurations:

- Select multiple existing ingestion configurations and click **Edit**. On the displayed page, select an ingestion type to modify the corresponding ingestion configurations.

- Select multiple existing ingestion configurations and click **Enable** or **Disable**. If you toggle off the switch in the **Status** column of an ingestion configuration, logs will not be collected for this configuration.
- Select multiple existing ingestion configurations and click **Delete**.

----End

2.1.17 Ingesting ServiceStage Cloud Host Logs to LTS

LTS collects log data from cloud hosts of ServiceStage. By processing a massive number of logs efficiently, securely, and in real time, LTS provides useful insights for you to optimize the availability and performance of cloud services and applications. It also helps you efficiently perform real-time decision-making, device O&M management, and service trend analysis.

Perform the following steps to complete the ingestion configuration:

1. [Step 1: Select a Log Stream](#)
2. [Step 2: \(Optional\) Select a Host Group](#)
3. [Step 3: Configure the Collection](#)
4. [Step 4: Configure Indexing](#)
5. [Step 5: Complete the Ingestion Configuration](#)

Prerequisites


- ICAgent has been installed and added to the host group.
- A ServiceStage application has been created.
- A ServiceStage environment has been created.
- A ServiceStage component has been created.

Step 1: Select a Log Stream

1. Log in to the management console and choose **Management & Deployment > Log Tank Service**.
2. Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **ServiceStage - Cloud Host Logs**.

Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > ServiceStage - Cloud Host Logs**.

Alternatively, choose **Log Management** in the navigation pane and click the

target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and click **Ingest Log**. In the displayed dialog box, click **ServiceStage - Cloud Host Logs**.

3. In the **Select Log Stream** step, set the following parameters:
 - a. Select a ServiceStage application and ServiceStage environment.
 - b. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
 - c. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.

4. Click **Next: (Optional) Select Host Group**.

Step 2: (Optional) Select a Host Group

1. Select one or more host groups from which you want to collect logs. If there are no desired host groups, click **Create** above the host group list to create one.

NOTE

You can skip this step and configure host groups after the ingestion configuration is complete. There are two ways to do this:

- Choose **Host Management > Host Groups** in the navigation pane and associate host groups with ingestion configurations.
 - Choose **Log Ingestion > Ingestion Management** in the navigation pane. In the ingestion configuration list, click **Modify** in the **Operation** column. On the page displayed, select required host groups.
2. Click **Next: Configurations**.

Step 3: Configure the Collection

Perform the following steps to configure the collection:

1. **Collection Configuration Name:** Enter 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Do not start with a period or underscore, or end with a period.
2. **Collection Paths:** Add one or more host paths. LTS will collect logs from these paths. The rules for setting collection paths are as follows:
 - Logs can be collected recursively. A double asterisk (**) can represent up to 5 directory levels in a path.

For example, **/var/logs/**/a.log** will match the following logs:

```
/var/logs/a.log  
/var/logs/1/a.log  
/var/logs/1/2/a.log  
/var/logs/1/2/3/a.log  
/var/logs/1/2/3/4/a.log  
/var/logs/1/2/3/4/5/a.log
```

NOTE

- **/1/2/3/4/5/** indicates the 5 levels of directories under the **/var/logs** directory. All the **a.log** files found in all these levels of directories will be collected.
 - Only one double asterisk (**) can be contained in a collection path. For example, **/var/logs/**/a.log** is acceptable but **/opt/test/**/log/**** is not.
 - A collection path cannot begin with a double asterisk (**), such as **/**/test**, to avoid collecting system files.
- You can use an asterisk (*) as a wildcard for fuzzy match. The wildcard (*) can represent one or more characters of a directory or file name.

NOTE

If a log collection path is similar to **C:\windows\system32** but logs cannot be collected, enable WAF and configure the path again.

- Example 1: **/var/logs/*/a.log** will match all **a.log** files found in all directories under the **/var/logs/** directory:
`/var/logs/1/a.log`

/var/logs/2/a.log

- Example 2: **/var/logs/service-*/a.log** will match files as follows:

/var/logs/service-1/a.log

/var/logs/service-2/a.log

- Example 3: **/var/logs/service/a*.log** will match files as follows:

/var/logs/service/a1.log

/var/logs/service/a2.log

- If the collection path is set to a file name, the corresponding file is collected. Only text files can be collected.

3. **Allow Repeated File Collection** (not available to Windows)

After you enable this function, one host log file can be collected to multiple log streams.

After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams.

4. **Set Collection Filters:** Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.

Blacklist filters can be exact matches or wildcard pattern matches. For details, see [Collection Paths](#).

 **NOTE**

If you blacklist a file or directory that has been set as a collection path in the previous step, the blacklist settings will be used and the file or files in the directory will be filtered out.

5. **Collect Windows Event Logs:** To collect logs from Windows hosts, enable this option and set the following parameters.

Table 2-19 Parameters for collecting windows event logs

Parameter	Description
Log Type	Log types include System, Application, Security, and Startup .
First Collection Time Offset	If you set this parameter to 7 , logs generated within the 7 days before the collection start time are collected. This offset takes effect only for the first collection to ensure that the logs are not repeatedly collected. The maximum value is 7 days.
Event Level	You can filter and collect Windows events based on their severity (information, warning, error, critical, and verbose). This function is available only to Windows Vista or later.

6. Set the ServiceStage matching rule by selecting the corresponding component.
7. Set other configurations.

Table 2-20 Other configurations

Parameter	Description
Max Directory Depth	<p>The maximum directory depth is 20 levels.</p> <p>Collection paths can use double asterisks (**) for multi-layer fuzzy match. Specify the maximum directory depth in the text box. For example, if your log path is <code>/var/logs/department/app/a.log</code> and your collection path is <code>/var/logs/**/a.log</code>, logs will not be collected when this parameter is set to 1, but will be collected when this parameter is set to 2 or a larger number.</p>
Split Logs	<ul style="list-style-type: none"> • If log splitting is enabled, logs exceeding the specified size will be split into multiple logs for collection. Specify the size in the range from 500 KB to 1,024 KB. For example, if you set the size to 500 KB, a 600 KB log will be split into a 500 KB log and a 100 KB log. This restriction is applicable to single-line logs only, not multi-line logs. • If log splitting is disabled, when a log exceeds 500 KB, the extra part will be truncated and discarded.
Collect Binary Files	<p>LTS can collect binary files.</p> <p>Run the <code>file -i File_name</code> command to view the file type. charset=binary indicates that a log file is a binary file.</p> <p>If this option is enabled, binary log files will be collected, but only UTF-8 strings are supported. Other strings will be garbled on the LTS console.</p> <p>If this option is disabled, binary log files will not be collected.</p>
Log File Code	<p>The log file encoding format can be UTF-8 or GBK (not available to Windows).</p> <p>UTF-8 encoding is a variable-length encoding mode and represents Unicode character sets. GBK, an acronym for Chinese Internal Code Extension Specification, is a Chinese character encoding standard that extends both the ASCII and GB2312 encoding systems.</p>
Collection Policy	<p>Select incremental or full collection.</p> <ul style="list-style-type: none"> • Incremental: When collecting a new file, ICAgent reads the file from the end of the file. • All: When collecting a new file, ICAgent reads the file from the beginning of the file.

8. Configure the log format and time by referring to [Table 2-21](#).

Table 2-21 Log collection settings

Parameter	Description
Log Format	<ul style="list-style-type: none"> ● Single-line: Each log line is displayed as a single log event. ● Multi-line: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.
Log Time	<p>System time: log collection time by default. It is displayed at the beginning of each log event.</p> <p>NOTE</p> <ul style="list-style-type: none"> ● Log collection time is the time when logs are collected and sent by ICAgent to LTS. ● Log printing time is the time when logs are printed. ICAgent collects and sends logs to LTS with an interval of 1 second. ● Restriction on log collection time: Logs are collected within 24 hours before and after the system time. <p>Time wildcard: You can set a time wildcard so that ICAgent will look for the log printing time as the beginning of a log event.</p> <ul style="list-style-type: none"> ● If the time format in a log event is 2019-01-01 23:59:59.011, the time wildcard should be set to YYYY-MM-DD hh:mm:ss.SSS. ● If the time format in a log event is 19-1-1 23:59:59.011, the time wildcard should be set to YY-M-D hh:mm:ss.SSS. <p>NOTE If a log event does not contain year information, ICAgent regards it as printed in the current year.</p> <p>Example:</p> <pre> YY - year (19) YYYY - year (2019) M - month (1) MM - month (01) D - day (1) DD - day (01) hh - hours (23) mm - minutes (59) ss - seconds (59) SSS - millisecond (999) hpm - hours (03PM) h:mmpm - hours:minutes (03:04PM) h:mm:sspm - hours:minutes:seconds (03:04:05PM) hh:mm:ss ZZZZ (16:05:06 +0100) hh:mm:ss ZZZ (16:05:06 CET) hh:mm:ss ZZ (16:05:06 +01:00) </pre>
Log Segmentation	<p>This parameter needs to be specified if the Log Format is set to Multi-line. By generation time indicates that a time wildcard is used to detect log boundaries, whereas By regular expression indicates that a regular expression is used.</p>
By regular expression	<p>You can set a regular expression to look for a specific pattern to indicate the beginning of a log event. This parameter needs to be specified when you select Multi-line for Log Format and By regular expression for Log Segmentation.</p>

Step 4: Configure Indexing

1. (Optional) Configure indexing. For details, see [Setting Indexes](#).
2. Click **Submit**.

Step 5: Complete the Ingestion Configuration

The created ingestion configuration will be displayed.

- Click its name to view its details.
- Click **Modify** in the **Operation** column to modify the ingestion configuration.
- Click **Configure Tag** in the **Operation** column to add a tag.
- Click **More > Copy** in the **Operation** column to copy the ingestion configuration.
- Click **More > Delete** in the **Operation** column to delete the ingestion configuration.

NOTE

Deleting an ingestion configuration may lead to log collection failures, potentially resulting in service exceptions related to user logs. In addition, the deleted ingestion configuration cannot be restored. Exercise caution when performing this operation.

2.1.18 Ingesting VPC Logs to LTS

LTS can collect logs from VPC.

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Configuring VPC Log Ingestion

Perform the following operations to configure VPC log ingestion:

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**.

Step 2 Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **VPC (Virtual Private Cloud)**.

Step 3 Select a log stream.

1. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
2. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
3. Click **Next: Configure VPC**.

Step 4 Click **Configure VPC**.

Step 5 Click **Next: Configure Log Stream**.

Table 2-22 Log stream parameters

Parameter	Description
Auto Structure and Index	If this function is enabled, the structuring for the log stream is based on the VPC system template, and the indexing enables quick analysis for all parsed VPC fields.

Step 6 Click **Submit**.

----End

2.1.19 Ingesting WAF Logs to LTS

LTS can collect logs from WAF.

For details, see .

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

2.2 Using APIs to Ingest Logs to LTS

2.2.1 Collecting Logs Using APIs

You can report logs to LTS with REST APIs. LTS supports APIs for reporting logs and high-precision logs.

The application scenarios and access IP addresses of the APIs are as follows:

Table 2-23 Scenarios

Name	Log Time	Example	Scenario
<p>Reporting Logs</p>	<p>When invoking the API to upload a batch of logs, you can specify an initial time with log_time_ns field.</p> <p>Time of each log can be calculated with log_time_ns+sequence count.</p>	<pre data-bbox="895 342 1184 618"> { "log_time_ns": "1586850540000000000", "contents": ["log1", "log2"], "labels": { "user_tag": "string" } } </pre> <p>When reported to LTS:</p> <p>The time of log1 is 158685054000000000.</p> <p>The time of log2 is 158685054000000001.</p>	<p>The logs are generated in sequence at similar time.</p>
<p>Reporting High-Precision Logs</p>	<p>When you invoke the API to upload a batch of logs, the log_time_ns field must be used to specify the log time for each log.</p>	<pre data-bbox="895 936 1184 1413"> { "contents":[{ "log_time_ns":"158685054000000000", "log":"log3" }, { "log_time_ns":"158685054000000008", "log":"log4" }], "labels":{" "user_tag":"string" } } </pre> <p>When reported to LTS:</p> <p>The time of log3 is 158685054000000000.</p> <p>The time of log4 is 158685054000000008.</p>	<p>The uploaded logs are generated out of order at different times. Each log needs to have its own timestamp.</p>

 **NOTE**

You can obtain the access IP address from the command displayed on the ICAgent installation page of the LTS console. For details, see [Installing ICAgent \(Intra-Region Hosts\)](#).

2.2.2 API for Reporting Logs

Function

This API is used to report tenant logs from a host to LTS.

The access IP address is contained in the ICAgent installation command displayed on the LTS console. The port number is 8102. You can check the [Example Request](#) to see how to add the access IP address and port number in a request.

URI

POST /v2/{project_id}/lts/groups/{log_group_id}/streams/{log_stream_id}/tenant/contents

Table 2-24 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see . No default value. Value length: 32 characters
log_group_id	Yes	String	Log group ID. For details about how to obtain it, see . No default value. Value length: 36 characters
log_stream_id	Yes	String	Log stream ID. For details about how to obtain it, see . No default value. Value length: 36 characters NOTE A write rate exceeding 100 MB/s per log stream may cause log losses.

Request Parameters

Table 2-25 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Indicates the user token obtained from IAM. No default value. Minimum length: 1,000 characters Maximum length: 2,000 characters
Content-Type	Yes	String	Set this parameter to application/json;charset=UTF-8 . No default value. Minimum length: 30 characters Maximum length: 30 characters

Table 2-26 Request body parameters

Parameter	Mandatory	Type	Description
log_time_ns	Yes	Long	Time when log data is reported (UTC time in nanoseconds). NOTE Logs reported to LTS through APIs are retained for two days (from the log reporting time to the current time). Logs reported more than two days ago will be deleted.
contents	Yes	Array of String	Indicates the log content.
labels	Yes	Object	Custom labels.
tenant_project_id	No	String	Project ID. For details about how to obtain it, see .

Response Parameters

When the status code is **200**, the response parameters are as follows:

Table 2-27 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> • SVCSTG.ALS.200.200
errorMessage	String	Indicates the response description. Example value: <ul style="list-style-type: none"> • Report success.
result	String	Response result.

When the status code is **400**, the response parameters are as follows:

Table 2-28 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> • SVCSTG.ALS.200.201 • SVCSTG.ALS.200.210
errorMessage	String	Indicates the error description. Example value: <ul style="list-style-type: none"> • Request conditions must be json format. • projectid xxx log's quota has full.
result	String	Response result.

When the status code is **401**, the response parameters are as follows:

Table 2-29 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> • SVCSTG.ALS.403.105
errorMessage	String	Indicates the error description. Example value: <ul style="list-style-type: none"> • Project id is invalid.
result	String	Response result.

When the status code is **500**, the response parameters are as follows:

Table 2-30 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> LTS.200500
errorMessage	String	Indicates the error description. Example value: <ul style="list-style-type: none"> Internal error
result	String	Response result.

When the status code is **503**, the response parameter is as follows:

Table 2-31 Response body parameter

Parameter	Type	Description
result	String	The requested service is unavailable.

Example Request

```
POST https://{access_IP_address:8102}/v2/{project_id}/lts/groups/{log_group_id}/streams/{log_stream_id}/tenant/contents
{
  "log_time_ns": 1586850540000000000,
  "contents": [
    "Fri Feb 1 07:48:04 UTC 2019 0\n",
    "Sat Apri 18 16:04:04 UTC 2019"
  ],
  "labels": {
    "user_tag": "string"
  }
}
```

Example Response

Example response with status code **200**:

Logs are reported.

```
{
  "errorCode": "SVCSTG.ALS.200.200",
  "errorMessage": "Report success.",
  "result": null
}
```

Example response with status code **401**:

The authentication information is incorrect or invalid.

```
{
  "errorCode" : "SVCSTG.ALS.403.105",
  "errorMessage" : "Project id is invalid.",
  "result": null
}
```

Status Code

Status Code	Description
200	The request has succeeded.
400	The request is invalid. Modify the request based on the description in error_msg before a retry.
401	The authentication information is incorrect or invalid.
500	An internal error occurred.
503	The requested service is unavailable.

2.2.3 API for Reporting High-Precision Logs

Function

This API is used to report tenant logs from a host to LTS.

The access IP address is contained in the ICAgent installation command displayed on the LTS console. The port number is 8102. You can check the [Example Request](#) to see how to add the access IP address and port number in a request.

NOTE

Each log event will carry a nanosecond-level timestamp when it is reported. When you view logs on the LTS console, the log events are sorted by timestamp.

URI

POST /v2/{project_id}/lts/groups/{log_group_id}/streams/{log_stream_id}/tenant/contents/high-accuracy

Table 2-32 URI parameters

Parameter	Man dator y	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see . No default value. Value length: 32 characters
log_group_id	Yes	String	Log group ID. For details about how to obtain it, see . No default value. Value length: 36 characters
log_stream_id	Yes	String	Log stream ID. For details about how to obtain it, see . No default value. Value length: 36 characters NOTE A write rate exceeding 100 MB/s per log stream may cause log losses.

Request Parameters

Table 2-33 Request header parameters

Parameter	Man dator y	Type	Description
X-Auth-Token	Yes	String	Indicates the user token obtained from IAM. No default value. Minimum length: 1,000 characters Maximum length: 2,000 characters
Content-Type	Yes	String	Set this parameter to application/json;charset=UTF-8 . No default value. Minimum length: 30 characters Maximum length: 30 characters

Parameter	Mandatory	Type	Description
Content-Encoding	No	String	Log compression format. Example value: <ul style="list-style-type: none"> • GZIP • SNAPPY • gzip • snappy

Table 2-34 Request body parameters

Parameter	Mandatory	Type	Description
contents	Yes	Array of LogContents	Indicates a list of log events that carry reporting timestamps.
labels	Yes	Object	Custom labels.
tenant_project_id	No	String	Project ID. For details about how to obtain it, see .

Table 2-35 LogContents

Parameter	Mandatory	Type	Description
log_time_ns	Yes	Long	Time when log data is reported (UTC time in nanoseconds). NOTE Logs reported to LTS through APIs are retained for two days (from the log reporting time to the current time). Logs reported more than two days ago will be deleted.
log	Yes	String	Indicates the log content.

Response Parameters

When the status code is **200**, the response parameters are as follows:

Table 2-36 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> • SVCSTG.ALS.200.200
errorMessage	String	Indicates the response description. Example value: <ul style="list-style-type: none"> • Report success.
result	String	Response result.

When the status code is **400**, the response parameters are as follows:

Table 2-37 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> • SVCSTG.ALS.200.201 • SVCSTG.ALS.200.210
errorMessage	String	Indicates the error description. Example value: <ul style="list-style-type: none"> • Request conditions must be json format. • projectid xxx log's quota has full.
result	String	Response result.

When the status code is **401**, the response parameters are as follows:

Table 2-38 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> • SVCSTG.ALS.403.105
errorMessage	String	Indicates the error description. Example value: <ul style="list-style-type: none"> • Project id is invalid.
result	String	Response result.

When the status code is **500**, the response parameters are as follows:

Table 2-39 Response body parameters

Parameter	Type	Description
errorCode	String	Indicates the error code. Example value: <ul style="list-style-type: none"> LTS.200500
errorMessage	String	Indicates the error description. Example value: <ul style="list-style-type: none"> Internal error
result	String	Response result.

When the status code is **503**, the response parameter is as follows:

Table 2-40 Response body parameter

Parameter	Type	Description
result	String	The requested service is unavailable.

Example Request

POST https://{access_IP_address:8102}/v2/{project_id}/lts/groups/{log_group_id}/streams/{log_stream_id}/tenant/contents/high-accuracy

```
{
  "contents": [
    {
      "log_time_ns": 1586850540000000000,
      "log": "Fri Feb 15 15:48:04 UTC 2019"
    },
    {
      "log_time_ns": 1586850540000000001,
      "log": "Sat Apr 18 16:04:04 UTC 2019"
    }
  ],
  "labels": {
    "user_tag": "string"
  }
}
```

Example Response

Example response with status code **200**:

Logs are reported.

```
{
  "errorCode": "SVCSTG.ALS.200.200",
```

```
"errorMessage": "Report success.",
"result": null
}
```

Example response with status code **401**:

The authentication information is incorrect or invalid.

```
{
"errorCode" : "SVCSTG.ALS.403.105",
"errorMessage" : "Project id is invalid.",
"result": null
}
```

Status Code

Status Code	Description
200	The request has succeeded.
400	The request is invalid. Modify the request based on the description in error_msg before a retry.
401	The authentication information is incorrect or invalid.
500	An internal error occurred.
503	The requested service is unavailable.

2.3 Other Ingestion Modes

2.3.1 Ingesting Logs to LTS Across IAM Accounts

If you choose **Cross-Account Ingestion - Log Stream Mapping** as the log ingestion type, you can create an agency to map the log stream of the delegator account to that of the delegated account. The delegated account is the current account used to log in to LTS.

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Prerequisites

An agency relationship has been created.

Constraints


Before data synchronization is complete, data in the target and source log streams may be different. Check back later in one hour.

Setting Cross-Account Ingestion

If you choose cross-account ingestion as the log ingestion type, perform the following operations to configure the ingestion:

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **Cross-Account Ingestion - Log Stream Mapping**.

Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > Cross-Account Ingestion - Log Stream Mapping**.

Alternatively, choose **Log Management** in the navigation pane and click the target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and **Ingest Log**. In the displayed dialog box, click **Cross-Account Ingestion - Log Stream Mapping**.

- Step 3** Select an agency.
Set parameters by referring to [Table 2-41](#) and click **Next: Log Stream Mapping**.

Table 2-41 Agency parameters

Parameter	Description
Agency Name	Enter the name of the agency created by the delegator. A delegator account can create an agency to delegate resource management permissions to another account.
Delegator Account Name	Enter the delegator account name to verify the delegation.

- Step 4** Map log streams.

On the **Log Stream Mapping** page, there are two ways to configure ingestion rules: automatic and manual configuration.

- **Automatic configuration**
 - a. Click **Auto Configure**.
 - b. On the displayed page, set the required parameters and click **OK**.

Table 2-42 Parameters of automatic ingestion rule configuration

Parameter	Description
Rule Name Prefix	Enter the rule name prefix. In automatic configuration, this prefix is used to generate multiple ingestion rules. Can contain only letters, digits, underscores (_), hyphens (-), and periods (.). The prefix cannot start with a period or underscore, or end with a period. If you do not specify a prefix, the default rule name prefix rule will be used.
Select the log groups or log streams that you want to ingest from the delegator account.	Up to 20 log groups or log streams can be selected.

 **NOTE**

By default, the names of the target log groups and target log streams of the delegated account are the same as those of the source log groups and source log streams of the delegator account. You can also manually change the names of the target log groups and target log streams.

c. Click **Preview**.

 **NOTE**

1. There are two types of preview results:
 - **A new target log stream will be created:** A target log group or log stream will be created in the delegated account.
 - **An existing target log stream will be ingested:** The target log group or log stream already exists in the delegated account.
2. Preview error messages are as follows:
 - Source log stream *xxx* has been configured as the target log stream.
 - Target log stream *xxx* has been configured as the source log stream.
 - Target log stream *xxx* already exists in another log group.
 - Target log stream *xxx* exists in different target log groups.
 - Duplicate rule names.
 - The source log stream *xxx* is already mapped.
 - The number of log groups has reached the upper limit. Select an existing log group.

If any of the preceding error messages is displayed, delete the corresponding ingestion rule of the log stream.

d. After the preview is complete, click **Submit**.

- **Manual configuration**

- a. On the **Log Stream Mapping** page, click **Add Rule**. Set the rule by referring to [Table 2-43](#).

Table 2-43 Parameters

Parameter		Description
Rule Name		The default value is rule_XXX . You can also specify a name as needed. Can contain only letters, digits, underscores (_), hyphens (-), and periods (.). The name cannot start with a period or underscore, or end with a period.
Delegat or Account	Source Log Group	Log group of the delegator account. Select an existing log group.
	Source Log Stream	Log stream of the delegator account. Select an existing log stream.
Delegat ed Account	Target Log Group	Log group of the delegator account. You can select an existing log group or enter a name to create one.
	Target Log Stream	Log stream of the delegated account. You can select an existing log stream or enter a name to create one.

- b. Click **Preview**.

 **NOTE**

1. There are two types of preview results:
 - **A new target log stream will be created:** A target log group or log stream will be created in the delegated account.
 - **An existing target log stream will be ingested:** The target log group or log stream already exists in the delegated account.
 2. There are five types of preview errors:
 - Source log stream *xxx* has been configured as the target log stream.
 - Target log stream *xxx* has been configured as the source log stream.
 - Target log stream *xxx* already exists in another log group.
 - Target log stream *xxx* exists in different target log groups.
 - Duplicate rule names.
 - The source log stream *xxx* is already mapped.
 - The number of log groups or log streams exceeds the upper limit.

If any of the preceding error messages is displayed, delete the corresponding ingestion rule of the log stream.
- c. After the preview is complete, click **Submit** and wait until the log ingestion task is created.

Step 5 Complete the ingestion configuration.

 **NOTE**

After the configuration is complete, data will be synchronized within one hour. Please check back later.

- If multiple log streams are ingested, you can click **Back to Ingestion Configurations** to view the log ingestion list.
- If a single log stream is ingested, click **Back to Ingestion Configurations** to view the log ingestion list. Click **View Log Stream** to view details about the ingested log stream.

----End

2.3.2 Ingesting Self-Built Kubernetes Application Logs to LTS

Kubernetes is an open-source container orchestration engine. It provides automatic deployment, large-scale scalability, and containerized application management. LTS can store and analyze application logs of services and nodes reported from Kubernetes clusters.

Perform the following steps to complete the ingestion configuration:

1. [Step 1: Select a Log Stream](#)
2. [Step 2: Check Dependencies](#)
3. [Step 3: Install the Log Collection Component](#)
4. [Step 4: \(Optional\) Select a Host Group](#)
5. [Step 5: Configure the Collection](#)
6. [Step 6: Configure Indexing](#)
7. [Step 7: Complete the Ingestion Configuration](#)

To collect logs from multiple scenarios, [set multiple ingestion configurations in a batch](#).

 **NOTE**

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.


Prerequisites

- Ensure that the Helm v3 installation command has been executed in the Kubernetes cluster.
- Ensure that kubectl has been configured for the Kubernetes cluster.

Step 1: Select a Log Stream

1. Log in to the management console and choose **Management & Deployment > Log Tank Service**.
2. Choose **Log Ingestion > Ingestion Center** in the navigation pane and click **Self-built K8s - Application Logs**.

Alternatively, choose **Log Ingestion > Ingestion Management** in the navigation pane, and click **Ingest Log > Self-built K8s - Application Logs**.


Alternatively, choose **Log Management** in the navigation pane and click the target log stream to access its details page. Click  in the upper right corner. On the displayed page, click the **Log Ingestion** tab and click **Ingest Log**. In the displayed dialog box, click **Self-built K8s - Application Logs**.

3. Choose a collection mode between **Fixed log stream** and **Custom log stream**.
 - If you set **Collect** to **Fixed log stream**, perform the following steps:

Logs will be collected to a fixed log stream. The default log streams for a kubernetes cluster are **stdout-*{ClusterID}*** for standard output/errors, **hostfile-*{ClusterID}*** for node files, **event-*{ClusterID}*** for Kubernetes events, and **containerfile-*{ClusterID}*** for container files. Log streams are automatically named with a cluster ID. For example, if the cluster ID is **Cluster01**, the standard output/error log stream is **stdout-Cluster01**.

Log streams that can be created for a kubernetes cluster are **stdout-*{ClusterID}*** for standard output/errors, **hostfile-*{ClusterID}*** for node files, **event-*{ClusterID}*** for Kubernetes events, and **containerfile-*{ClusterID}*** for container files. If one of them has been created in a log group, the log stream will no longer be created in the same log group or other log groups.

 - i. Select **Fixed log stream** for **Collect**.
 - ii. Enter the cluster name and ID.
 - iii. Select a log group.

 **NOTE**

If there is no such group, the system displays the following message: This log group does not exist and will be automatically created to start collecting logs.
 - iv. Click **Next: Check Dependencies**.
 - If you set **Collect** to **Custom log stream**, perform the following steps:
 - i. Select **Custom log stream**.
 - ii. Enter the cluster name and ID.
 - iii. Select a log group from the **Log Group** drop-down list. If there are no desired log groups, click **Create Log Group** to create one.
 - iv. Select a log stream from the **Log Stream** drop-down list. If there are no desired log streams, click **Create Log Stream** to create one.
4. Click **Next: Check Dependencies**.

Step 2: Check Dependencies

1. The system automatically checks whether the following are met:
 - There is a host group with the custom identifier **k8s-log-*ClusterID***.
 - There is a log group named **k8s-log-*ClusterID***. The log retention period and description of the log group can be modified. If **Fixed log stream** is selected, this item is checked.
 - The recommended log stream exists. The log retention period and description of the log stream can be modified. If **Fixed log stream** is selected, this item is checked.

You need to meet all the requirements before moving on. If not, click **Auto Correct**.

 **NOTE**

- **Auto Correct:** a one-click option to finish the previous settings.
 - **Check Again:** Recheck dependencies.
2. Click **Next: Install ICAgent**.

Step 3: Install the Log Collection Component

On any host in the Kubernetes cluster, perform the following steps:

1. Obtain the ICAgent installation package.
 - a. Obtain the ICAgent installation package (based on your actual information).

```
wget https://icagent-{regionId}.{obsDomainName}/ICAgent_linux/icagentK8s-5.5.1.2.tar.gz
```
 - b. Decompress the ICAgent installation package.

```
tar -xzf icagentK8s-5.5.1.2.tar.gz
```
 - c. Go to the directory.

```
cd icagentK8s
```
 - d. Generate installation commands:
Select the region of ingested logs.
Select the project ID of the ingesting account.
For **Kubernetes Cluster**, select **Intra-Region**.
2. Install ICAgent.
 - a. Copy the ICAgent installation command. You must replace the AK/SK with the obtained one. Manually replace it when copying the command, or replace it as prompted when running the command.
To prevent your AK/SK from being disclosed, select **Turn off command history to prevent the AK/SK from being stored** to disable historical record collection.
 - b. Use a remote login tool (such as PuTTY) to log in to the target host as the **root** user and run the copied command.
If the message **ICAgent install success** is displayed, the installation is successful. Then check the ICAgent status in the host list.
3. Click **ICAgent Already Installed**.

Step 4: (Optional) Select a Host Group

1. In the host group list, the host group to which the cluster belongs is selected by default. You can also select host groups as required.
2. Click **Next: Configurations**.

Step 5: Configure the Collection

When you configure log ingestion for self-built Kubernetes clusters, the collection configuration details are as follows.

1. **Collection Configuration Name:** Enter 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed. Do not start with a period or underscore, or end with a period.
2. **Data Source:** Select a data source type and configure it.

Table 2-44 Collection configuration parameters

Type	Description
Container standard output	<p>Collects stderr and stdout logs of a specified container in the cluster.</p> <p>Collects stderr and stdout logs of a specified container in the cluster. Either Container Standard Output (stdout) or Container Standard Error (stderr) must be enabled.</p> <ul style="list-style-type: none"> • If you enable Container Standard Error (stderr), select your collection destination path: Collect standard output and standard error to different files (stdout.log and stderr.log) or Collect standard output and standard error to the same file (stdout.log). • The standard output of the matched container is collected to the specified log stream. Standard output to AOM stops. • Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams.

Type	Description
Container file	<p>Collects file logs of a specified container path in a cluster.</p> <ul style="list-style-type: none"> Collection Paths: Add one or more host paths. LTS will collect logs from these paths. <p>NOTE If a container mount path has been configured for the CCE cluster workload, the paths added for this field are invalid. The collection paths take effect only after the mount path is deleted.</p> Add Custom Wrapping Rule: ICAgent determines whether a file is wrapped based on the file name rule. If your wrapping rule does not comply with the built-in rules, you can add a custom wrap rule to prevent log loss during repeated collection and wrapping. <p>The built-in rules are <i>{basename}{connector}{wrapping identifier}.{suffix}</i> and <i>{basename}.{suffix}{connector}{wrapping identifier}</i>. Connectors can be hyphens (-), periods (.), or underscores (_), wrapping identifiers can contain only non-letter characters, and the suffix can contain only letters.</p> <p>A custom wrapping rule consists of <i>{basename}</i> and the feature regular expression of the wrapped file. Example: If your log file name is test.out.log and the names after wrapping are test.2024-01-01.0.out.log and test.2024-01-01.1.out.log, configure the collection path to /opt/*.log, and add a custom wrapping rule: <i>{basename}\.\d{4}-\d{2}-\d{2}\.\d{1}.out.log</i>.</p> Allow Repeated File Collection (not available to Windows) <p>After you enable this function, one host log file can be collected to multiple log streams.</p> <p>After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams.</p> Set Collection Filters: Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.

Type	Description
Node file	<p>Collects files of a specified node path in a cluster.</p> <ul style="list-style-type: none"> ● Collection Paths: Add one or more host paths. LTS will collect logs from these paths. ● Add Custom Wrapping Rule: ICAgent determines whether a file is wrapped based on the file name rule. If your wrapping rule does not comply with the built-in rules, you can add a custom wrap rule to prevent log loss during repeated collection and wrapping. The built-in rules are <i>{basename}{connector}{wrapping identifier}.{suffix}</i> and <i>{basename}.{suffix}{connector}{wrapping identifier}</i>. Connectors can be hyphens (-), periods (.), or underscores (_), wrapping identifiers can contain only non-letter characters, and the suffix can contain only letters. A custom wrapping rule consists of <i>{basename}</i> and the feature regular expression of the wrapped file. Example: If your log file name is test.out.log and the names after wrapping are test.2024-01-01.0.out.log and test.2024-01-01.1.out.log, configure the collection path to /opt/*.log, and add a custom wrapping rule: <i>{basename}\.\d{4}-\d{2}-\d{2}\.\d{1}.out.log</i>. ● Allow Repeated File Collection (not available to Windows) After you enable this function, one host log file can be collected to multiple log streams. After you disable this function, each collection path must be unique. That is, the same log file in the same host cannot be collected to different log streams. ● Set Collection Filters: Blacklisted directories or files will not be collected. If you specify a directory, all files in the directory are filtered out.
Kubernetes event	<p>Collects event logs of the Kubernetes cluster. You do not need to configure this parameter. Only ICAgent 5.12.150 or later is supported.</p> <p>NOTE Kubernetes events of a Kubernetes cluster can be collected to only one log stream.</p>

3. Set kubernetes matching rules only when the data source type is set to **Container standard output** or **Container file**.

 **NOTE**

After entering a regular expression, click **Verify** to verify it.

Table 2-45 Kubernetes matching rules

Parameter	Description
Namespace Name Regular Expression	<p>Specifies the container whose logs are to be collected based on the namespace name. Regular expression matching is supported.</p> <p>NOTE LTS will collect logs of the namespaces with names matching this expression. To collect logs of all namespaces, leave this field empty.</p>
Pod Name Regular Expression	<p>Specifies the container whose logs are to be collected based on the pod name. Regular expression matching is supported.</p> <p>NOTE LTS will collect logs of the pods with names matching this expression. To collect logs of all pods, leave this field empty.</p>
Container Name Regular Expression	<p>Specifies the container whose logs are to be collected based on the container name (the Kubernetes container name is defined in spec.containers). Regular expression matching is supported.</p> <p>NOTE LTS will collect logs of the containers with names matching this expression. To collect logs of all containers, leave this field empty.</p>
Label Whitelist	<p>Specifies the containers whose logs are to be collected. If you want to set a Kubernetes label whitelist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple whitelists, you can select the And or Or relationship. This means a container will be matched when it satisfies all or any of the whitelists.</p> <p>NOTE If Label Value is empty, LTS will match all containers whose Kubernetes label contains a specified Label Key. If Label Value is not empty, only containers whose Kubernetes label contains a specified Label Key that is equal to its Label Value are matched. Label Key requires full matching while Label Value supports regular matching.</p>
Label Blacklist	<p>Specifies the containers whose logs are not to be collected. If you want to set a Kubernetes label blacklist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple blacklists, you can select the And or Or relationship. This means a container will be excluded when it satisfies all or any of the blacklists.</p> <p>NOTE If Label Value is empty, LTS will exclude all containers whose Kubernetes label contains a specified Label Key. If Label Value is not empty, only containers whose Kubernetes label contains a specified Label Key that is equal to its Label Value will be excluded. Label Key requires full matching while Label Value supports regular matching.</p>

Parameter	Description
Kubernetes Label	<p>After the Kubernetes Label is set, LTS adds related fields to logs.</p> <p>NOTE LTS adds the specified fields to the log when each Label Key has a corresponding Label Value. For example, if you enter "app" as the key and "app_alias" as the value, when the container label contains "app=lhs", "{app_alias: lhs}" will be added to the log.</p>
Container Label Whitelist	<p>Specifies the containers whose logs are to be collected. If you want to set a container label whitelist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple whitelists, you can select the And or Or relationship. This means a container will be matched when it satisfies all or any of the whitelists.</p> <p>NOTE If Label Value is empty, LTS will match all containers whose container label contains a specified Label Key. If Label Value is not empty, only containers whose container label contains a specified Label Key that is equal to its Label Value are matched. Label Key requires full matching while Label Value supports regular matching.</p>
Container Label Blacklist	<p>Specifies the containers whose logs are not to be collected. If you want to set a container label blacklist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple blacklists, you can select the And or Or relationship. This means a container will be excluded when it satisfies all or any of the blacklists.</p> <p>NOTE If Label Value is empty, LTS will exclude all containers whose container label contains a specified Label Key. If Label Value is not empty, only containers whose container label contains a specified Label Key that is equal to its Label Value will be excluded. Label Key requires full matching while Label Value supports regular matching.</p>
Container Label	<p>After the Container Label is set, LTS adds related fields to logs.</p> <p>NOTE LTS adds the specified fields to the log when each Label Key has a corresponding Label Value. For example, if you enter "app" as the key and "app_alias" as the value, when the container label contains "app=lhs", "{app_alias: lhs}" will be added to the log.</p>

Parameter	Description
Environment Variable Whitelist	<p>Specifies the containers whose logs are to be collected. If you want to set an environment variable whitelist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple whitelists, you can select the And or Or relationship. This means a container will be matched when it satisfies all or any of the whitelists.</p> <p>NOTE If Environment Variable Value is empty, LTS will match all containers whose environment variable contains a specified Environment Variable Key. If Environment Variable Value is not empty, only containers whose environment variable contains a specified Environment Variable Key that is equal to its Environment Variable Value are matched. Label Key requires full matching while Label Value supports regular matching.</p>
Environment Variable Blacklist	<p>Specifies the containers whose logs are not to be collected. If you want to set an environment variable blacklist, Label Key is mandatory and Label Value is optional.</p> <p>When adding multiple blacklists, you can select the And or Or relationship. This means a container will be excluded when it satisfies all or any of the blacklists.</p> <p>NOTE If Environment Variable Value is empty, LTS will exclude all containers whose environment variable contains a specified Environment Variable Key. If Environment Variable Value is not empty, only containers whose environment variable contains a specified Environment Variable Key that is equal to its Environment Variable Value will be excluded. Label Key requires full matching while Label Value supports regular matching.</p>
Environment Variable Label	<p>After the environment variable label is set, the log service adds related fields to the log.</p> <p>NOTE LTS adds the specified fields to the log when each Environment Variable Key has a corresponding Environment Variable Value. For example, if you enter "app" as the key and "app_alias" as the value, when the Kubernetes environment variable contains "app=lts", "{app_alias: lts}" will be added to the log.</p>

4. Enable structuring parsing. For details, see [Setting ICAgent Structuring Parsing Rules](#).

LTS enables combined parsing, allowing you to create different structuring parsing rules for each collection configuration of a log stream.

If you have configured cloud structuring parsing, delete its configurations before configuring ICAgent structuring parsing.

5. Set other configurations.

Table 2-46 Other configurations

Parameter	Description
Max Directory Depth	<p>The maximum directory depth is 20 levels.</p> <p>Collection paths can use double asterisks (**) for multi-layer fuzzy match. Specify the maximum directory depth in the text box. For example, if your log path is /var/logs/department/app/a.log and your collection path is /var/logs/**/a.log, logs will not be collected when this parameter is set to 1, but will be collected when this parameter is set to 2 or a larger number.</p>
Split Logs	<ul style="list-style-type: none"> • If log splitting is enabled, logs exceeding the specified size will be split into multiple logs for collection. Specify the size in the range from 500 KB to 1,024 KB. For example, if you set the size to 500 KB, a 600 KB log will be split into a 500 KB log and a 100 KB log. This restriction is applicable to single-line logs only, not multi-line logs. • If log splitting is disabled, when a log exceeds 500 KB, the extra part will be truncated and discarded.
Collect Binary Files	<p>LTS can collect binary files.</p> <p>Run the file -i File_name command to view the file type. charset=binary indicates that a log file is a binary file.</p> <p>If this option is enabled, binary log files will be collected, but only UTF-8 strings are supported. Other strings will be garbled on the LTS console.</p> <p>If this option is disabled, binary log files will not be collected.</p>
Log File Code	<p>The log file encoding format can be UTF-8 or GBK (not available to Windows).</p> <p>UTF-8 encoding is a variable-length encoding mode and represents Unicode character sets. GBK, an acronym for Chinese Internal Code Extension Specification, is a Chinese character encoding standard that extends both the ASCII and GB2312 encoding systems.</p>
Collection Policy	<p>Select incremental or full collection.</p> <ul style="list-style-type: none"> • Incremental: When collecting a new file, ICAgent reads the file from the end of the file. • All: When collecting a new file, ICAgent reads the file from the beginning of the file.

Parameter	Description
Custom Metadata	<ul style="list-style-type: none"> If this option is disabled, ICAgent will report logs to LTS based on the default system fields. You do not need to and cannot configure the fields. If this option is enabled, ICAgent will report logs based on your selected built-in fields and fields created with custom key-value pairs. Built-in Fields: Select built-in fields as required. Custom Key-Value Pairs: Click Add and set a key and value.

6. Configure the log format and time by referring to [Table 2-47](#).

Table 2-47 Log collection settings

Parameter	Description
Log Format	<ul style="list-style-type: none"> Single-line: Each log line is displayed as a single log event. Multi-line: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.
Log Time	<p>System time: log collection time by default. It is displayed at the beginning of each log event.</p> <p>NOTE</p> <ul style="list-style-type: none"> Log collection time is the time when logs are collected and sent by ICAgent to LTS. Log printing time is the time when logs are printed. ICAgent collects and sends logs to LTS with an interval of 1 second. Restriction on log collection time: Logs are collected within 24 hours before and after the system time.

Parameter	Description
	<p>Time wildcard: You can set a time wildcard so that ICAgent will look for the log printing time as the beginning of a log event.</p> <ul style="list-style-type: none"> • If the time format in a log event is 2019-01-01 23:59:59.011, the time wildcard should be set to YYYY-MM-DD hh:mm:ss.SSS. • If the time format in a log event is 19-1-1 23:59:59.011, the time wildcard should be set to YY-M-D hh:mm:ss.SSS. <p>NOTE If a log event does not contain year information, ICAgent regards it as printed in the current year.</p> <p>Example:</p> <pre> YY - year (19) YYYY - year (2019) M - month (1) MM - month (01) D - day (1) DD - day (01) hh - hours (23) mm - minutes (59) ss - seconds (59) SSS - millisecond (999) hpm - hours (03PM) h:mmpm - hours:minutes (03:04PM) h:mm:sspm - hours:minutes:seconds (03:04:05PM) hh:mm:ss ZZZZ (16:05:06 +0100) hh:mm:ss ZZZ (16:05:06 CET) hh:mm:ss ZZ (16:05:06 +01:00) </pre>
Log Segmentation	<p>This parameter needs to be specified if the Log Format is set to Multi-line. By generation time indicates that a time wildcard is used to detect log boundaries, whereas By regular expression indicates that a regular expression is used.</p>
By regular expression	<p>You can set a regular expression to look for a specific pattern to indicate the beginning of a log event. This parameter needs to be specified when you select Multi-line for Log Format and By regular expression for Log Segmentation.</p>

 **NOTE**

The time wildcard and regular expression will look for the specified pattern right from the beginning of each log line. If no match is found, the system time, which may be different from the time in the log event, is used. In general cases, you are advised to select **Single-line** for **Log Format** and **System time** for **Log Time**.

Step 6: Configure Indexing

1. (Optional) Configure indexing. For details, see [Setting Indexes](#).

2. Click **Submit**.

Step 7: Complete the Ingestion Configuration

The created ingestion configuration will be displayed.

- Click its name to view its details.
- Click **Modify** in the **Operation** column to modify the ingestion configuration.
- Click **Configure Tag** in the **Operation** column to add a tag.
- Click **More > Copy** in the **Operation** column to copy the ingestion configuration.
- Click **More > Delete** in the **Operation** column to delete the ingestion configuration.

NOTE

Deleting an ingestion configuration may lead to log collection failures, potentially resulting in service exceptions related to user logs. In addition, the deleted ingestion configuration cannot be restored. Exercise caution when performing this operation.

Setting Multiple Ingestion Configurations in a Batch

You can set multiple ingestion configurations for multiple scenarios in a batch, avoiding repetitive setups.

- Step 1** On the **Ingestion Management** page, click **Batch Ingest** to go to the details page. For details, see [Table 2-48](#).

Table 2-48 Adding configurations in batches

Type	Parameter	Description
Basic Settings	Ingestion Type	Select Self-built K8s - Application Logs .
	Configurations to Add	Enter the number of ingestion configurations in the text box and click Add . A maximum of 100 ingestion configurations can be added, including the one already exists under Ingestion Settings by default. Therefore, you can add up to 99 more.
Ingestion Settings	Configuration List	<ol style="list-style-type: none"> 1. The ingestion configurations are displayed on the left. You can add up to 99 more configurations. 2. The ingestion configuration details are displayed on the right. Set them by referring to Step 5: Configure the Collection. 3. After an ingestion configuration is complete, you can click Apply to Other Configurations to copy its settings to other configurations.

- Step 2** Click **Check Parameters**. After the check is successful, click **Submit**.

The added ingestion configurations will be displayed on the **Ingestion Management** page after the batch creation is successful.

Step 3 (Optional) Perform the following operations on ingestion configurations:

- Select multiple existing ingestion configurations and click **Edit**. On the displayed page, select an ingestion type to modify the corresponding ingestion configurations.
- Select multiple existing ingestion configurations and click **Enable** or **Disable**. If you toggle off the switch in the **Status** column of an ingestion configuration, logs will not be collected for this configuration.
- Select multiple existing ingestion configurations and click **Delete**.

----End

2.4 Setting ICAgent Structuring Parsing Rules

You can use ICAgent to collect logs to LTS. When creating a log ingestion task, you can customize collection policies, such as parsing, whitelist, and blacklist rules, and raw log uploading settings. An ICAgent collection configuration defines the process of collecting logs of the same type from a server, parsing them, and sending them to a specified log stream.

The ICAgent structuring function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Advantages

- Collecting log data in non-intrusive mode as log files: You do not need to modify application code, and log collection does not affect application running.
- Handling various exceptions during log collection: Security measures such as proactive retry and local cache are taken when a network or server exception occurs.
- Centralized management: After installing ICAgent, you only need to set configurations such as host groups and ICAgent collection on the LTS console.
- Comprehensive self-protection mechanisms: ICAgent is designed with strict limitations and protective measures regarding CPU, memory, and network usage, to minimize its impact on the performance of other services sharing the same server.

Before configuring log ingestion, get to know the structuring parsing rules of ICAgent collection to facilitate your operations. LTS enables combined parsing, allowing you to create different structuring parsing rules for each collection configuration of a log stream.

LTS supports the following log structuring parsing rules:

- **Single-Line - Full-Text Log**: Each log line is displayed as a single log event.
- **Multi-Line - Full-Text Log**: Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.

- **JSON**: splits JSON logs into key-value pairs.
- **Delimiter**: applicable to logs separated by fixed symbols (such as spaces, commas, and colons).
- **Single-Line - Completely Regular**: uses a regular expression to extract fields from single-line logs in any format. After entering a regular expression, click **Verify** to verify it.
- **Multi-Line - Completely Regular**: uses a regular expression to extract fields from multi-line logs in any format. The regular expression of the first line can be automatically generated or manually entered. After entering a regular expression, click **Verify** to verify it.
- **Combined Parsing**: applicable to logs in multiple nested formats, for example, JSON logs with delimiters. For logs with complex structures and requiring multiple parsing modes, you can use this mode. It allows you to input code of JSON format on the console to define the pipeline logic for log parsing. You can add one or more plug-in processing configurations. ICAgent executes the configurations one by one based on the specified sequence.

NOTE

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

- **From now**: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- **From last**: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified**: queries log data that is generated in a specified time range.



Single-Line - Full-Text Log

If you want to display each line of log data as a single log on the LTS page, select **Single-Line**.

1. Select **Single-Line - Full-Text Log**.
2. Enable **Log Filtering** (disabled by default) as required and add a maximum of 20 whitelist or blacklist rules.
 - Add a whitelist rule (available only when **Log Filtering** is enabled).


You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a matching criterion, collecting and reporting only logs that match the specified regular expression.


 - i. Click **Add** and enter a key and filtering rule (regular expression). In single-line and multi-line full-text modes, **content** is used as the key name *{key}* of the full text by default. The relationship between multiple filtering rules is **OR**. For example, to collect log events containing **hello** from the log source file, set the filtering rule to **.*hello.***.

- ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.
- Add a blacklist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a discarding criterion, discarding logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). In single-line and multi-line full-text modes, **content** is used as the key name *{key}* of the full text by default. The relationship between multiple filtering rules is **OR**. For example, if you do not want to collect log events containing **test** from the log source file, set the filtering rule to **.*test.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.

Multi-Line - Full-Text Log



Multiple lines of exception log events can be displayed as a single log event. This is helpful when you check logs to locate problems.

1. Select **Multi-Line - Full-Text Log**.
2. Select a log example from existing logs or paste it from the clipboard.
 - Click **Select from Existing Logs**, select a log event, and click **OK**. You can select different time ranges to filter logs.
 - Click **Paste from Clipboard** to paste the copied log content to the **Log Example** box.
3. A regular expression can be automatically generated or manually entered under **Regular Expression of the First Line**. The regular expression of the first line must match the entire first line, not just the beginning of the first line.
4. Enable **Log Filtering** (disabled by default) as required and add a maximum of 20 whitelist or blacklist rules.
 - Add a whitelist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a matching criterion, collecting and reporting only logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). In single-line and multi-line full-text modes, **content** is used as the key name *{key}* of the full text by default. The relationship between multiple filtering rules is **OR**. For example, to collect log events containing **hello** from the log source file, set the filtering rule to **.*hello.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.

- Add a blacklist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a discarding criterion, discarding logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). In single-line and multi-line full-text modes, **content** is used as the key name *{key}* of the full text by default. The relationship between multiple filtering rules is **OR**. For example, if you do not want to collect log events containing **test** from the log source file, set the filtering rule to **.*test.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.

JSON

This option is applicable to JSON logs and splits them into key-value pairs.

1. Choose **JSON**.
2. Enable **Log Filtering** (disabled by default) as required and add a maximum of 20 whitelist or blacklist rules.
 - Add a whitelist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a matching criterion, collecting and reporting only logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, to collect log events containing **hello** from the log source file, set the filtering rule to **.*hello.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.
 - Add a blacklist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a discarding criterion, discarding logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, if you do not want to collect log events containing **test** from the log source file, set the filtering rule to **.*test.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.
3. **Raw Log Upload:**

After this function is enabled, raw logs are uploaded to LTS as the value of the **content** field.

4. **Upload Parsing Failure Log:**

After this function is enabled, raw logs are uploaded to LTS as the value of the **_content_parse_fail_** field.

5. **Custom Time:**

Enabling this lets you specify a field as the log time. Otherwise, the time set during ingestion configuration is used.

6. **JSON Parsing Layers:** Add 1 to 4 JSON parsing layers. The value must be an integer and is **1** by default.


This function expands the fields of a JSON log. For example, for raw log `{"key1":{"key2":"value"}}`, if you choose to parse it into 1 layer, the log will become `{"key1":{"key2":"value"}}`; if you choose to parse it into 2 layers, the log will become `{"key1.key2":"value"}`.

Delimiter


Logs can be parsed by delimiters, such as commas (,), spaces, or other special characters.

1. Select a delimiter.
2. Select or customize a delimiter.
3. Select a log example from existing logs or paste it from the clipboard, click **Verify**, and view the results under **Extraction Results**.
 - Click **Select from Existing Logs**, select a log event, and click **OK**. You can select different time ranges to filter logs.
 - Click **Paste from Clipboard** to paste the copied log content to the **Log Example** box.
4. Enable **Log Filtering** (disabled by default) as required and add a maximum of 20 whitelist or blacklist rules.
 - Add a whitelist rule (available only when **Log Filtering** is enabled).

You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a matching criterion, collecting and reporting only logs that match the specified regular expression.


 - i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, to collect log events containing **hello** from the log source file, set the filtering rule to **.*hello.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.
 - Add a blacklist rule (available only when **Log Filtering** is enabled).


You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a discarding criterion, discarding logs that match the specified regular expression.

- i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, if you do not want to collect log events containing **test** from the log source file, set the filtering rule to **.*test.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.
5. **Raw Log Upload:**
After this function is enabled, raw logs are uploaded to LTS as the value of the **content** field.
6. **Upload Parsing Failure Log:**
After this function is enabled, raw logs are uploaded to LTS as the value of the **_content_parse_fail_** field.
7. **Custom Time:**
Enabling this lets you specify a field as the log time. Otherwise, the time set during ingestion configuration is used.

Single-Line - Completely Regular

This option is applicable to single-line logs in any format and uses a regular expression to extract fields.

1. Select **Single-Line - Completely Regular**.
2. Select a log example from existing logs or paste it from the clipboard.
 - Click **Select from Existing Logs**, select a log event, and click **OK**. You can select different time ranges to filter logs.
 - Click **Paste from Clipboard** to paste the copied log content to the **Log Example** box.
3. Enter a regular expression for extracting the log under **Extraction Regular Expression**, click **Verify**, and view the results under **Extraction Results**.
Alternatively, click **automatic generation of regular expressions**. In the displayed dialog box, extract fields based on the log example, enter the key, and click **OK** to automatically generate a regular expression. Then, click **OK**.
4. Enable **Log Filtering** (disabled by default) as required and add a maximum of 20 whitelist or blacklist rules.
 - Add a whitelist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a matching criterion, collecting and reporting only logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, to collect log events containing **hello** from the log source file, set the filtering rule to **.*hello.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.

- Add a blacklist rule (available only when **Log Filtering** is enabled).
You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a discarding criterion, discarding logs that match the specified regular expression.
 - i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, if you do not want to collect log events containing **test** from the log source file, set the filtering rule to **.*test.***.
 - ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.
5. **Raw Log Upload:**
After this function is enabled, raw logs are uploaded to LTS as the value of the **content** field.
 6. **Upload Parsing Failure Log:**
After this function is enabled, raw logs are uploaded to LTS as the value of the **_content_parse_fail_** field.
 7. **Custom Time:**
Enabling this lets you specify a field as the log time. Otherwise, the time set during ingestion configuration is used.

Multi-Line - Completely Regular

This option is applicable to multi-line logs in any format and uses a regular expression to extract fields.

1. Select **Multi-Line - Completely Regular**.
2. Select a log example from existing logs or paste it from the clipboard.
 - Click **Select from Existing Logs**, select a log event, and click **OK**. You can select different time ranges to filter logs.
 - Click **Paste from Clipboard** to paste the copied log content to the **Log Example** box.
3. A regular expression can be automatically generated or manually entered under **Regular Expression of the First Line**. The regular expression of the first line must match the entire first line, not just the beginning of the first line.

 **NOTE**

The regular expression of the first line is used to identify the beginning of a multi-line log. Example:

```
2024-10-11 10:59:07.000 a.log:1 level:warn
no.1 log
2024-10-11 10:59:17.000 a.log:2 level:warn
no.2 log
```

Complete lines:

```
2024-10-11 10:59:07.000 a.log:1 level:warn
no.1 log
```

First line:

```
2024-10-11 10:59:07.000 a.log:1 level:warn
```

Example of the regular expression of the first line: $^(\d+-\d+-\d+)\s+\d+:\d+:\d+.\d+)$. The date in each first line is unique. Therefore, the regular expression in the first line can be generated based on the date.

4. Enter a regular expression for extracting the log under **Extraction Regular Expression**, click **Verify**, and view the results under **Extraction Results**.

Alternatively, click **automatic generation of regular expressions**. In the displayed dialog box, extract fields based on the log example, enter the key, and click **OK** to automatically generate a regular expression. Then, click **OK**.

 **NOTE**

The extraction result is the execution result of the extraction regular expression instead of the first line regular expression. To check the execution result of the first line regular expression, go to the target log stream.


If you enter an incorrect regular expression for **Regular Expression of the First Line**, you cannot view the reported log stream data.

5. Enable **Log Filtering** (disabled by default) as required and add a maximum of 20 whitelist or blacklist rules.

- Add a whitelist rule (available only when **Log Filtering** is enabled).

You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a matching criterion, collecting and reporting only logs that match the specified regular expression.

- i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, to collect log events containing **hello** from the log source file, set the filtering rule to **.*hello.***.

- ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.

- Add a blacklist rule (available only when **Log Filtering** is enabled).

You can add filtering rules to retain valuable log data by applying regular expressions to the values of specified keys. A filtering rule acts as a discarding criterion, discarding logs that match the specified regular expression.

- i. Click **Add** and enter a key and filtering rule (regular expression). A key is a log field name. The relationship between multiple filtering rules is **OR**. For example, if you do not want to collect log events

containing **test** from the log source file, set the filtering rule to **.*test.***.

- ii. Click  in the **Operation** column, enter a field value, and click **Verify** to verify the rule.

6. **Raw Log Upload:**

After this function is enabled, raw logs are uploaded to LTS as the value of the **content** field.

7. **Upload Parsing Failure Log:**

After this function is enabled, raw logs are uploaded to LTS as the value of the **_content_parse_fail_** field.

8. **Custom Time:**

Enabling this lets you specify a field as the log time. Otherwise, the time set during ingestion configuration is used.

Combined Parsing

This option is applicable to logs in multiple nested formats, for example, JSON logs with delimiters. You can customize parsing rules based on the syntax.

1. Select **Combined Parsing**.
2. Select a log example from existing logs or paste it from the clipboard and enter the configuration content under **Plug-in Settings**.
3. Customize the settings based on the log content by referring to the following plug-in syntaxes.
 - processor_regex

Table 2-49 Regular expression extraction

Parameter	Type	Description
source_key	string	Original field name.
regex	string	() in a regular expression indicates the field to be extracted.
keys	string	Field name for the extracted content.
keep_source	boolean	Whether to retain the original field.
keep_source_if_parsing_error	boolean	Whether to retain the original field when a parsing error occurs.

- processor_split_string

Table 2-50 Parsing using delimiters

Parameter	Type	Description
source_key	string	Original field name.
split_sep	string	Delimiter string.
keys	string	Field name for the extracted content.
keep_source	boolean	Whether to retain the original field in the parsed log.
split_type	char/special_char/string	Delimiter type. The options are char (single character), special_char (invisible character), and string .
keep_source_if_parse_error	boolean	Whether to retain the original field when a parsing error occurs.

- processor_split_key_value

Table 2-51 Key-value pair segmentation

Parameter	Type	Description
source_key	string	Original field name.
split_sep	string	Delimiter between key-value pairs. The default value is the tab character (\t).
expand_connector	string	Delimiter between the key and value in a key-value pair. The default value is a colon (:).
keep_source	boolean	Whether to retain the original field in the parsed log.

- processor_add_fields

Table 2-52 Adding a field

Parameter	Type	Description
fields	json/object	Name and value of the field to be added. The field is in key-value pair format. Multiple key-value pairs can be added.

- processor_drop

Table 2-53 Discarded fields

Parameter	Type	Description
drop_keys	string	List of discarded fields.

- processor_rename

Table 2-54 Renaming a field

Parameter	Type	Description
source_keys	string	Original name.
destkeys	string	New name.

- processor_json

Table 2-55 JSON expansion and extraction

Parameter	Type	Description
source_key	string	Original field name.
keep_source	string	Whether to retain the original field in the parsed log.
expand_depth	int	JSON expansion depth. The default value 0 indicates that the depth is not limited. Other numbers, such as 1 , indicate the current level.
expand_connector	string	Connector for expanding JSON. The default value is an underscore (_).
prefix	string	Prefix added to a field name when JSON is expanded.
keep_source_if_parse_error	boolean	Whether to retain the original field when a parsing error occurs.

- processor_filter_regex

Table 2-56 Filters

Parameter	Type	Description
include	json/object	The key indicates the log field, and the value indicates the regular expression to be matched.

Parameter	Type	Description
exclude	json/object	The key indicates the log field, and the value indicates the regular expression to be matched.

- processor_gotime

Table 2-57 Extraction time

Parameter	Type	Description
source_key	string	Original field name.
source_format	string	Original time format.
source_location	int	Original time zone. If the value is empty, it indicates the time zone of the host or container where the logtail is located.
dest_key	string	Target field after parsing.
dest_format	string	Time format after parsing.
dest_location	int	Time zone after parsing. If this parameter is left blank, the time zone of the local host is used.
set_time	boolean	Whether to set the parsed time as the log time.
keep_source	boolean	Whether to retain the original field in the parsed log.

4. Example:

```
[
  {
    "type": "processor_regex",
    "detail": {
      "source_key": "content",
      "regex": "*",
      "keys": [
        "key1",
        "key2"
      ],
      "multi_line_regex": "*",
      "keep_source": true,
      "keep_source_if_parse_error": true
    }
  },
  {
    "type": "processor_split_string",
    "detail": {
      "split_sep": ".",
      "split_type": ".",
      "split_keys": [
        "key1",
        "key2"
      ]
    }
  }
]
```



```

    ],
    "source_key": "context",
    "keep_source": true,
    "keep_source_if_parse_error": true
  }
},
{
  "type": "processor_add_fields",
  "detail": {
    "fields": [
      {
        "key1": "value1"
      },
      {
        "key2": "value2"
      }
    ]
  }
},
{
  "type": "processor_drop",
  "detail": {
    "drop_keys": [
      "key1",
      "key2"
    ]
  }
},
{
  "type": "processor_rename",
  "detail": {
    "source_key": [
      "skey1",
      "skey2"
    ],
    "dest_keys": [
      "dkey1",
      "dkey2"
    ]
  }
},
{
  "type": "processor_json",
  "detail": {
    "source_key": "context",
    "expand_depth": 4,
    "expand_connector": "_",
    "prefix": "prefix",
    "keep_source": true,
    "keep_source_if_parse_error": true
  }
},
{
  "type": "processor_gotime",
  "detail": {
    "source_key": "skey",
    "source_format": "ydm",
    "source_location": 8,
    "dest_key": "dkey",
    "dest_format": "ydm",
    "dest_location": 8,
    "set_time": true,
    "keep_source": true,
    "keep_source_if_parse_error": true
  }
},
{
  "type": "processor_filter_regex",
  "detail": {

```

```

    "include": {
      "ikey1": "*",
      "ikey2": "*"
    },
    "exclude": {
      "ekey1": "*"
    }
  }
}
]

```


Custom Time

Enable **Custom Time** and set parameters by referring to [Table 2-58](#).

 **NOTE**

- If the time format is incorrect or the specified field does not exist, the log time is the time set during ingestion configuration.
- The time field needs to be verified again when operations such as field name modification, field deletion, and field type modification are performed on structuring parsing.

Table 2-58 Parameter configuration

Parameter	Description	Example
Key Name of the Time Field	Name of an extracted field. You can select an extracted field from the drop-down list. The field is of the string or long type.	test
Field Value	Value of an extracted field. After a Key is selected, the Value is automatically filled in. NOTE The value of the field must be within 24 hours earlier or later than the current time.	2023-07-19 12:12:00
Time Format	For details, see Common Log Time Formats .	yyyy-MM-dd HH:mm:ss
Operation	Click the verification icon (). If message The time format is successfully matched with the time field value. is displayed, the verification is successful.	-

Common Log Time Formats

[Table 2-59](#) lists common log time formats.

 **NOTE**

By default, log timestamps in LTS are accurate to seconds. You do not need to configure information such as milliseconds and microseconds.

Table 2-59 Time formats

Format	Description	Example
EEE	Abbreviation for Week.	Fri
EEEE	Full name for Week.	Friday
MMM	Abbreviation for Month.	Jan
MMMM	Full name for Month.	January
dd	Number of the day in a month, ranging from 01 to 31 (decimal).	07, 31
HH	Hour, in 24-hour format.	22
hh	Hour, in 12-hour format.	11
MM	Number of the month, ranging from 01 to 12 (decimal).	08
mm	Number of the minute, ranging from 00 to 59 (decimal).	59
a	AM or PM	AM, PM
hh:mm:ss a	Time in the 12-hour format.	11:59:59 AM
HH:mm	Hour and minute format.	23:59
ss	Number of the second, ranging from 00 to 59 (decimal).	59
yy	Year without century, ranging from 00 to 99 (decimal).	04, 98
yyyy	Year (decimal).	2004, 1998
d	Number of the day in a month, ranging from 1 to 31 (decimal).	7, 31
DDD	Number of the day in a year, ranging from 001 to 366 (decimal).	365
u	Number of the day in a week, ranging from 1 to 7 (decimal). The value 1 indicates Monday.	2
w	Number of the week in a year. Sunday is the start of a week. The value ranges from 00 to 53.	23

Format	Description	Example
W	Number of the week in a month, ranging from 0 to 5.	2
U	Number of the day in a week, ranging from 0 to 6 (decimal). The value 0 indicates Sunday.	5
EEE MMM dd HH:mm:ss yyyy	Standard date and time.	Tue Nov 20 14:12:58 2020
EEE MMM dd yyyy	Standard date without time.	Tue Nov 20 2020
HH:mm:ss	Standard time without date.	11:59:59
%s	UNIX Timestamp.	147618725

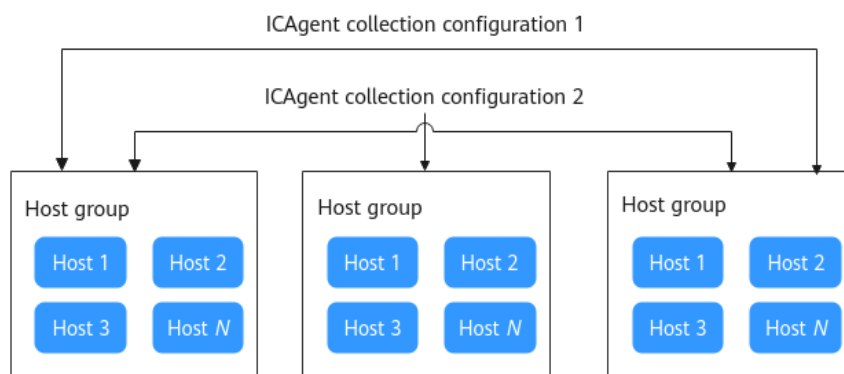
3 Host Management

3.1 Managing Host Groups

Host groups allow you to configure host log ingestion efficiently. You can add multiple hosts to a host group and associate the host group with log ingestion configurations. The ingestion configurations will then be applied to all the hosts in the host group. For details, see [Figure 3-1](#).

- When there is a new host, simply add it to a host group and the host will automatically inherit the log ingestion configurations associated with the host group.
- You can also use host groups to modify the log collection paths for multiple hosts at one go.

Figure 3-1 Host groups



You can create host groups of the IP address and custom identifier types.

- **Creating a Host Group (IP Address):** Select hosts of the IP address type and add them to the host group.
- **Creating a Host Group (Custom Identifier):** You need to create identifiers for each host group and host. Hosts with an identifier will automatically be included in the corresponding host group sharing that identifier.

NOTE

Host groups with custom identifiers are suitable for the following scenarios:

- In custom network environments like VPCs, potential IP address conflicts among hosts may impede ICAgent management in LTS. Using custom identifiers can resolve this issue.
- Multiple servers using the same custom identifier enable auto scaling of host groups. Simply assign a custom identifier for a new host; LTS will then automatically identify the host and add it to corresponding host group with the same identifier.

Creating a Host Group (IP Address)

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Host Management > Host Groups** in the navigation pane.
- Step 3** Click **Create Host Group** in the upper right corner.
- Step 4** In the displayed slide-out panel, enter a host group name, select **IP** for **Host Group Type**, and select a host OS (**Linux** or **Windows**).

Figure 3-2 Creating an IP address host group

The screenshot shows the 'Create Host Group' dialog box. It has the following fields and controls:

- Host Group:** A text input field.
- Host Group Type:** A dropdown menu with 'IP' selected and 'Custom identifier' as an alternative.
- Host Type:** A dropdown menu with 'Linux' selected and 'Windows' as an alternative.
- Remark:** A text area with a character count of 0/1024.
- Add Host:** A section containing:
 - Buttons: 'Install ICAgent', 'Uninstall ICAgent', 'Search by Host IP Address' (dropdown), 'View Selected (0)'.
 - Search box: 'Click here to choose a filter condition'.
 - Filter tabs: 'Host Name', 'Host IPv4 ...', 'Host IPv6 ...', 'Enterpris...', 'ICAgen...', 'ICAgen...', 'Upda...'.
 - Message: 'No hosts available. If your hosts are not displayed here, install ICAgent on the hosts.' with an 'Install ICAgent' button.

- Step 5** In the host list, select one or more hosts to add to the group and click **OK**.
 - You can filter hosts by host name or host IP address. You can also click **Search by Host IP Address** and enter multiple host IP addresses in the displayed search box to search for matches.
 - If your desired hosts are not in the list, click **Install ICAgent**. On the displayed page, install ICAgent on the hosts as prompted. For details, see [Installing ICAgent \(Intra-Region Hosts\)](#).

----End


Creating a Host Group (Custom Identifier)

To create a host group of the custom identifier type, you need to plan the hosts to be identified in advance.

Step 1 Click **Create Host Group** in the upper right corner.

Step 2 In the displayed slide-out panel, enter a host group name, select **Custom Identifier** for **Host Group Type**, and select a host OS (**Linux** or **Windows**).

Figure 3-3 Creating a custom identifier host group

Step 3 Enter a custom identifier. You can also click  **Add** to enter more.

NOTE

Up to 10 custom identifiers can be added.

Step 4 Click **OK**. After the host group is created, go to **5** to add hosts to it.

Step 5 Perform the following operations to create the **custom_tag** file to save host tags:

1. Log in to the host and run the **cd /opt/cloud** command. If the system indicates that the **/opt/cloud** directory does not exist, run the **mkdir /opt/cloud/** command to create it. If the **/opt/cloud** directory already exists, navigate to it and run the **mkdir lts** command to create the **lts** directory in it.
2. Run the **chmod 750 lts** command to modify the permission on the **lts** directory.
3. Run the **touch custom_tag** command in the **lts** directory to create the **custom_tag** file.
4. Run the **chmod 640 custom_tag;vi custom_tag** command to modify the **custom_tag** permission and open the file.
5. Press **i** to enter the insert mode, enter a custom identifier, press **Esc**, enter **:wq!**, save the modification and exit.
6. Use either of the following methods to add a host to the custom identifier host group:

Table 3-1 Methods


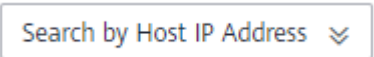
Type	Method 1 (Recommended)	Method 2
Linux host	View the host's identifier in the custom_tag file of the /opt/cloud/lts directory on the host. Then, add the identifier to the host group to include the host within it. For example, if the custom_tag file in the /opt/cloud/lts directory shows the host's identifier as test1 , simply add test1 to the group's custom identifiers.	<ul style="list-style-type: none"> - Add the host group's custom identifier to the custom_tag file in the /opt/cloud/lts directory on the host to include the host within the host group. For example, if the group's custom identifier is test, enter test into the custom_tag file. - If the group has multiple custom identifiers, simply enter any one of them into the custom_tag file of the /opt/cloud/lts directory on the host.
Windows host	View the host's identifier in the custom_tag file of the C:\opt\cloud\lts directory on the host. Then, add the identifier to the host group to include the host within it. For example, if the custom_tag file in the C:\opt\cloud\lts directory shows the host's identifier as test1 , simply add test1 to the group's custom identifiers.	<ul style="list-style-type: none"> - Add the host group's custom identifier to the custom_tag file in the C:\opt\cloud\lts directory on the host to include the host within the host group. For example, if the group's custom identifier is test, enter test into the custom_tag file. - If the group has multiple custom identifiers, simply enter any one of them into the custom_tag file of the C:\opt\cloud\lts directory on the host.





----End

Modifying a Host Group

You can change the name of a host group, add hosts to or remove hosts from a host group, or associate a host group with log ingestion configurations. For details, see [Table 3-2](#).

Table 3-2 Operations on host groups

Operation	Procedure
Changing a host group name	<ol style="list-style-type: none"> Go to the Host Groups page. In the host group list, click the modification button in the Operation column of the target host group. On the displayed dialog box, modify the information such as the host group name and custom identifier. Click OK.
Adding hosts to a host group	<p>Method 1:</p> <ol style="list-style-type: none"> In the host group list, click  in the row containing the target host group whose type is IP. Click Add Host. In the displayed slide-out panel, all hosts that are not in the host group and run the selected OS type are displayed. Select the hosts to be added to the host group. <ul style="list-style-type: none"> You can filter hosts by host name or host IP address. You can also click  and enter multiple host IP addresses in the displayed search box to search for matches. If your desired hosts are not in the list, click Install ICAgent. On the displayed page, install ICAgent on the hosts as prompted. For details, see Installing ICAgent. Click OK. <p>Method 2:</p> <ol style="list-style-type: none"> Choose Host Management > Hosts in the navigation pane. In the host list, select the target hosts and click Add to Host Group. In the displayed slide-out panel, select the target host group. Click OK.

Operation	Procedure
Removing a host from a host group	<ol style="list-style-type: none"> In the host group list, click  in the row containing the target host group. In the host list, click Remove in the Operation column of the row containing the host to be removed. In the displayed dialog box, click OK. <p>NOTE This operation is not supported for hosts in the custom identifier host group.</p>
Uninstalling ICAgent from a host	<ol style="list-style-type: none"> In the host group list, click  in the row containing the target host group. In the host list, click Uninstall ICAgent in the Operation column of the row containing the target host. In the displayed dialog box, click OK to uninstall ICAgent from the host and remove the host from the host group. <p>NOTE</p> <ul style="list-style-type: none"> This operation is not supported for hosts in the custom identifier host group. If the host has also been added to other host groups, it will be removed from those groups as well.
Removing hosts from a host group	<ol style="list-style-type: none"> In the host group list, click  in the row containing the target host group. In the host list, select the target hosts and click the Remove button above the list. Click OK.
Associating a host group with an ingestion configuration	<ol style="list-style-type: none"> In the host group list, click  in the row containing the target host group. Click the Associated Ingestion Configuration tab. Click Associate. In the displayed slide-out panel, select the target ingestion configuration. Click OK. The associated ingestion configuration is displayed in the list.
Disassociating a host group from an ingestion configuration	<ol style="list-style-type: none"> On the Associated Ingestion Configuration tab, click Disassociate in the Operation column of the row containing the target ingestion configuration. Click OK.

Operation	Procedure
Disassociating a host group from multiple ingestion configurations	<ol style="list-style-type: none"> 1. On the Associated Ingestion Configuration tab, select the target ingestion configurations and click the Disassociate button above the list. 2. Click OK.
Copying a host group ID	Hover your cursor over a host group name to copy the host group ID.
Exporting host information	<ol style="list-style-type: none"> 1. On the Hosts page, switch to the Intra-Region Hosts, CCE Cluster, or Extra-Region Hosts tab and select the desired hosts. 2. Click Export to export the information of the selected hosts to the local PC.

Deleting Host Groups

Step 1 Choose **Host Management > Host Groups** in the navigation pane.

Step 2 Delete a host group:

1. Click the deletion icon in the **Operation** column of the row containing the target host group.
2. In the displayed dialog box, click **OK**.

Step 3 Delete host groups in batches:

1. Select host groups to be deleted and click **Delete** above the list.
2. In the displayed dialog box, click **OK**.

----End

3.2 Managing Hosts

3.2.1 Installing ICAgent (Intra-Region Hosts)

To use LTS to collect logs (such as host metrics, container metrics, node logs, container logs, and standard output logs) from intra-region hosts, you need to install ICAgent on the hosts. ICAgent is a log collection tool for LTS. It runs on hosts where logs need to be collected.

Prerequisites

Before installing ICAgent, ensure that the time and time zone of your local browser are consistent with those of the host. If they are inconsistent, errors may occur during log reporting.

Installation Methods

There are two methods to install ICAgent.

Table 3-3 Installation methods

Method	Scenario
Initial installation	You can use this method to install ICAgent on a host that has no ICAgent installed. Run the ps -aux grep icagent command on the host to check whether there is an ICAgent process. If no, the ICAgent has not been installed.
Inherited installation (supported only for Linux hosts)	When ICAgent has already been installed on one host but needs to be installed on multiple hosts, you can use this method.

Initial Installation (Linux)

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Host Management > Hosts** in the navigation pane.
- Step 3** Click **Install ICAgent** in the upper right corner.

 **NOTE**

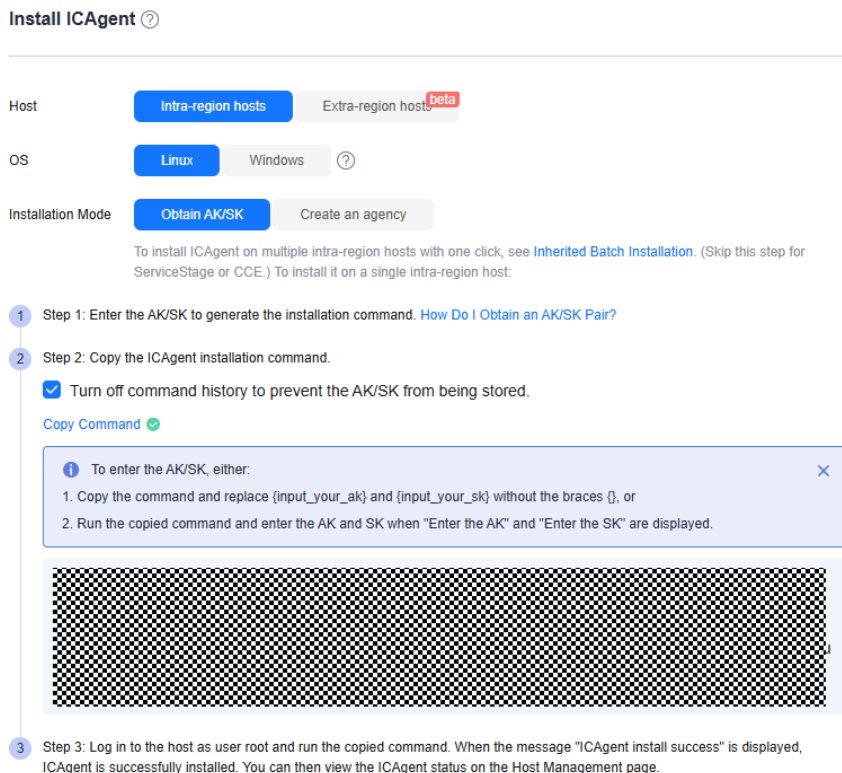
Before installing ICAgent, ensure that the time and time zone of your local browser are consistent with those of the host.

Table 3-4 Installing ICAgent

Parameter	Description	Example
Host	Intra-region hosts is selected by default. Check whether the host whose logs need to be collected is in or out of the region.	-
OS	Linux is selected by default.	-

Parameter	Description	Example
Installation Mode	<ul style="list-style-type: none"> If you select Obtain AK/SK, you need to obtain the AK/SK in advance. An AK is used together with an SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct. For details, see How Do I Obtain an Access Key (AK/SK)? <p>NOTE Ensure that the public account and AK/SK will not be deleted or disabled. If the AK/SK is deleted, the ICAgent cannot report data to LTS.</p> <ul style="list-style-type: none"> If you select Create an agency, you do not need to obtain and enter the AK/SK. ICAgent automatically obtains the AK/SK during agency creation. For details, see How Do I Install ICAgent by Creating an Agency? 	-

Figure 3-4 Installing ICAgent



Step 4 On the **Install ICAgent** page, click **Copy Command** to copy the ICAgent installation command.

Step 5 Log in as user **root** to the host which is deployed in the region same as that you are logged in to (by using a remote login tool such as PuTTY) and run the copied

command. If you have chosen **Obtain AK/SK** as the installation mode, enter the AK/SK as prompted.

NOTE

- When message **ICAgent install success** is displayed, ICAgent has been installed in the **/opt/oss/servicemgr/** directory of the host.
- If the installation fails, uninstall ICAgent and then install it again.

Step 6 Choose **Host Management > Hosts** in the navigation pane of the LTS console and check whether the ICAgent status is **Running**.

----End

Inherited Installation (Linux)

Assume that you need to install ICAgent on multiple hosts, and one of the hosts already has ICAgent installed. The ICAgent installation package, **ICProbeAgent.tar.gz**, is in the **/opt/ICAgent/** directory. To install ICAgent on other hosts one by one:

1. Run the following command on the host where ICAgent has been installed, where *x.x.x.x* is the IP address of the host you want to install ICAgent on.

```
bash /opt/oss/servicemgr/ICAgent/bin/remotelInstall/remote_install.sh -ip x.x.x.x
```
2. Enter the password for user **root** of the host when prompted.

NOTE

- If the Expect tool is installed on the host that has ICAgent installed, the ICAgent installation should be able to complete without prompting you for a password. Otherwise, enter the password as prompted.
- Ensure that user **root** can run SSH or SCP commands on the host where ICAgent has been installed to remotely communicate with the remote host to install ICAgent.
- When message **ICAgent install success** is displayed, ICAgent has been installed in the **/opt/oss/servicemgr/** directory of the host. You can then choose **Host Management > Hosts** in the navigation pane of the LTS console to check the ICAgent status.
- If **ICAgent install success** is not displayed, the installation fails. Uninstall ICAgent and install it again.

Batch Inherited Installation (Linux)

Assume that you need to install ICAgent on multiple hosts, and one of the hosts already has ICAgent installed. The ICAgent installation package, **ICProbeAgent.tar.gz**, is in the **/opt/ICAgent/** directory. In this case, you can follow the directions below to install ICAgent on other hosts in batches.

- The hosts must all belong to the same VPC and be on the same subnet.
- **Python 3.*** is required for batch installation. If you are prompted that Python cannot be found during ICAgent installation, install Python of a proper version on the host and try again.

Prerequisites

The IP addresses and **root**'s passwords of all hosts to install ICAgent have been collected, sorted in the **iplist.cfg** file, and uploaded to the **/opt/ICAgent/** directory

on the host that has ICAgent installed. An IP address and the password of a host's user **root** in the **iplist.cfg** file must be separated by a space. Examples:

192.168.0.109 Password (Replace the IP address and password with the actual ones)

192.168.0.39 Password (Replace the IP address and password with the actual ones)

NOTE

- The **iplist.cfg** file contains sensitive information. You are advised to clear it after using it.
- If all hosts share a password, list only IP addresses in the **iplist.cfg** file and enter the password manually during execution. If one of the hosts uses a different password, type the password behind its IP address.

Procedure

1. Run the following command on the host that has ICAgent installed:

```
bash /opt/oss/servicemgr/ICAgent/bin/remotelInstall/remotel_install.sh -batchModeConfig /opt/ICAgent/iplist.cfg
```

Enter the default password for user **root** of the hosts to install ICAgent. If the passwords of all hosts have been configured in the **iplist.cfg** file, press **Enter** to skip this step.

```
batch install begin
Please input default passwd:
send cmd to 192.168.0.109
send cmd to 192.168.0.39
2 tasks running, please wait...
2 tasks running, please wait...
2 tasks running, please wait...
End of install agent: 192.168.0.39
End of install agent: 192.168.0.109
All hosts install icagent finish.
```

Wait for a while. When message **All hosts install icagent finish.** is displayed, ICAgent has been installed on all the hosts listed in the configuration file.

2. Choose **Host Management > Hosts** in the navigation pane of the LTS console to check the **ICAgent status**.

3.2.2 Installing ICAgent (Extra-Region Hosts)

An extra-region host refers to a host located outside the current Huawei Cloud region or a non-Huawei Cloud host. This category includes hosts in self-built Internet Data Centers (IDCs), those provided by third parties, and those in other Huawei Cloud regions. To collect logs from extra-region hosts to LTS, ensure that the hosts can communicate with LTS located in the current Huawei Cloud region, and then install ICAgent on the hosts. ICAgent is a log collection tool for LTS. It runs on hosts where logs need to be collected.

Extra-region hosts use two types of network channels to report logs to LTS:

- **Public network:** An extra-region host connected to the public network can communicate with and report logs to LTS in the current Huawei Cloud region through the public network. However, for security reasons, private lines are usually preferred in actual production environments.

 **NOTE**

If you select **Public network** when installing ICAgent on an extra-region host, ensure that the region support log reporting via public networks.

- Private line: Extra-region hosts connect to LTS in the current Huawei Cloud region through a jump server or VPCEP, offering greater security and stability. In this scenario, extra-region hosts cannot communicate with LTS in the current region by default, and ICAgent installed on these hosts cannot directly access the network segment used by the Huawei Cloud management plane to report logs. Therefore, you need to configure a network connection solution to use a jump server or VPCEP to connect to the LTS backend and forward data to LTS.
 - Jump server: functions as a data forwarder and forwards the data collected by ICAgent from extra-region hosts to LTS. This solution is suitable for tests or scenarios with low log traffic. VPCEP is recommended for scenarios with high log traffic.
 - VPCEP: provides convenient and secure channels to connect to LTS in the current region, enabling resources in the VPC to access VPCEP without the need for EIPs. This solution reduces the risks of data transmission on public networks and improves the transmission security and efficiency.

Prerequisites

Before installing ICAgent, ensure that the time and time zone of your local browser are consistent with those of the host. If they are inconsistent, errors may occur during log reporting.

Constraints

- Linux: ICAgent can only be installed on hosts running the Linux OSs listed in [OS Usage Restrictions](#).
- Windows: ICAgent can only be installed on hosts running the following 64-bit Windows OSs:

Windows Server 2016 R2 Datacenter
Windows Server 2016 R2 Standard
Windows Server 2016 Datacenter English
Windows Server 2016 R2 Standard English

Windows Server 2012 R2 Datacenter
Windows Server 2012 R2 Standard
Windows Server 2012 Datacenter English
Windows Server 2012 R2 Standard English

Windows Server 2008 R2 Enterprise
Windows Server 2008 R2 Standard
Windows Server 2008 Enterprise English
Windows Server 2008 R2 Standard English

ICAgent installed on Windows hosts cannot be upgraded or uninstalled on the **Host Management** page. To use a newer ICAgent version, uninstall the original one and then install the newer version.

Installation Methods

There are two methods to install ICAgent.

Table 3-5 Installation methods

Method	Scenario
Initial installation	You can use this method to install ICAgent on a host that has no ICAgent installed.
Inherited installation (supported only for Linux hosts)	When ICAgent has already been installed on one host but needs to be installed on multiple hosts, you can use this method.

Initial Installation (Linux)

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Host Management > Hosts** in the navigation pane.
- Step 3** Set **Host** to **Extra-region hosts**.
- Step 4** Set **OS** to **Linux**.
- Step 5** Set **Network Connectivity**. For extra-region hosts to report logs to LTS in the current region, you are advised to select **Private line** for higher stability and reliability.
- If you select **Public network**, start from [Step 7](#).
 - If you select **Private line**, start from [Step 6](#).
- Step 6** Set **LTS Backend Connection**. If you select **Private line** in the previous step, the extra-region host cannot communicate with LTS in the current region by default, and ICAgent installed on the host cannot directly access the network segment used by the Huawei Cloud management plane to report logs. Therefore, you need to configure a network connection solution to use a jump server or VPCEP to connect to the LTS.
- If you set **LTS Backend Connection** to **VPCEP**:
Configure a VPCEP domain name. With the assistance of Huawei Cloud network engineers, configure DNS domain name resolution rules in other regions to resolve VPCEP domain names to specified IP addresses. Then, copy the command as prompted on the **Install ICAgent** page.

```
ping {VPCEP domain name}
```


Run this command on the host you want to collect logs from. If the ping command succeeds, the network configuration is correct. Proceed to [Step 7](#).
 - If you set **LTS Backend Connection** to **Jump server**:
 - a. Create a Linux ECS as a jump server.
Log in to the ECS console and create a Linux ECS. For details, see [Purchasing an ECS](#). If you have an ECS that meets the requirements for use as a jump server, skip this step.

 NOTE

- The minimum specifications for the ECS are 1 vCPU and 1 GB of memory. The recommended specifications are 2 vCPUs and 4 GB of memory. You are advised to use an image of **CentOS 6.5 64bit** or later version.
 - If the jump server communicates with the extra-region host over the public network, an EIP must be enabled. Conversely, when it uses a VPC peering connection, an EIP is not required.
 - The region of the jump server must be the same as the current region of LTS.
- b. Add security group rules for the jump server and enable the corresponding inbound ports to ensure data connectivity between the extra-region hosts and the jump server.
- i. Log in to the ECS console, check the ECS list, and locate the ECS that you created as the jump server.
 - ii. Click its name to go to the ECS details page. Click the security group name to access the security group details page.
 - iii. Click the **Inbound Rules** tab and click **Add Rule**. Set the ports by referring to **Table 3-6**. Set other parameters based on your network requirements.

Table 3-6 Security group rule

Action	Protocol	Port	Description
All ow	TCP	8149,8102,8923,3020 0,30201,80	Ports used by ICAgent to send data to the jump server, ensuring data connectivity between hosts in other regions and the jump server.

- c. Return to the ECS list, locate the ECS created in **Step 6.a**, and view its private IP address and EIP (available if an EIP has been enabled).
- d. Go back to the LTS console. On the **Install ICAgent** page, enter the obtained private IP address of the jump server to generate its SSH tunneling command.

 NOTE

The private IP address of the jump server refers to the internal IP address of the VPC where the jump server is located.

- e. On the **Install ICAgent** page, click **Copy Command** to copy the SSH tunneling command.

```
ssh -f -N -L {Private IP address of the jump server}:8149:{LTS reporting IP address}:8149 -L {Private IP address of the jump server}:8102:{LTS reporting IP address}:8102 -L {Private IP address of the jump server}:8923:{LTS reporting IP address}:8923 -L {Private IP address of the jump server}:30200:{LTS reporting IP address}:30200 -L {Private IP address of the jump server}:30201:{LTS reporting IP address}:30201 -L {Private IP address of the jump server}:80:icagent-{Region}://{OBS domain name}:80 {Private IP address of the jump server}
```

- f. Log in to the jump server as user **root** and run the copied SSH tunneling command.

- g. Run the `netstat -lnp | grep ssh` command to check whether the corresponding TCP ports are being listened to. If the command output similar to [Figure 3-5](#) is returned, the ports are open.

Figure 3-5 Open TCP ports

```
root@ecs-debc-qff-tiaobanji ~]# netstat -lnp | grep ssh
tcp        0      0 192.168.0.79:8102    0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:80     0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:8149   0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      1772/sshd
tcp        0      0 192.168.0.79:30200  0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:30201  0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:8923   0.0.0.0:*           LISTEN      1994/sshd
tcp6       0      0 :::22               :::*                 LISTEN      1772/sshd
```

 NOTE

If the jump server powers off and restarts, run the `netstat -lnp | grep ssh` command again.

- h. On the **Install ICAgent** page, enter the DC and the connection IP address of the jump server.
- **DC:** Specify a name for the data center of the host so it is easier to find the host. Enter up to 64 characters. Use only digits, letters, hyphens (-), and underscores (_).
 - **Connection IP:** If the jump server communicates with the extra-region host via EIP connection, enter the EIP of the jump server. Conversely, when using a VPC peering connection, enter the internal IP address (private IP address) of the VPC where the jump server locates. For the EIP and private IP address, see [Step 6.c](#).

Step 7 Obtain an AK/SK. On the **Install ICAgent** page, copy the ICAgent installation command and replace the AK/SK in the command with the obtained one.

Step 8 Log in to the extra-region host as user **root** (by using a remote login tool such as PuTTY) and run the copied command.

When message **ICAgent install success** is displayed, ICAgent has been installed in the `/opt/oss/servicemgr/` directory of the host. You can then choose **Host Management > Hosts** in the navigation pane of the LTS console to check the ICAgent status.

 NOTE

If the installation fails, uninstall ICAgent and reinstall it. If the reinstallation fails, contact technical support.

----End

Initial Installation (Windows)

Step 1 Log in to the LTS console and choose **Host Management** in the navigation pane.

Step 2 Choose **Host Management > Hosts** in the navigation pane.

Step 3 Set **Host** to **Extra-region hosts**.

Step 4 Set **OS** to **Windows**.

Step 5 Set **Network Connectivity**. Extra-region hosts report logs to LTS via a public network or a private line. The latter is recommended for higher stability and reliability.

- If you select **Public network**, start from [Step 7](#).
- If you select **Private line**, start from [Step 6](#).

Step 6 Set **LTS Backend Connection**. If you select **Private line** in the previous step, the extra-region host cannot communicate with LTS in the current region by default, and ICAgent installed on the host cannot directly access the network segment used by the Huawei Cloud management plane to report logs. Therefore, you need to configure a network connection solution to use a jump server or VPCEP to connect to the LTS.

- If you set **LTS Backend Connection** to **VPCEP**:

Configure a VPCEP domain name. With the assistance of Huawei Cloud network engineers, configure DNS domain name resolution rules in other regions to resolve VPCEP domain names to specified IP addresses. Then, copy the command as prompted on the **Install ICAgent** page.

```
ping {VPCEP domain name}
```

Run this command on the host you want to collect logs from. If the ping command succeeds, the network configuration is correct. Proceed to [Step 7](#).

- If you set **LTS Backend Connection** to **Jump server**:

a. Create a Linux ECS as a jump server.

Log in to the ECS console and create a Linux ECS. For details, see [Purchasing an ECS](#). If you have an ECS that meets the requirements for use as a jump server, skip this step.

NOTE

- The minimum specifications for the ECS are 1 vCPU and 1 GB of memory. The recommended specifications are 2 vCPUs and 4 GB of memory. You are advised to use an image of **CentOS 6.5 64bit** or later version.
 - If the jump server communicates with the extra-region host over the public network, an EIP must be enabled. Conversely, when it uses a VPC peering connection, an EIP is not required.
 - The region of the jump server must be the same as the current region of LTS.
- b. Add security group rules for the jump server and enable the corresponding inbound ports to ensure data connectivity between the extra-region hosts and the jump server.
- i. Log in to the ECS console, check the ECS list, and locate the ECS that you created as the jump server.
 - ii. Click its name to go to the ECS details page. Click the security group name to access the security group details page.
 - iii. Click the **Inbound Rules** tab and click **Add Rule**. Set the ports by referring to [Table 3-7](#). Set other parameters based on your network requirements.

Table 3-7 Security group rule

Action	Protocol	Port	Description
Allow	TCP	8149,8102,8923,30200,30201,80	Ports used by ICAgent to send data to the jump server, ensuring data connectivity between hosts in other regions and the jump server.

- c. Return to the ECS list, locate the ECS created in [Step 6.a](#), and view its private IP address and EIP (available if an EIP has been enabled).
- d. Go back to the LTS console. On the **Install ICAgent** page, enter the obtained private IP address of the jump server to generate its SSH tunneling command.

NOTE

The private IP address of the jump server refers to the internal IP address of the VPC where the jump server is located.

- e. On the **Install ICAgent** page, click **Copy Command** to copy the SSH tunneling command.


```
ssh -f -N -L {Private IP address of the jump server}:8149:{LTS reporting IP address}:8149 -L {Private IP address of the jump server}:8102:{LTS reporting IP address}:8102 -L {Private IP address of the jump server}:8923:{LTS reporting IP address}:8923 -L {Private IP address of the jump server}:30200:{LTS reporting IP address}:30200 -L {Private IP address of the jump server}:30201:{LTS reporting IP address}:30201 -L {Private IP address of the jump server}:80:icagent-{Region}:{OBS domain name}:80 {Private IP address of the jump server}
```
- f. Log in to the jump server as user **root** and run the copied SSH tunneling command.
- g. Run the **netstat -lnp | grep ssh** command to check whether the corresponding TCP ports are being listened to. If the command output similar to [Figure 3-6](#) is returned, the ports are open.

Figure 3-6 Open TCP ports

```
root@ecs-debc-qff-tiaobanji ~# netstat -lnp | grep ssh
tcp        0      0 192.168.0.79:8102    0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:80     0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:8149   0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      1772/sshd
tcp        0      0 192.168.0.79:30200  0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:30201  0.0.0.0:*           LISTEN      1994/sshd
tcp        0      0 192.168.0.79:8923   0.0.0.0:*           LISTEN      1994/sshd
tcp6       0      0 :::22               :::*                 LISTEN      1772/sshd
```

NOTE

If the jump server powers off and restarts, run the preceding command again.

Step 7 Click the link on the **Install ICAgent** page to download the ICAgent installation package.

Step 8 Save the ICAgent installation package to a directory on the Windows host, for example, **C:\ICAgent**, and decompress the package.

Step 9 Obtain an AK/SK and save it to replace the AK/SK in the installation command.

If you set **LTS Backend Connection to Jump Server**, you also need to set **Connection IP**.

Connection IP: If the jump server communicates with the extra-region host via EIP connection, enter the EIP of the jump server. Conversely, when using a VPC peering connection, enter the internal IP address (private IP address) of the VPC where the jump server locates. For the EIP and private IP address, see [Step 6.c](#).

Step 10 On the **Install ICAgent** page, click **Copy Command** to copy the ICAgent installation command.

Step 11 Log in to the Windows host, open the Command Prompt, go to the directory where the ICAgent installation package is decompressed, and run the ICAgent installation command.

If the message **Service icagent installed successfully** is displayed, the installation is successful. You can then choose **Host Management > Hosts** in the navigation pane of the LTS console to check the ICAgent status.

 **NOTE**

If the installation fails, uninstall ICAgent and reinstall it. If the reinstallation fails, contact technical support.

----End

Creating Multiple Jump Servers for Load Balancing Using ELB

A single jump server may encounter a single point of failure (SPOF), potentially leading to O&M instability. To mitigate this, you can create multiple jump servers and use ELB to distribute traffic among them, enhancing access reliability.

Step 1 Create a Linux ECS as a jump server.

Log in to the ECS console and create a Linux ECS. For details, see [Purchasing an ECS](#). If you have an ECS that meets the requirements for use as a jump server, skip this step.

 **NOTE**

- The minimum specifications for the ECS are 1 vCPU and 1 GB of memory. The recommended specifications are 2 vCPUs and 4 GB of memory. You are advised to use an image of **CentOS 6.5 64bit** or later version.
- If the jump server communicates with the extra-region host over the public network, an EIP must be enabled. Conversely, when it uses a VPC peering connection, an EIP is not required.
- The region of the jump server must be the same as the current region of LTS.

Step 2 Add security group rules for the jump server and enable the corresponding inbound ports to ensure data connectivity between the extra-region hosts and the jump server.

1. Log in to the ECS console, check the ECS list, and locate the ECS that you created as the jump server.
2. Click its name to go to the ECS details page. Click the security group name to access the security group details page.

3. Click the **Inbound Rules** tab and click **Add Rule**. Set the ports by referring to [Table 3-8](#). Set other parameters based on your network requirements.

Table 3-8 Security group rule

Action	Protocol	Port	Description
Allow	TCP	8149,8102,8923,30200,30201,80	Ports used by ICAgent to send data to the jump server, ensuring data connectivity between hosts in other regions and the jump server.

- Step 3** Return to the ECS list, locate the ECS created in [Step 1](#), and view its private IP address and EIP (available if an EIP has been enabled).
- Step 4** Log in to the LTS console. In the navigation pane, choose **Host Management** > **Hosts**. Click **Install ICAgent**. On the displayed page, enter the private IP address of the jump server to generate the SSH tunneling command.

NOTE

The private IP address of the jump server refers to the internal IP address of the VPC where the jump server is located.

- Step 5** On the **Install ICAgent** page, click **Copy Command** to copy the SSH tunneling command.

```
ssh -f -N -L {Private IP address of the jump server}:8149:{LTS reporting IP address}:8149 -L {Private IP address of the jump server}:8102:{LTS reporting IP address}:8102 -L {Private IP address of the jump server}:8923:{LTS reporting IP address}:8923 -L {Private IP address of the jump server}:30200:{LTS reporting IP address}:30200 -L {Private IP address of the jump server}:30201:{LTS reporting IP address}:30201 -L {Private IP address of the jump server}:80:icagent-{Region}.{OBS domain name}:80 {Private IP address of the jump server}
```

- Step 6** Log in to the jump server as user **root** and run the copied SSH tunneling command.
- Step 7** Repeat the preceding steps to create multiple jump servers. Add them to the same VPC by selecting the same VPC for **Network** during their creation.
- Step 8** Log in to the ELB console and create a load balancer. For details, see [Creating a Dedicated Load Balancer](#). When creating the load balancer, you should:
1. Select the same VPC as that of the jump servers during network configuration.
 2. Create an EIP for connecting to the jump servers.
 3. Apply for the bandwidth based on the service requirements.
- Step 9** Add listeners for TCP ports 30200, 30201, 8149, 8923, 8102, and 80. For details, see [Adding a TCP Listener](#).
- Step 10** Create a backend server group and add all jump servers in the group. For details, see [Adding Backend Servers](#).

- Step 11** Return to the LTS console. On the **Install ICAgent** page, enter the EIP of the load balancer in **Connection IP**, copy the installation command, and run it on the extra-region host.

----End

Inherited Installation (Linux)

Assume that you need to install ICAgent on multiple hosts, and one of the hosts already has ICAgent installed. The ICAgent installation package, **ICProbeAgent.tar.gz**, is in the **/opt/ICAgent/** directory. To install ICAgent on other hosts one by one:

1. Run the following command on the host where ICAgent has been installed, where *x.x.x.x* is the IP address of the host you want to install ICAgent on.

```
bash /opt/oss/servicemgr/ICAgent/bin/remotelInstall/remote_install.sh -ip x.x.x.x
```
2. Enter the password for user **root** of the host when prompted.

NOTE

- If the Expect tool is installed on the host that has ICAgent installed, the ICAgent installation should be able to complete without prompting you for a password. Otherwise, enter the password as prompted.
 - Ensure that user **root** can run SSH or SCP commands on the host where ICAgent has been installed to remotely communicate with the remote host to install ICAgent.
 - If the installation fails, uninstall ICAgent and reinstall it. If the reinstallation fails, contact technical support.
3. When message **ICAgent install success** is displayed, ICAgent has been installed in the **/opt/oss/servicemgr/** directory of the host. You can then choose **Host Management > Hosts** in the navigation pane of the LTS console to check the ICAgent status.

Batch Inherited Installation (Linux)

Assume that you need to install ICAgent on multiple hosts, and one of the hosts already has ICAgent installed. The ICAgent installation package, **ICProbeAgent.tar.gz**, is in the **/opt/ICAgent/** directory. In this case, you can follow the directions below to install ICAgent on other hosts in batches.

- The hosts must all belong to the same VPC and be on the same subnet.
- **Python 3.*** is required for batch installation. If you are prompted that Python cannot be found during ICAgent installation, install Python of a proper version on the host and try again.

Prerequisites

The IP addresses and **root**'s passwords of all hosts to install ICAgent have been collected, sorted in the **iplist.cfg** file, and uploaded to the **/opt/ICAgent/** directory on the host that has ICAgent installed. An IP address and user **root**'s password in the **iplist.cfg** file must be separated by a space. Examples:

192.168.0.109 Password (Replace the IP address and password with the actual ones)

192.168.0.39 Password (Replace the IP address and password with the actual ones)

 NOTE

- The **iplist.cfg** file contains sensitive information. You are advised to clear it after using it.
- If all hosts share a password, list only IP addresses in the **iplist.cfg** file and enter the password manually during execution. If one of the hosts uses a different password, type the password behind its IP address.

Procedure

1. Run the following command on the host that has ICAgent installed:

```
bash /opt/oss/servicemgr/ICAgent/bin/remotelInstall/remote_install.sh -batchModeConfig /opt/ICAgent/iplist.cfg
```

Enter the default password for user **root** of the hosts to install ICAgent. If the passwords of all hosts have been configured in the **iplist.cfg** file, press **Enter** to skip this step.

```
batch install begin
Please input default passwd:
send cmd to 192.168.0.109
send cmd to 192.168.0.39
2 tasks running, please wait...
2 tasks running, please wait...
2 tasks running, please wait...
End of install agent: 192.168.0.39
End of install agent: 192.168.0.109
All hosts install icagent finish.
```

If the message **All hosts install icagent finish.** is displayed, ICAgent has been installed on all the hosts listed in the configuration file.

2. After the installation is complete, choose **Host Management > Hosts** in the navigation pane to view the host status. For details, see [Checking the ICAgent Status](#).

3.2.3 Managing ICAgent

After ICAgent is installed, you can [upgrade](#) and [uninstall](#) it, and [view its status](#).

Upgrading ICAgent

To deliver a better collection experience, LTS regularly upgrades ICAgent. When LTS prompts you that a new ICAgent version is available, you can follow the directions here to obtain the latest version.

Linux hosts support ICAgent upgrade on the LTS console.

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Host Management > Hosts** in the navigation pane.
- Step 3** Select **Intra-Region Hosts** or **Extra-Region Hosts**. When the system prompts you that a new ICAgent version is available, select one or more check boxes of hosts where ICAgent is to be upgraded, and click **Upgrade ICAgent**.
- Step 4** Click the **CCE Cluster** tab. Search for and select the cluster whose ICAgent is to be upgraded, and click **Upgrade ICAgent**.
 - If you create a CCE cluster for the first time, ICAgent will be installed on hosts in the cluster by default, and logs will be reported to AOM. **Output to AOM**

is enabled by default. To report logs to LTS, disable **Output to AOM** before upgrading ICAgent. You are advised to choose **Log Ingestion > Cloud Service > Cloud Container Engine (CCE)** to collect container data and output it to LTS instead of AOM.

- CCE cluster ID (ClusterID): Each cluster has a fixed ID.
- When ICAgent is upgraded, LTS creates log groups and host groups for your CCE cluster. The name of the log group and host group is **k8s-log-*{ClusterID}***. You can create an ingestion configuration (**Cloud Services > Cloud Container Engine (CCE)**) to add logs of the current CCE cluster to the log group.
- If the hosts of a cluster have no ICAgent installed or outdated ICAgent installed, click **Upgrade ICAgent** to install or upgrade ICAgent on all hosts in the cluster.

Step 5 In the displayed dialog box, click **OK**.

The upgrade begins. This process takes about a minute. When the ICAgent status changes from **Upgrading** to **Running**, the ICAgent upgrade has completed.

If the ICAgent is abnormal after the upgrade or if the upgrade fails, log in to the host and run the installation command. ICAgent can be re-installed on top of itself.

----End

Uninstalling ICAgent

If ICAgent is uninstalled from a host, log collection will be affected. Exercise caution when performing this operation.

NOTE

Uninstalling ICAgent does not delete the installation files. You need to delete them manually if necessary.

You can uninstall ICAgent using either of the following methods:

- Uninstalling ICAgent on the console: applies to the scenario where ICAgent has been successfully installed.
 - a. Choose **Host Management > Hosts** in the navigation pane.
 - b. Select one or more hosts where ICAgent is to be uninstalled and click **Uninstall ICAgent**.
 - c. In the displayed dialog box, click **OK**.

The uninstallation begins. This process takes about a minute.

The hosts with ICAgent uninstalled will be removed from the host list.

NOTE

To reinstall ICAgent, wait for 5 minutes after it is uninstalled. Otherwise, the ICAgent may be automatically uninstalled again.

- Logging in to the host to uninstall ICAgent: applies to the scenario where ICAgent fails to be installed.
 - a. Log in to a host where ICAgent is to be uninstalled as user **root**.
 - b. Run the following command:

```
bash /opt/oss/servicemgr/ICAgent/bin/manual/uninstall.sh
```

If the message **ICAgent uninstall success** is displayed, the uninstallation has completed.

- Remotely uninstalling ICAgent: applies to the scenario where the ICAgent has been successfully installed and needs to be remotely uninstalled.

You can uninstall ICAgent on one host remotely from another host.

- a. Run the following command on the host where ICAgent has been installed. *x.x.x.x* indicates the IP address of the host you want to uninstall ICAgent from.

```
bash /opt/oss/servicemgr/ICAgent/bin/remoteUninstall/remote_uninstall.sh -ip x.x.x.x
```

- b. Enter the password for user **root** of the host when prompted.

NOTE

- If the Expect tool is installed on the host that has ICAgent installed, the ICAgent uninstallation should be able to complete without prompting you for a password. Otherwise, enter the password as prompted.
 - Ensure that user **root** can run SSH or SCP commands on the host where ICAgent has been installed to communicate with the remote host.
 - If the message **ICAgent uninstall success** is displayed, the uninstallation has completed.
- Batch uninstalling ICAgent: applies to the scenario where the ICAgent has been installed and needs to be uninstalled in batches.

If ICAgent has been installed on a host and the ICAgent installation package **ICProbeAgent.tar.gz** is in the **/opt/ICAgent/** directory of the host, you can use this method to uninstall ICAgent from multiple hosts at once.

NOTE

The hosts must all belong to the same VPC and be on the same subnet.

Prerequisites

The IP addresses and passwords of all hosts to uninstall ICAgent have been collected, sorted in the **iplist.cfg** file, and uploaded to the **/opt/ICAgent/** directory on the host that has ICAgent installed. Each IP address and password in the **iplist.cfg** file must be separated by a space. Examples:

192.168.0.109 Password (Replace the IP address and password with the actual ones)

192.168.0.39 Password (Replace the IP address and password with the actual ones)

NOTE

- The **iplist.cfg** file contains sensitive information. You are advised to clear it after using it.
 - If all hosts share a password, list only IP addresses in the **iplist.cfg** file and enter the password manually during execution. If one of the hosts uses a different password, type the password behind its IP address.
- a. Run the following command on the host that has ICAgent installed:

```
bash /opt/oss/servicemgr/ICAgent/bin/remoteUninstall/remote_uninstall.sh -batchModeConfig /opt/ICAgent/iplist.cfg
```

Enter the default password for user **root** of the hosts to uninstall ICAgent. If the passwords of all hosts have been configured in the **iplist.cfg** file, press **Enter** to skip this step.

```
batch uninstall begin
Please input default passwd:
send cmd to 192.168.0.109
send cmd to 192.168.0.39
2 tasks running, please wait...
End of uninstall agent: 192.168.0.109
End of uninstall agent: 192.168.0.39
All hosts uninstall icagent finish.
```

If message **All hosts uninstall icagent finish.** is displayed, the batch uninstallation has completed.

- b. Choose **Host Management > Hosts** in the navigation pane of the LTS console to view the ICAGENT status.

Checking the ICAGENT Status

Choose **Host Management > Hosts** in the navigation pane to check the ICAGENT status of the target host. The following table lists the ICAGENT statuses.

Table 3-9 ICAGENT statuses

Status	Description
Running	ICAGENT is running properly.
Uninstalled	ICAGENT is not installed.
Installing	ICAGENT is being installed. This process takes about one minute.
Installation failed	ICAGENT installation failed.
Upgrading	ICAGENT is being upgraded. This process takes about one minute.
Upgrade failed	ICAGENT upgrade failed.
Offline	ICAGENT is abnormal because the AK/SK pair is incorrect. Obtain the correct AK/SK pair and install ICAGENT again.
Faulty	ICAGENT is faulty. Contact technical support.
Uninstalling	ICAGENT is being uninstalled. This process takes about one minute.
Authentication error	Authentication fails because parameters were incorrectly configured during ICAGENT installation.

4 Log Search and Analysis

4.1 Overview

Log search and analysis are indispensable to O&M. After configuring log ingestion, you can search and analyze the collected log data on LTS. Its efficient and professional log collection, search, and analysis help you monitor and manage your systems and applications.

- To facilitate search and analysis, configure structuring and indexing for the reported logs first to ensure the log length and format meet the requirements. For details, see [Setting Cloud Structuring Parsing](#) and [Setting Indexes](#).
- After structuring the logs, use LTS [search syntax](#) to set search criteria for higher efficiency. For details, see [Searching Logs](#).

4.2 Setting Cloud Structuring Parsing

4.2.1 Overview

Log data can be structured or unstructured. Structured data is quantitative data or can be defined by unified data models. It has a fixed length and format. Unstructured data has no pre-defined data models and cannot be fit into two-dimensional tables of databases.

During log structuring, logs with fixed or similar formats are extracted from a log stream based on your defined structuring method and irrelevant logs are filtered out.

Log structuring parsing is a process of converting log data from unstructured or semi-structured to structured for better storage, query, and analysis, improving log data readability, searchability, and query efficiency.

Precautions

- Log structuring is performed on a per-log-stream basis.

- Log structuring is recommended when most logs in a log stream share a similar pattern.
- After the structuring configuration is modified, the modification takes effect only for newly written log data.

4.2.2 Setting Cloud Structuring Parsing

LTS provides five log structuring modes: regular expressions, JSON, delimiters, Nginx, and structuring templates. You can make your choice flexibly.

- **Regular Expressions:** This mode applies to scenarios where each line in the log text is a raw log event and each log event can be extracted into multiple key-value pairs based on regular expressions. To use this mode to extract fields, you need to enter a log sample and customize a regular expression. Then, LTS extracts the corresponding key-value pairs based on the capture group in the regular expression.
- **JSON:** This mode applies to scenarios where each line in the log text is a raw log event and each log event can be extracted into multiple key-value pairs based on the JSON parsing rule.
- **Delimiter:** This mode applies to scenarios where each line in the log text is a raw log event and each log event can be extracted into multiple key-value pairs based on specified delimiters (such as colons, spaces, or characters).
- **Nginx:** This mode applies to scenarios where each line in the log text is a raw log event, each log event complies with the Nginx format, and the access log format can be defined by the `log_format` command.
- **Structuring Template:** This mode applies to scenarios where the log structure is complex or key-value extraction needs to be customized. You can use a built-in system template or a custom template to extract fields.

NOTE

- If **Auto Configuration and Analysis** is enabled, structured fields are used for field indexing and **Quick Analysis** is also enabled. The system fields `hostIP`, `hostName`, and `pathFile` are set as index fields by default.
- The following system fields cannot be extracted during log structuring: `groupName`, `logStream`, `lineNum`, `content`, `logContent`, `logContentSize`, `collectTime`, `category`, `clusterId`, `clusterName`, `containerName`, `hostIP`, `hostId`, `hostName`, `nameSpace`, `pathFile`, and `podName`.

Constraints

- If indexing has not been configured, delimiters for structured fields default to being empty. The maximum size of a field is 20 KB, with any excess data being truncated.
- If indexing has been configured, the default delimiters of structured fields are those listed in [Configuring Log Content Delimiters](#). In this case, the maximum size of a field is 500 KB.

Cloud Structuring Parsing

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.



Step 2 Expand the target log group and click the name of the target stream.

Step 3 On the log stream details page, click  in the upper right corner. On the page displayed, click the **Cloud Structuring Parsing** tab to configure log structuring.

 **NOTE**

- If **Raw Log Save** is enabled, raw logs will be stored in LTS as the value of the **content** field. The **content** field is also counted in both resource statistics and charging.
- If **Upload Parsing Failure Log** is enabled, raw logs will be uploaded to LTS as the value of the **_content_parse_fail_** field.
- **Regular Expressions**: Extract fields using regular expressions.
- **JSON**: Extract key-value pairs from JSON log events.
- **Delimiter**: Extract fields using delimiters (such as commas or spaces).
- **Nginx**: Customize the format of access logs by using the **log_format** command.
- **Structuring Template**: Extract fields using a custom or system template.

Step 4 Modify or delete the configured structuring configuration.

- On the **Cloud Structuring Parsing** tab page, click  to modify the structuring configuration.
- On the **Cloud Structuring Parsing** tab page, click  to delete the structuring configuration.

 **NOTE**

Deleted structuring configurations cannot be restored. Exercise caution when performing this operation.

----End

Regular Expressions

If you choose regular expressions, fields are extracted based on your defined regular expressions.

Step 1 Select a typical log event as the sample.

- Click **Select from Existing Logs**, select a log event, and click **OK**. You can select different time ranges to filter logs.
- Click **Paste from Clipboard** to paste the copied log content to the sample log box.

 **NOTE**

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

- From now: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- From last: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified**: queries log data that is generated in a specified time range.

Step 2 Extract fields. Extracted fields are shown with their example values. You can extract fields in two ways:

- **Auto generate**: Select the log content you want to extract as a field in the sample log event. In the dialog box displayed, set the field name. The name must start with a letter and contain only letters and digits. Then click **Add**.
- **Manually enter**: Enter a regular expression in the text box and click **Extract Field**. A regular expression may contain multiple capturing groups, which group strings with parentheses. There are three types of capturing groups:
 - (*exp*): Capturing groups are numbered by counting their opening parentheses from left to right. The numbering starts with 1.
 - (?<*name*>*exp*): named capturing group. It captures text that matches *exp* into the group *name*. The group name must start with a letter and contain only letters and digits. A group is recalled by group name or number.
 - (?*:exp*): non-capturing group. It captures text that matches *exp*, but it is not named or numbered and cannot be recalled.

 **NOTE**

- When you select **manually enter**, the regular expression can contain up to 5000 characters. You do not have to name capturing groups when writing the regular expression. When you click **Extract Field**, those unnamed groups will be named as **field1**, **field2**, **field3**, and so on.

Step 3 Specify a field as the log time. For details, see [Setting Custom Log Time](#).

Step 4 Click **Save**. The type of extracted fields cannot be changed after the structuring is complete.

----End

JSON

If you choose **JSON**, JSON logs are split into key-value pairs.

Step 1 Select a typical log event as the sample. Click **Select from existing log events**, select a log event, or enter a log event in the text box, and click **OK**. You can select different time ranges to filter logs.

 **NOTE**

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

- **From now:** queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- **From last:** queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified:** queries log data that is generated in a specified time range.

Step 2 Extract fields. Extract fields from the log event. Extracted fields are shown with their example values.

Click **Intelligent Extraction**. Take the following log event as an example.

Enter the log event in the text box.

```
{"a1": "a1", "b1": "b1", "c1": "c1", "d1": "d1"}
```

 **NOTE**

- The **float** data type has 16 digit precision. If a value contains more than 16 valid digits, the extracted content is incorrect, which affects visualization and quick analysis. In this case, you are advised to change the field type to **string**.
- If the data type of the extracted fields is set to **long** and the log content contains more than 16 valid digits, only the first 16 valid digits are displayed, and the subsequent digits are changed to 0.
- If the data type of the extracted fields is set to **long** and the log content contains more than 21 valid digits, the fields are identified as the **float** type. You are advised to change the field type to **string**.

Check and edit the fields if needed. For details about rules for configuring extracted fields, see [Setting Structured Fields](#).

Step 3 Specify a field as the log time. For details, see [Setting Custom Log Time](#).

Step 4 Click **Save**. The type of extracted fields cannot be changed after the structuring is complete.

----End

Delimiter

Logs can be parsed by delimiters, such as commas (,), spaces, or other special characters.

Step 1 Select a typical log event as the sample. Click **Select from existing log events**, select a log event, or enter a log event in the text box, and click **OK**. You can select different time ranges to filter logs.

 **NOTE**

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

- From now: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- From last: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified**: queries log data that is generated in a specified time range.

Step 2 Select or customize a delimiter. **NOTE**

- For invisible characters, enter hexadecimal characters starting with 0x. The length ranges from 0 to 4 characters. There are 32 invisible characters in total.
- For custom characters, enter 1 to 10 characters, each as an independent delimiter.
- For a custom string, enter 1 to 30 characters as one whole delimiter.

Step 3 Extract fields. Extract fields from the log event. Extracted fields are shown with their example values.

Click **Intelligent Extraction**. Take the following log event as an example.

Enter the log event in the text box.

```
1 5f67944957444bd6bb4fe3b367de8f3d 1d515d18-1b36-47dc-a983-bd6512aed4bd 192.168.0.154
192.168.3.25 38929 53 17 1 96 1548752136 1548752736 ACCEPT OK
```

 **NOTE**

The **float** data type has seven digit precision.

If a value contains more than seven valid digits, the extracted content is incorrect, which affects visualization and quick analysis. In this case, you are advised to change the field type to **string**.

Check and edit the fields if needed. For details about rules for configuring extracted fields, see [Setting Structured Fields](#).

Step 4 Specify a field as the log time. For details, see [Setting Custom Log Time](#).**Step 5** Click **Save**. The type of extracted fields cannot be changed after the structuring is complete.

----End

Nginx

You can customize the format of access logs by the **log_format** command.

Step 1 Select a typical log event as the sample. Click **Select from existing log events**, select a log event, or enter a log event in the text box, and click **OK**. You can select different time ranges to filter logs.

 NOTE

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

- From now: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- From last: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified**: queries log data that is generated in a specified time range.

Step 2 Define the Nginx log format. You can click **Apply Default Nginx Log Format** to apply the default format,

 NOTE

In standard Nginx configuration files, the portion starting with **log_format** indicates the log configuration.

Log format

- Default Nginx log format:

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for";'
```
- You can also customize a format. The format must meet the following requirements:
 - Cannot be blank.
 - Must start with **log_format** and contain apostrophes (') and field names.
 - Can contain up to 5000 characters.
 - Must match the sample log event.
 - Any character except letters, digits, underscores (_), and hyphens (-) can be used to separate fields.
 - Must end with an apostrophe (') or an apostrophe plus a semicolon (;).

Step 3 Extract fields. Extract fields from the log event. Extracted fields are shown with their example values.

Click **Intelligent Extraction**. Take the following log event as an example.

Enter the log event in the text box.

```
39.149.31.187 - - [12/Mar/2020:12:24:02 +0800] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.132 Safari/537.36" "-"
```

Configure the following Nginx log format in step 2:

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for";'
```

 NOTE

- The **float** data type has seven digit precision.
- If a value contains more than seven valid digits, the extracted content is incorrect, which affects visualization and quick analysis. In this case, you are advised to change the field type to **string**.

Check and edit the fields if needed. For details about rules for configuring extracted fields, see [Setting Structured Fields](#).

- Step 4** Specify a field as the log time. For details, see [Setting Custom Log Time](#).
- Step 5** Click **Save**. The type of extracted fields cannot be changed after the structuring is complete.
- End

Structuring Template

A structuring template extracts fields from either a customized template or a built-in template.

For details, see [Setting a Structuring Template](#).

4.2.3 Setting a Structuring Template

LTS supports two types of structuring templates: system and custom templates.

System Templates

You can choose from multiple system templates, but cannot modify the field types in them or delete the fields. For details, see [Table 4-1](#).

- Step 1** On the **Cloud Structuring Parsing** tab page, select **Structuring Template**.
- Step 2** Click **System template** and select a template. A sample log event is displayed for each template.
- Step 3** View the log parsing results in the **Template Details** table.

NOTE

- If you select a system template for structuring, it uses the custom log time.
- Fields of the string type do not support range query using the >, =, or < operators or the "in" syntax. Use asterisks (*) or question marks (?) for fuzzy query. You need to reconfigure the structuring and change the value of this field to a number.

Table 4-1 System template fields

Template Name	Field Name	Field Type Can Be Changed	Field Can Be Deleted
ELB	Defined by ELB.	No	No
VPC	Defined by VPC.	No	No
CTS	Keys in JSON log events.	No	No
APIG	Defined by APIG.	No	No
DCS audit logs	Defined by DCS.	No	No
TOMCAT	Defined by Tomcat.	No	No
NGINX	Defined by Nginx.	No	No

Template Name	Field Name	Field Type Can Be Changed	Field Can Be Deleted
GAUSSV5 audit logs	Defined by GAUSSV5.	No	No
DDS audit logs	Defined by DDS.	No	No
DDS error logs	Defined by DDS.	No	No
DDS slow query logs	Defined by DDS.	No	No
CFW access control logs	Defined by CFW.	No	No
CFW attack logs	Defined by CFW.	No	No
CFW traffic logs	Defined by CFW.	No	No
MySQL error logs	Defined by MySQL.	No	No
MySQL slow query logs	Defined by MySQL.	No	No
POSTGRESQL slow query logs	Defined by PostgreSQL.	No	No
POSTGRESQL error logs	Defined by PostgreSQL.	No	No
SQLServer error logs	Defined by SQL Server.	No	No
GeminiDB Redis slow query logs	Defined by GeminiDB Redis.	No	No
CDN	Defined by CDN.	No	No
SMN	Defined by SMN.	No	No
GAUSSDB_MY SQL error logs	Defined by GaussDB(for MySQL).	No	No
GaussDB_MySQL slow query logs	Defined by GaussDB(for MySQL).	No	No
ER Enterprise Router	Defined by Enterprise Router.	No	No

Template Name	Field Name	Field Type Can Be Changed	Field Can Be Deleted
MySQL audit logs	Defined by MySQL.	No	No
GeminiDB Cassandra slow query logs	Defined by GeminiDB Cassandra.	No	No
GeminiDB Mongo slow query logs	Defined by GeminiDB Mongo.	No	No
GeminiDB Mongo error logs	Defined by GeminiDB Mongo.	No	No
WAF access logs	Defined by WAF.	No	No
WAF attack logs	Defined by WAF.	No	No
DMS rebalancing logs	Defined by DMS.	No	No
CCE audit logs	Defined by CCE.	No	No
CCE event logs	Defined by CCE.	No	No
CCE NGINX-INGRESS logs	Defined by CCE.	No	-
GeminiDB Redis audit logs	Defined by GeminiDB Redis.	No	No
Influx slow query logs	Defined by Influx.	No	No
Microgateway	Defined by Microgateway.	No	No
GeminiDB Mongo audit logs	Defined by GeminiDB Mongo.	No	No

Step 4 Click **Save**.

----End

Custom Templates

Click **Custom template** and select a template. Custom templates can be obtained in the following ways:

- When you extract fields using methods of regular expression, JSON, delimiter, or Nginx, click **Save as Template** in the lower left corner. In the displayed dialog box, enter a template name and click **OK**. The template will be displayed in the custom template list.
- Create a custom template under the **Structuring Template** option.
 - a. Click **Custom template** and **Create Template**.
 - b. On the displayed page, select **Regular Expressions, JSON, Delimiter, or Nginx**.
 - c. After configuration, enter a template name and click **Save**. The template will be displayed in the custom template list.

4.2.4 Setting Structured and Tag Fields

Setting Structured Fields

You can set extracted fields after cloud structuring. For details, see [Table 4-2](#).

Table 4-2 Rules for configuring structured fields

Template Name	Field Name	Field Type Can Be Changed	Field Can Be Deleted
Regular expressions (auto generate)	User-defined. The name must start with a letter and contain only letters and digits.	Yes	Yes
Regular expressions (manually enter)	<ul style="list-style-type: none"> • User-defined. • Default names such as field1, field2, and field3 will be used. 	Yes	Yes
JSON	Names are set automatically, but you can set aliases for fields.	Yes	Yes
Delimiter	Default names such as field1 , field2 , field3 are used. You can modify these names.	Yes	Yes
Nginx	Names are set based on Nginx configuration, but you can set aliases for fields.	Yes	Yes
Custom templates	User-defined.	Yes	Yes

 NOTE

When you use regular expressions (manually entered), JSON, delimiters, Nginx, or custom templates to structure logs, field names:

- Can contain only letters, digits, hyphens (-), underscores (_), and periods (.).
- Cannot start with a period (.) or underscore (_) or end with a period (.).
- Can contain 1 to 64 characters.

Setting Tag Fields

When you structure logs, you can configure tag fields, so you can use these fields to run SQL queries on the **Visualization** page.

Step 1 In **Step 2 Extract fields**, click the **Tag Fields** tab and **Add Field**.

Step 2 In the **Field** column, enter a name for the tag field, for example, **hostIP**.

 NOTE

If you configure tag fields for a structuring rule that was created before the function of tag fields was brought online, no example values will be shown with the tag fields.

Step 3 To add more fields, click **Add Field**.

Step 4 Click **Save**.

 NOTE

- Tag fields can be the following system fields: **category**, **clusterId**, **clusterName**, **containerName**, **hostIP**, **hostId**, **hostName**, **nameSpace**, **pathFile**, and **podName**.
- Tag fields cannot be the following system fields: **groupName**, **logStream**, **lineNum**, **content**, **logContent**, **logContentSize**, and **collectTime**.
- You can configure both field extraction and tag fields during log structuring.

----End


4.2.5 Setting Custom Log Time

When configuring log ingestion, you can enable **Custom Log Time** to set a time field in the logs as the ingestion configuration time.

Enabling Custom Log Time

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.

Step 2 Expand the target log group and click the name of the target stream.

Step 3 On the log stream details page, click  in the upper right corner. On the page displayed, click the **Cloud Structuring Parsing** tab. For details, see [Setting Cloud Structuring Parsing](#).

Step 4 After cloud structuring parsing is configured, enable **Custom Log Time** and specify the following parameters.

 **NOTE**

A time deviation may occur around the time displayed on the log search page when you enable or disable **Custom Log Time**. Do not frequently enable or disable it.

Table 4-3 Parameter configuration

Parameter	Description	Example
Key	Name of an extracted field. You can select an extracted field from the drop-down list. The field is of the string or long type.	test
Value	Value of an extracted field. After a Key is selected, the Value is automatically filled in. NOTE The value of the field must be within 24 hours earlier or later than the current time.	2022-07-19 12:12:00
Format	For details, see Common Log Time Formats .	yyyy-MM-dd HH:mm:ss
Operation	Click Verify . If the message The time format is successfully matched with the time field value. is displayed, the verification is successful.	-

----End

Common Log Time Formats

[Table 4-4](#) lists common log time formats.

 **NOTE**

By default, log timestamps in LTS are accurate to seconds. You do not need to configure information such as milliseconds and microseconds.

Table 4-4 Time formats

Format	Description	Example
EEE	Abbreviation for Week.	Fri
EEEE	Full name for Week.	Friday
MMM	Abbreviation for Month.	Jan
MMMM	Full name for Month.	January
dd	Number of the day in a month, ranging from 01 to 31 (decimal).	07, 31

Format	Description	Example
HH	Hour, in 24-hour format.	22
hh	Hour, in 12-hour format.	11
MM	Number of the month, ranging from 01 to 12 (decimal).	08
mm	Number of the minute, ranging from 00 to 59 (decimal).	59
a	AM or PM	AM, PM
hh:mm:ss a	Time in the 12-hour format.	11:59:59 AM
HH:mm	Hour and minute format.	23:59
ss	Number of the second, ranging from 00 to 59 (decimal).	59
yy	Year without century, ranging from 00 to 99 (decimal).	04, 98
yyyy	Year (decimal).	2004, 1998
d	Number of the day in a month, ranging from 1 to 31 (decimal).	7, 31
DDD	Number of the day in a year, ranging from 001 to 366 (decimal).	365
u	Number of the day in a week, ranging from 1 to 7 (decimal). The value 1 indicates Monday.	2
w	Number of the week in a year. Sunday is the start of a week. The value ranges from 00 to 53.	23
W	Number of the week in a month, ranging from 0 to 5.	2
U	Number of the day in a week, ranging from 0 to 6 (decimal). The value 0 indicates Sunday.	5
EEE MMM dd HH:mm:ss yyyy	Standard date and time.	Tue Nov 20 14:12:58 2020
EEE MMM dd yyyy	Standard date without time.	Tue Nov 20 2020
HH:mm:ss	Standard time without date.	11:59:59
%s	UNIX Timestamp.	147618725

Examples

Table 4-5 lists common time standards, examples, and expressions.

Table 4-5 Examples

Example	Time Expression	Time Standard
2022-07-14T19:57:36+08:00	yyyy-MM-dd'T'HH:mm:ssXXX	Custom
1548752136	%s	Custom
27/Jan/2022:15:56:44	dd/MMM/yyyy:HH:mm:ss	Custom
2022-08-15 17:53:23+08	yyyy-MM-dd HH:mm:ssX	Custom
2022-08-05T08:24:15.536+0000	yyyy-MM-dd'T'HH:mm:ss.SSSZ	Custom
2022-08-20T10:04:03.204000Z	yyyy-MM-dd'T'HH:mm:ss.SSSZ	Custom
2022-08-22T06:52:08Z	yyyy-MM-dd'T'HH:mm:ssZ	Custom
2022-07-24T10:06:41.000	yyyy-MM-dd'T'HH:mm:ss.SSS	Custom
Monday, 02-Jan-06 15:04:05 MST	EEEE, dd-MMM-yy HH:mm:ss Z	RFC850
Mon, 02 Jan 2006 15:04:05 MST	EEE, dd MMM yyyy HH:mm:ss Z	RFC1123
02 Jan 06 15:04 MST	dd MMM yy HH:mm Z	RFC822
02 Jan 06 15:04 -0700	dd MMM yy HH:mm Z	RFC822Z
2023-01-02T15:04:05Z07:00	yyyy-MM-dd'T'HH:mm:ssZ	RFC3339
2022-12-11 15:05:07	yyyy-MM-dd HH:mm:ss	Custom

4.3 Setting Indexes

An index is a storage structure used to query and analyze logs. Different index settings will generate different query and analysis results. Configure the index settings as required.

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Index Types

LTS supports full-text and field indexes. For details, see [Table 4-6](#).


Table 4-6 Index types

Index Type	Description
Index Whole Text	<p>LTS splits all field values of an entire log into multiple words when this function is enabled.</p> <p>NOTE</p> <ul style="list-style-type: none"> The custom label field uploaded by the user is not included in the full-text index. If you want to search for the custom label field, add the corresponding index field. System reserved fields are not included in full-text indexes. You need to use index fields (Key:Value) to search for them. For details, see System Reserved Fields.
Index Fields	<p>Query logs by specified field names and values (Key:Value).</p> <p>NOTE</p> <ul style="list-style-type: none"> LTS creates index fields for certain system reserved fields by default. For details, see System Reserved Fields. If an index field is configured for a field, the delimiter of the field value is subject to the index field configuration. The quick analysis column in structuring settings has been removed. To use quick analysis, configure index fields and enable quick analysis for the required fields. <p>Here are two examples:</p> <ul style="list-style-type: none"> In the log example, the level and status index fields are configured. The level field is of the string type, the field value is error, and a delimiter is configured. The status field is of the long type, and no delimiter needs to be configured. You can use level:error to search for all logs whose level value is error. In the log example, LTS creates indexes for system reserved fields such as hostName, hostIP, and pathFile by default.

Precautions

- Either whole text indexing or index fields must be configured.
- After the index function is disabled, the storage space of historical indexes is automatically cleared after the data storage period of the current log stream expires.
- LTS creates index fields for certain system reserved fields by default. For details, see [System Reserved Fields](#).
- Different index settings will generate different query and analysis results. Configure the index settings as required. Full-text indexes and index fields do not affect each other.
- After the index configuration is modified, the modification takes effect only for newly written log data.

Configuring Full-Text Indexing

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.
- Step 2** Expand the target log group and click the name of the target stream.
- Step 3** On the log stream details page, click  in the upper right corner. On the page displayed, click the **Index Settings** tab.
- Step 4** Set index parameters by referring to [Table 4-7](#). **Index Whole Text** is enabled by default.

 **NOTE**

- For automatic configuration, LTS obtains the intersection of the raw logs and system fields in the last 15 minutes by default, and combines the intersection with current structured fields and tag fields to form the table data below **Index Fields**.
- If no raw log is generated in the last 15 minutes, LTS obtains **hostIP**, **hostName**, **pathFile**, structured fields, and tag fields to form the table data.
- When structuring is configured for ECS ingestion, the **category**, **hostName**, **hostId**, **hostIP**, **hostIPv6** and **pathFile** fields are automatically added to **Index Fields** on the **Index Settings** page. A field will not be added if the same one already exists.
- When structuring is configured for CCE ingestion, the **category**, **clusterId**, **clusterName**, **nameSpace**, **podName**, **containerName**, **appName**, **hostName**, **hostId**, **hostIP**, **hostIPv6** and **pathFile** fields are automatically added to **Index Fields** on the **Index Settings** page. A field will not be added if the same one already exists.

Table 4-7 Custom full-text indexing parameters

Parameter	Description
Index Whole Text	If Index Whole Text is enabled, a full-text index is created.
Case-Sensitive	Indicates whether letters are case-sensitive during query. <ul style="list-style-type: none"> • If this function is enabled, the query result is case-sensitive. For example, if the example log contains Know, you can query the log only with Know. • If this function is disabled, the query result is case-insensitive. For example, if the example log contains Know, you can also query the log with KNOW or know.

Parameter	Description
Include Chinese	<p>Indicates whether to distinguish between Chinese and English during query.</p> <ul style="list-style-type: none"> • After the function is enabled, if the log contains Chinese characters, the Chinese content is split based on unigram segmentation and the English content is split based on delimiters. <p>NOTE</p> <ul style="list-style-type: none"> - Unigram segmentation is to split a Chinese string into Chinese characters. - The advantage of unigram segmentation is efficient word segmentation of massive logs, and other Chinese segmentation methods have great impact on the write speed. - If Include Chinese is enabled, unigram segmentation is used for Chinese characters (each Chinese character is segmented separately). To obtain more accurate search results, use phrases with the syntax <i>"# phrase to be searched for"</i>. <ul style="list-style-type: none"> • After this function is disabled, all content is split based on delimiters. <p>For example, assume that the log content is: error,400,I Know TodayIsMonday.</p> <ul style="list-style-type: none"> • After this function is disabled, the English content is split based on delimiters. The log is split into error, 400, I, Know, and TodayIsMonday. You can search for the log by error or TodayIsMonday. • After this function is enabled, the background analyzer of LTS splits the log into error, 400, I, Know, Today, Is, and Monday. You can search for the log by error or Today.

Parameter	Description
Delimiters	<p>Splits the log content into multiple words based on the specified delimiters. If the default settings cannot meet your requirements, you can customize delimiters. All ASCII codes can be defined as delimiters.</p> <p>If you leave Delimiters blank, the field value is regarded as a whole. You can search for logs only through a complete string or by fuzzy match.</p> <p>Click Preview to see the effect.</p> <p>For example, assume that the log content is: error,400,I Know TodayIsMonday.</p> <ul style="list-style-type: none"> • If no delimiter is set, the entire log is regarded as a string error,400,I Know TodayIsMonday. You can search for the log only by the complete string error,400,I Know TodayIsMonday or by fuzzy match error,400,I K*. • If the delimiter is set to a comma (,), the raw log is split into: error, 400, and I Know TodayIsMonday. You can search for the log by fuzzy or exact match, for example, error, 400, I Kn*, and TodayIs*. • If the delimiters are set to a comma (,) and space, the raw log is split into: error, 400, I, Know, TodayIsMonday. You can search for the log by fuzzy or exact match, for example, Know, and TodayIs*.
ASCII Delimiters	<p>Click Add ASCII Delimiter and enter the ASCII value by referring to ASCII Table.</p>

Step 5 Click **OK**.

----End

Configuring Index Fields

When creating a field index, you can add a maximum of 500 fields. A maximum of 100 subfields can be added for JSON fields.

Step 1 Specify the number of samples for quick analysis. The value ranges from 100,000 (default) to 10 million. Quick analysis provides a fast overview by sampling field value statistics rather than analyzing all log data. The more logs sampled, the slower the analysis.

Step 2 Click **Add Field** under **Index Fields** to configure field indexing. For details, see [Table 4-8](#).

NOTE

- The indexing parameters take effect only for the current field.
- Index fields that do not exist in log content are invalid.
- For details about system fields, see [System Reserved Fields](#).
- Automatically configuring field indexes: Click **Auto Configure**. LTS generates field indexes based on the first log event in the preview or common system


reserved fields (such as **hostIP**, **hostName**, and **pathFile**). You can add or delete fields as required.

- Configuring field indexes in batches: Select fields and click **Batch Configure**.

Table 4-8 Custom index field parameters

Parameter	Description
Field	<p>Log field name, such as level in the example log.</p> <p>The field name can contain only letters, digits, and underscores (_), and must start with a letter or underscore (_). The field name cannot contain double underscores (__).</p> <p>NOTE</p> <ul style="list-style-type: none"> • Double underscores (__) are used in built-in reserved fields that are not displayed to users in LTS. Avoid using them in custom log field names, as this will prevent the configuration of field index names. • LTS creates index fields for certain system reserved fields by default. For details, see System Reserved Fields. • For system fields, system is displayed next to their names.
Action	<p>Displays the field status, including new, not modified, modified, and deleted. After the index field is changed, click Compare to view the differences between the original and modified configurations.</p> <ul style="list-style-type: none"> • New fields cannot be modified. • When you modify the settings of Type, Case-Sensitive, Common Delimiters, ASCII Delimiters, Include Chinese, or Quick Analysis, the system compares the modified settings with the original settings and changes the action to modified. • After you click OK, the fields whose action is deleted are not saved.
Type	<ul style="list-style-type: none"> • Data type of the log field value. The options are string, long, and float. • Fields of long and float types do not support Case-Sensitivity, Include Chinese and Delimiters.
Case-Sensitive	<p>Indicates whether letters are case-sensitive during query.</p> <ul style="list-style-type: none"> • If this function is enabled, the query result is case-sensitive. For example, if the message field in the example log contains Know, you can query the log only with message:Know. • If this function is disabled, the query result is case-insensitive. For example, if the message field in the example log contains Know, you can also query the log with message:KNOW or message:know.

Parameter	Description
Common Delimiters	<p>Splits the log content into multiple words based on the specified delimiters. If the default settings cannot meet your requirements, you can customize delimiters. All ASCII codes can be defined as delimiters.</p> <p>If you leave Delimiters blank, the field value is regarded as a whole. You can search for logs only through a complete string or by fuzzy match.</p> <p>For example, the content of the message field in the example log is I Know TodayIsMonday.</p> <ul style="list-style-type: none"> • If no delimiter is set, the entire log is regarded as a string I Know TodayIsMonday. You can search for the log only by the complete string message:I Know TodayIsMonday or by fuzzy search message:I Know TodayIs*. • If the delimiter is set to a space, the raw log is split into: I, Know, and TodayIsMonday. You can find the log by fuzzy search or exact words, for example, message:Know, or message: TodayIsMonday.
ASCII Delimiters	<p>Click Add ASCII Delimiter and enter the ASCII value by referring to ASCII Table.</p>

Parameter	Description
Include Chinese	<p>Indicates whether to distinguish between Chinese and English during query.</p> <ul style="list-style-type: none"> After the function is enabled, if the log contains Chinese characters, the Chinese content is split based on unigram segmentation and the English content is split based on delimiters. <p>NOTE</p> <ul style="list-style-type: none"> Unigram segmentation is to split a Chinese string into Chinese characters. The advantage of unigram segmentation is efficient word segmentation of massive logs, and other Chinese segmentation methods have great impact on the write speed. If Include Chinese is enabled, unigram segmentation is used for Chinese characters (each Chinese character is segmented separately). To obtain more accurate search results, use phrases with the syntax <i>#"phrase to be searched for"</i>. <ul style="list-style-type: none"> After this function is disabled, all content is split based on delimiters. <p>For example, the content of the message field in the example log is I Know TodayIsMonday.</p> <ul style="list-style-type: none"> After this function is disabled, the English content is split based on delimiters. The log is split into I, Know, and TodayIsMonday. You can search for the log by message:Know or message:TodayIsMonday. After this function is enabled, the background analyzer of LTS splits the log into I, Know, Today, Is, and Monday. You can search for the log by message:Know or message:Today.
Quick Analysis	<p>By default, this option is enabled, indicating that this field will be sampled and collected. For details, see Creating an LTS Quick Analysis Task.</p> <p>NOTE</p> <ul style="list-style-type: none"> The principle of quick analysis is to collect statistics on 100,000 logs that match the search criteria, not all logs. The maximum length of a field for quick analysis is 2000 bytes. The quick analysis field area displays the first 100 records.
Operation	<p>Click  to delete the target field.</p>

Step 3 Click **OK**.

----End

System Reserved Fields

During log collection, LTS adds information such as the collection time, log type, and host IP address to logs in the form of Key-Value pairs. These fields are system reserved fields.

 **NOTE**

- When using APIs to write log data or add ICAgent configurations, avoid using the same field names as reserved field names to prevent issues such as duplicate field names and incorrect queries.
- A custom log field must not contain double underscores (__) in its name. If it does, indexing cannot be configured for the field.

Table 4-9 System reserved field description

Field	Data Format	Index and Statistics Settings	Description
collectTime	Integer, Unix timestamp (ms)	Index setting: After indexing is enabled, a field index is created for collectTime by default. The index data type is long. Enter collectTime : xxx during the query.	Indicates the time when logs are collected by ICAgent. Example: "collectTime":"1681896081334" indicates 2023-04-19 17:21:21 when converted into standard time.
__time__	Integer, Unix timestamp (ms)	Index setting: After indexing is enabled, a field index is created for time by default. The index data type is long. This field cannot be queried.	Log time refers to the time when a log is displayed on the console. In the example, "__time__":"1681896081334" is 2023-04-19 17:21:21 when converted into standard time. By default, the collection time is used as the log time. You can also customize the log time.

Field	Data Format	Index and Statistics Settings	Description
lineNum	Integer	Index setting: After indexing is enabled, a field index is created for lineNum by default. The index data type is long.	Line number (offset), which is used to sort logs. Non-high-precision logs are generated based on the value of collectTime . The default value is collectTime * 1000000 + 1 . For high-precision logs, the value is the nanosecond value reported by users. Example: "lineNum":"1681896081333991900"
category	String	Index setting: After indexing is enabled, a field index is created for category by default. The index data type is string, and the delimiters are empty. Enter category: xxx during the query.	Log type, indicating the source of the log. Example: The field value of logs collected by ICAgent is LTS , and that of logs reported by a cloud service such as DCS is DCS .
clusterName	String	Index setting: After indexing is enabled, a field index is created for clusterName by default. The index data type is string, and the delimiters are empty. Enter clusterName: xxx during the query.	Cluster name, used in the Kubernetes scenario. Example: "clusterName":"eps test"

Field	Data Format	Index and Statistics Settings	Description
clusterId	String	Index setting: After indexing is enabled, a field index is created for clusterId by default. The index data type is string, and the delimiters are empty. Enter clusterId: xxx during the query.	Cluster ID, used in the Kubernetes scenario. Example: "clusterId":"c7f3f4a5-xxxx-11ed-a4ec-0255ac100b07"
nameSpace	String	Index setting: After indexing is enabled, a field index is created for nameSpace by default. The index data type is string, and the delimiters are empty. Enter nameSpace: xxx during the query.	Namespace used in the Kubernetes scenario. Example: "nameSpace":"monitoring"
appName	String	Index setting: After indexing is enabled, a field index is created for appName by default. The index data type is string, and the delimiters are empty. Enter appName: xxx during the query.	Component name, that is, the workload name in the Kubernetes scenario. Example: "appName":"alertmanager-alertmanager"
serviceID	String	Index setting: After indexing is enabled, a field index is created for serviceID by default. The index data type is string, and the delimiters are empty. Enter serviceID: xxx during the query.	Workload ID in the Kubernetes scenario. Example: "serviceID":"cf5b453xxxad61d4c483b50da3fad5ad"

Field	Data Format	Index and Statistics Settings	Description
podName	String	Index setting: After indexing is enabled, a field index is created for podName by default. The index data type is string, and the delimiters are empty. Enter podName: xxx during the query.	Pod name in the Kubernetes scenario. Example: "podName":"alertmanager-alertmanager-0"
podIp	String	Index setting: After indexing is enabled, a field index is created for podIp by default. The index data type is string, and the delimiters are empty. Enter podIp: xxx during the query.	Pod IP address in the Kubernetes scenario. Example: "podIp":"10.0.0.145"
containerName	String	Index setting: After indexing is enabled, a field index is created for containerName by default. The index data type is string, and the delimiters are empty. Enter containerName: xxx during the query.	Container name used in the Kubernetes scenario. Example: "containerName":"config-reloader"
hostName	String	Index setting: After this function is enabled, a field index is created for hostName by default. The index data type is string, and the delimiters are empty. Enter hostName: xxx during the query.	Indicates the host name where ICAgent resides. Such as "hostName":"epstest-xx518" in the example.

Field	Data Format	Index and Statistics Settings	Description
hostId	String	Index setting: After this function is enabled, a field index is created for hostId by default. The index data type is string, and the delimiters are empty. Enter hostId: xxx during the query.	Indicates the host ID where ICAgent resides. The ID is generated by ICAgent. Such as "hostId":"318c02fe-xxxx-4c91-b5bb-6923513b6c34" in the example.
hostIP	String	Index setting: After this function is enabled, a field index is created for hostIP by default. The index data type is string, and the delimiters are empty. Enter hostIP: xxx during the query.	Host IP address where the log collector resides (applicable to IPv4 scenario) Such as "hostIP":"192.168.0.31" in the example.
hostIPv6	String	Index setting: After this function is enabled, a field index is created for hostIPv6 by default. The index data type is string, and the delimiters are empty. Enter hostIPv6: xxx during the query.	Host IP address where the log collector resides (applicable to IPv6 scenario) Such as "hostIPv6":"" in the example.
pathFile	String	Index setting: After this function is enabled, a field index is created for pathFile by default. The index data type is string, and the delimiters are empty. Enter pathFile: xxx during the query.	File path is the path of the collected log file. Such as "pathFile":"stdout.log" in the example.

Field	Data Format	Index and Statistics Settings	Description
content	String	Index setting: After Index Whole Text is enabled, the delimiters defined in Index Whole Text are used to delimit the value of the content field. The content field cannot be configured in the field index.	Original log content. Example: "content":"level=error ts=2023-04-19T09:21:21.333895559Z"
__receive_time__	Integer, Unix timestamp (ms)	Index setting: After this function is enabled, a field index is created for __receive_time__ by default. The index data type is long.	Time when a log is reported to the server, which is same as the time when the LTS collector receives the log.
_content_parse_fail_	String	Index setting: After indexing is enabled, a field index is created for _content_parse_fail_ by default. The index data type is string, and the default delimiter is used. Enter _content_parse_fail_ : xxx during the query.	Content of the log that fails to be parsed.
__time	Integer, Unix timestamp (ms)	The __time field cannot be configured in the field index.	N/A
logContent	String	The logContent field cannot be configured in the field index.	N/A
logContentSize	Integer	The logContentSize field cannot be configured in the field index.	N/A
logIndexSize	Integer	The logIndexSize field cannot be configured in the field index.	N/A

Field	Data Format	Index and Statistics Settings	Description
groupName	String	The groupName field cannot be configured in the field index.	N/A
logStream	String	The logStream field cannot be configured in the field index.	N/A

ASCII Table

Table 4-10 ASCII table

AS CII Val ue	Character	ASC II Val ue	Character	AS CII Val ue	Character	AS CII Val ue	Character
0	NUL (Null)	32	Space	64	@	96	`
1	SOH (Start of heading)	33	!	65	A	97	a
2	STX (Start of text)	34	"	66	B	98	b
3	ETX (End of text)	35	#	67	C	99	c
4	EOT (End of transmission)	36	\$	68	D	100	d
5	ENQ (Enquiry)	37	%	69	E	101	e
6	ACK (Acknowledg e)	38	&	70	F	102	f
7	BEL (Bell)	39	'	71	G	103	g
8	BS (Backspace)	40	(72	H	104	h
9	HT (Horizontal tab)	41)	73	I	105	i
10	LF (Line feed)	42	*	74	J	106	j

AS CII Value	Character	ASC II Value	Character	AS CII Value	Character	AS CII Value	Character
11	VT (Vertical tab)	43	+	75	K	107	k
12	FF (Form feed)	44	,	76	L	108	l
13	CR (Carriage return)	45	-	77	M	109	m
14	SO (Shift out)	46	.	78	N	110	n
15	SI (Shift in)	47	/	79	O	111	o
16	DLE (Data link escape)	48	0	80	P	112	p
17	DC1 (Device control 1)	49	1	81	Q	113	q
18	DC2 (Device control 2)	50	2	82	R	114	r
19	DC3 (Device control 3)	51	3	83	S	115	s
20	DC4 (Device control 4)	52	4	84	T	116	t
21	NAK (Negative acknowledge)	53	5	85	U	117	u
22	SYN (Synchronous idle)	54	6	86	V	118	v
23	ETB (End of transmission block)	55	7	87	W	119	w
24	CAN (Cancel)	56	8	88	X	120	x
25	EM (End of medium)	57	9	89	Y	121	y
26	SUB (Substitute)	58	:	90	Z	122	z
27	ESC (Escape)	59	;	91	[123	{
28	FS (File separator)	60	<	92	\	124	

AS CII Value	Character	ASC II Value	Character	AS CII Value	Character	AS CII Value	Character
29	GS (Group separator)	61	=	93]	125	}
30	RS (Record separator)	62	>	94	^	126	~
31	US (Unit separator)	63	?	95	_	127	DEL (Delete)

4.4 Searching Logs

4.4.1 Accessing the Log Search Page

After configuring log structuring parsing and indexing, you can enter statements to search for log records that contain specific keywords. You can also search for log data by time range to locate events and issues that occur in a specified period.

Search statements are used to define the filter criteria for log query and obtain the logs that meet the criteria. A search statement may be a keyword, a value, a value range, a space, an asterisk (*), or the like. If it is a space or asterisk (*), no filtering criteria is specified. For more information, see [Using LTS Search Syntax](#).

Searching Logs

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.

Step 2 Click the target log group or stream. The log stream details page is displayed.

Step 3 You can select a time range above the search box to view log data accordingly.

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

NOTE

- **From now:** queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- **From last:** queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified:** queries log data that is generated in a specified time range.

Step 4 Enter search criteria in the search box based on [Using LTS Search Syntax](#) to view, search for, and filter log data.

- In the search area, click the search box, enter a keyword or select a field or keyword from the drop-down list, and click **Search**.

 **NOTE**

- The structured fields are displayed in **key:value** format.
- In the search area, press the up and down arrows on the keyboard to select a keyword or search syntax from the drop-down list, press **Tab** or **Enter** to select a keyword or syntax, and click **Search**.
- Click a field for which quick analysis has been enabled to add it to the search box. For details about how to enable quick analysis, see [Creating an LTS Quick Analysis Task](#).


 **NOTE**

If the field you click already exists in the search box, it will be replaced by this newly added one. If the field is added for the first time, fields in the search box are searched using the AND operator.

Step 5 On the **Log Search** tab page, perform the following operations. For more operations, see [Common Log Search Operations](#).

1. Under **Log Statistics**, view the bar chart showing the log quantity in different time segments. The scale of the log quantity is displayed on the left.
2. In the log content area, hover the cursor over a field and click the log content in blue. You can search for logs by copying, adding to query, and excluding from query.
3. In the log content area, you can select a list or raw log to display its log content.

Step 6 Set the layout of log data, including whether to display fields or display fields in a simple view.









1. Select **Edit layouts** from the drop-down list to access the layout setting page. The list also contains options such as the default layout, pure layout, and default container log layout, for you to set whether to display fields.
 - **Cloud**: This mode is applicable to users who have the write permission. Layout information is stored on the cloud.
 - **Local Cache**: This mode is applicable to users who have only the read permission. Layout information is cached in the local browser.
2. Click  to add a custom layout and set the layout name and visibility of layout fields.
3. After the setting is complete, click **OK**. The new custom layout is displayed in the drop-down list.







----End




Common Log Search Operations

In the log content display area, you can share and download logs, and view context. For details, see [Table 4-11](#).

Table 4-11 Common operations

Operation	Description
Interactive search	Click Interactive Mode in front of the search box. In the displayed Interactive Search dialog box, select fields for index configuration, set the filtering mode, and add associations and groups. After the setting is complete, you can preview the search syntax.
Creating quick search	Click  to create a quick search.
Sharing logs	Click  to copy the link of the current log search page to share the logs that you have searched.
Refreshing logs	<p>You can click  to refresh logs in two modes: manual refresh and automatic refresh.</p> <ul style="list-style-type: none"> • Manual refresh: Select Refresh Now from the drop-down list. • Automatic refresh: Select an interval from the drop-down list to automatically refresh logs. The interval can be 15 seconds, 30 seconds, 1 minute, or 5 minutes.
Copying logs	Click  to copy the log content.
Viewing context of a log	<p>Click  to view the log context.</p> <p>NOTE You can select Simple View to view the log context. You can also download the context.</p>
More operations	<p>Click  to access the log details page of the time segment and view more log information.</p> <ul style="list-style-type: none"> • On the Extended Fields tab page, view field names and values. You can also click buttons in the Operation column to add a field to or exclude a field from a query, set whether a field exists or does not exist, or set whether a field is hidden. • On the JSON Format tab page, view the JSON format of logs. • On the Context Logs tab page, you can set the number of lines to be queried and filtered fields. You can also download logs and enable the summary mode.
Unfold/Fold	<p>Click  to display all the log content. Click  to fold the log content.</p> <p>NOTE Unfold is enabled by default.</p>

Operation	Description
<p>Downloading logs</p>	<p>Click . On the displayed Download Logs page, click Direct Download or Transfer and Download.</p> <ul style="list-style-type: none"> • Direct Download: Download log files to the local PC. Up to 5,000 logs can be downloaded at a time. Select .csv or .txt from the drop-down list and click Download to export logs to the local PC. <p>NOTE</p> <ul style="list-style-type: none"> • If you select Export .csv, logs are exported as a table. • If you select Export .txt, logs are exported as a .txt file. <ul style="list-style-type: none"> • Transfer and Download: Download log files through OBS transfer tasks. Up to 20 million logs can be downloaded at a time. Click Transfer to access the Configure Log Transfer page. For details, see Transferring Logs to OBS.
<p>Hiding/ Expanding all</p>	<p>Click  to set the number of lines displayed in the log content. Click  to hide the log content.</p> <p>NOTE By default, logs are not collapsed, and two lines of logs are shown after collapsing. You can display up to six lines.</p>
<p>JSON</p>	<p>Move the cursor over , click JSON, and set JSON formatting.</p> <p>NOTE Formatting is enabled by default. The default number of expanded levels is 2.</p> <ul style="list-style-type: none"> • Formatting enabled: Set the default number of expanded levels. Maximum value: 10. • Formatting disabled: JSON logs will not be formatted for display.
<p>Collapse configuration</p>	<p>Move the cursor over , click Log Collapse, and set the maximum characters to display in a log.</p> <p>If the number of characters in a log exceeds the maximum, the extra characters will be hidden. Click Expand to view all.</p> <p>NOTE Logs are collapsed by default, with a default character limit of 400.</p>
<p>Log time display</p>	<p>Move the cursor over  and click Log time display. On the page that is displayed, set whether to display milliseconds and whether to display the time zone.</p> <p>NOTE By default, the function of displaying milliseconds is enabled.</p>

Operation	Description
Virtual Scrolling	<p>Move the cursor over  and click Virtual Scrolling. On the page that is displayed, set whether to enable virtual scrolling and enter the buffer size.</p> <p>NOTE</p> <ul style="list-style-type: none"> Virtual scrolling eliminates or minimizes frame and page freezing for better user experience. Data is re-rendered during the process. This may affect smoothness. The buffer size determines the amount of data that can be loaded simultaneously. The larger the buffer, the more data loaded simultaneously, but the worse the scrolling performance.
Invisible fields ()	<p>This list displays the invisible fields configured in the layout settings.</p> <ul style="list-style-type: none"> The  button is unavailable for log streams without layout settings configured. If the log content is CONFIG_FILE and layout settings are not configured, the default invisible fields include appName, clusterId, clusterName, containerName, hostIPv6, NameSpace, podName, and serviceID.

4.4.2 Using LTS Search Syntax

LTS provides a range of search syntax to set search criteria and filtering rules for filtering records that meet the search criteria. Then, you can apply analysis statements on the filtering results for advanced analysis and processing.

To quickly understand and use the search syntax, you are advised to learn [Search Modes](#), [Phrase Search](#), [Operators](#), and [Search Statement Examples](#).

NOTE

- Before using the search syntax, set the delimiters in **Index Settings**. If there is no special requirement, use the default delimiters: `;"=() []{}@&<>/:\\?\n\t\r` and spaces.
- The search syntax does not support search by delimiter.

Search statements do not support delimiters. For example, in the search statement **var/log**, `/` is a delimiter. The search statement is equivalent to **var log** and is used to search for all logs that contain both **var** and **log**. Similarly, the search statements such as **"var:log"** and **var;log** are used to search for all logs that contain both **var** and **log**.

Search Modes

Search statements are used to define the filter criteria for log search and obtain the logs that meet the criteria.

Depending on the index configuration mode, it can be classified into full-text search and field search; according to the search accuracy, it can be classified into exact search and fuzzy search. Other types of search modes include range search and phrase search.

Table 4-12 Search modes

Search Mode	Description	Example
By full text	<p>LTS splits an entire log into multiple keywords when full-text index is set.</p> <p>NOTE</p> <ul style="list-style-type: none"> • content is a built-in field corresponding to the original log text. The search statement GET is equivalent to content:GET. By default, the original log content is matched. • By default, multiple keywords are connected through AND. The search statement GET POST is equivalent to GET and POST. 	<ul style="list-style-type: none"> • GET POST • GET and POST • content:GET and content:POST <p>The preceding search statements have the same function, indicating that logs containing both GET and POST are searched.</p>
By field	<p>Search for specified field names and values (key:value) after field indexing is configured. You can perform multiple types of basic search and combined search based on the data type set in the field index.</p> <p>NOTE</p> <ul style="list-style-type: none"> • The value parameter cannot be empty. You can use the key: "" statement to search for logs with empty field values. • When field search is used together with the not operator, logs that do not contain this field are matched. 	<ul style="list-style-type: none"> • request_time>60 and request_method:po* indicates that the system searches for logs in which the value of request_time is greater than 60 and the value of request_method starts with po. • request_method: "" indicates that logs in which the value of request_method is empty are searched. • not request_method:GET indicates that logs that do not contain the request_method field and whose request_method value is not GET are searched.

Search Mode	Description	Example
By exact match	<p>Use exact words for search. LTS searches with word segmentation, which does not define the sequence of keywords.</p> <p>NOTE If the search statement is abc def, all logs that contain both abc and def are matched, for example, logs abc def and def abc. To ensure the sequence of keywords, use #"abc def".</p>	<ul style="list-style-type: none"> ● GET POST indicates that logs containing both GET and POST are searched. ● request_method:GET indicates that logs in which the value of request_method contains GET are searched. ● #"/var/log" indicates that logs containing phrase /var/log are searched.
By fuzzy match	<p>Specify a keyword in the search statement and add a wildcard, that is, an asterisk (*) or a question mark (?), to the middle or end of the keyword. LTS searches all logs for 100 words that meet the search criteria and returns logs that contain the words. The more precise the specified word is, the more accurate the search results are.</p> <p>NOTE</p> <ul style="list-style-type: none"> ● The asterisk (*) indicates that multiple characters are matched, and the question mark (?) indicates that one character is matched. ● When an asterisk (*) and a question mark (?) are used as delimiters, fuzzy search is not supported since the question mark is a default delimiter. To perform a fuzzy search, remove the question mark from delimiters. ● Words cannot start with an asterisk (*) or a question mark (?). ● Long and float data does not support fuzzy search using asterisks (*) or question marks (?). ● If the fuzzy condition prefix is short and more than 100 words meet the criteria, the search results will be inaccurate. 	<ul style="list-style-type: none"> ● GE* indicates that the system searches for words starting with GE in all logs and returns logs containing these words. ● request_method:GE* indicates that the system searches for request_method values starting with GE in all logs and returns logs containing these words.

Search Mode	Description	Example
By scope	<p>The long and float data supports range search.</p> <ul style="list-style-type: none"> Method 1: Use operators such as = (equal to) > (greater than) < (less than) operators to search for logs. Method 2: Use the in operator to search for logs. The open/closed interval can be modified. <p>NOTE The string fields do not support range query.</p>	<ul style="list-style-type: none"> request_time>=60 indicates that the system searches for logs whose request_time value is greater than or equal to 60. request_time in (60 120] indicates that the system searches for logs whose request_time value is greater than 60 and less than or equal to 120.
By phrase	<p>Phrase search is used to fully match target phrases in logs to ensure the sequence in which keywords appear.</p> <p>NOTE Fuzzy search is not supported for phrase search.</p>	<p>#"abc def" indicates that the system searches all logs for the logs that contain the target phrase abc def.</p>

- Delimiters

LTS splits the log content into multiple words based on delimiters. Default delimiters include `,";=()[]{}@&<>/:\|?\\n\t\r` and spaces.

For example, the default delimiters divide the log `2023-01-01 09:30:00` into four parts: `2023-01-01`, `09`, `30`, and `00`.

In this case, the search statement `2023` cannot match the log. You can search for the log using `2023-01*` or `2023-01-01`.

If the delimiter is set to null, the field value is regarded as a whole. You can search for the corresponding log only through complete log content or fuzzy search.

- Keyword sequence

Only the phrase search `#"abc def"` can ensure the sequence of keywords. In other search modes, multiple keywords are connected by AND.

For example, `request_method:GET POST` is used to query logs that contain both `GET` and `POST`, and the sequence of `GET` and `POST` is not ensured. To ensure their sequence, [Phrase Search](#) is recommended.

- Chinese search

Fuzzy search is not required for Chinese search. Phrase search is recommended to match more accurate results.

In LTS, English content is split into words of different lengths. Therefore, you can use fuzzy search to match logs with English words with the same prefix.

Unigram segmentation is used to a Chinese string into Chinese characters. Each Chinese character is independent, and the length of each part is 1 character.

For example, the search statement **Monday** indicates that logs containing M, o, n, d, a, and y are searched. The search statement **#"Monday"** indicates that logs containing the target phrase **Monday** are searched.

- Invalid keyword

The syntax keywords of log search statements include: && || AND OR and or NOT not in : > < = () []

When **and AND or OR NOT not in** are used as syntax keywords, separate them with a space.

If the log contains syntax keywords and needs to be searched, the search statement must be enclosed in double quotation marks. Otherwise, syntax errors may occur or incorrect results may be found.

For example, if the search statement **content:and** contains the syntax keyword **and**, change it to **content:"and"**.

Phrase Search

Phrase search precisely matches target phrases. For example, the search statement **#"abc def"** searches all logs containing both **abc** and **def** in that specific sequence, with **abc** preceding **def**. For details about the differences between phrase search and keyword search, see [Table 4-13](#).

- Phrase search: It is implemented based on the keyword search syntax. Phrase search can distinguish the sequence of keywords and is used to accurately match target phrases, making the search result more accurate. Phrase search is applicable to English phrases and Chinese phrases, but cannot be used together with fuzzy search.
- Keyword search: Keyword search is implemented based on word segmentation. Delimiters are used to split the search content into multiple keywords for log matching. Keyword search does not distinguish the sequence of keywords. Therefore, as long as a keyword can be matched in a log based on the AND or NOT logic, the log can be found.

Table 4-13 Differences between two search modes

Search Mode	Description	Example
By phrase	Distinguishes the sequence of keywords and is used to accurately match target phrases, making the search result more accurate.	Assume that your log stream contains the following two raw logs: <ul style="list-style-type: none"> • Raw log 1: this service is lts • Raw log 2: lts is service If you search for #"is lts" or #"lts is" , each query matches one log.

Search Mode	Description	Example
By keyword	Does not distinguish the sequence of keywords. The keyword is matched based on the search logic.	<p>Assume that your log stream contains the following two raw logs:</p> <ul style="list-style-type: none"> Raw log 1: this service is lts Raw log 2: lts is service <p>If you search for is lts or lts is, each query matches two logs.</p>

The constraints are as follows:

- Fuzzy search cannot be used together with phrase search.

The asterisk (*) and question mark (?) in phrase search are regarded as common characters. Therefore, phrase search does not support fuzzy search and can be used to search for the asterisk (*) and question mark (?) in logs.
- Phrase search does not support search by delimiter.

For example, in the search statement **#"var/log"**, / is a delimiter. The search statement is equivalent to **#"var log"**, and is used to search for logs containing the target phrase **var log**. Similarly, search statements such as **#"var:log"** and **#"var;log"** are used to search for logs that contain the target phrase **var log**.
- Phrase search is recommended for search in Chinese.

By default, unary word segmentation is used for Chinese characters. Each Chinese character is segmented separately. During the search, logs that contain each Chinese character in the search statement are matched, which is similar to fuzzy search. When more accurate results are required, phrase search is recommended.

Operators

For details about operators supported by the search statements, see [Table 4-14](#).

NOTE

- Except the in operator, other operators are case-insensitive.
- The priorities of operators in descending order are as follows:
 - Colon (:)
 - Double quotation marks (")
 - Parentheses: ()
 - and, not
 - or

Table 4-14 Operators

Operator	Description	Example
and	<p>If there is no syntax keyword between multiple keywords, the and relationship is used by default.</p> <p>NOTE When and is used as an operator, use a space before and after it. For example, 1 and 2 indicates that logs containing both 1 and 2 are searched, and 1and2 indicates that logs containing 1and2 are searched.</p>	GET 200 is equivalent to GET and 200 .
AND	AND operator, equivalent to and .	GET AND 200
&&	<p>AND operator.</p> <p>NOTE When && is used as an operator, spaces are not necessary. For example, 1 && 2 is equivalent to 1&&2, indicating that logs containing both 1 and 2 are searched.</p>	1&&2
or	<p>or operator.</p> <p>NOTE When or is used as an operator, use a space before and after it.</p>	request_method:GET or status:200
OR	OR operator, equivalent to or .	request_method:GET OR status:200
	OR operator. When is used as an operator, spaces are not necessary.	request_method:GET status:200
not	<p>not operator.</p> <p>NOTE</p> <ul style="list-style-type: none"> When not is used as an operator, use a space before and after it. When field search is used together with the not operator, logs that do not contain this field are matched. 	request_method:GET not status:200, not status:200
()	Specifies fields that should be matched with higher priority.	(request_method:GET or request_method:POST) and status:200

Operator	Description	Example
:	Searches for a specified field (key:value). NOTE Use double quotation marks (") to enclose a field name (key) or value that contains reserved characters, such as spaces and colons (:). Examples: <ul style="list-style-type: none"> • "request method":GET • message:"This is a log" • time:"09:00:00" • ipv6:"2024:AC8:2ac::d09" 	request_method:GET
""	Encloses a syntax keyword to convert it into common characters. For example, "and" means searching for logs that contain this word. The word and here is not an operator.	request_method:"GET"
\	Escapes double quotation marks ("). The escaped quotation marks indicate the symbol itself.	To search for instance_id:nginx"01" , use instance_id:nginx\"01\" .
*	An asterisk is a wildcard that matches zero, single, or multiple characters. NOTE Put * in the middle or at the end of a keyword.	request_method:P*T
?	A question mark matches a single character. NOTE Put ? in the middle or at the end of a keyword.	request_method:P?T can match PUT but cannot match POST .
>	Searches logs in which the value of a field is greater than a specified value.	request_time>100
>=	Searches logs in which the value of a field is greater than or equal to a specified value.	request_time>=100
<	Searches logs in which the value of a field is less than a specified value.	request_time<100
<=	Searches logs in which the value of a field is less than or equal to a specified value.	request_time<=100

Operator	Description	Example
=	Searches logs in which the value of a field is equal to a specified value, applying only to float or long fields. For fields of this type, the equal sign (=) and colon (:) have the same function.	request_time=100 is equivalent to request_time:100 .
in	Searches for logs whose field values are in a specified range. Brackets indicate a closed interval, and parentheses indicate an open interval. Numbers are separated with spaces. NOTE Enter in in lowercase. When it is used as an operator, use a space before and after it.	<ul style="list-style-type: none"> • request_time in [100 200] • request_time in (100 200]
#""	Searches for logs that contain the target phrase, ensuring the sequence of keywords. NOTE The asterisk (*) and question mark (?) in phrase search are regarded as common characters. Therefore, phrase search does not support fuzzy search and can be used to search for the asterisk (*) and question mark (?) in logs.	request_method:#"GET POST"

Search Statement Examples

For the same search statement, different search results are displayed for different log content and index configurations. This section describes search statement examples based on the following log sample and indexes.

Figure 4-1 Log sample

```

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
content: {
  request_method: POST
  request_uri: /authui/login
  request_time: 56
  request_length: 3718
  status: 200
  x-language: zh-cn
  date: Mon, 17 Apr 2023 00:33:48 GMT
  content-type: application/json
  content-encoding: gzip
  scheme: https
  sec-ch-ua-mobile: ?0
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
  week: 20230417
}
content-encoding: gzip
content-type: application/json
date: Mon, 17 Apr 2023 00:33:48 GMT
request_length: 3718
request_method: POST
request_time: 56
request_uri: /authui/login
scheme: https
sec-ch-ua-mobile: ?0
status: 200
week: 20230417
x-language: zh-cn
    
```

Table 4-15 Search statement examples

Search Requirement	Search Statement
Logs of POST requests whose status code is 200	request_method:POST and status=200
Logs of successful GET or POST requests (status codes 200 to 299)	(request_method:POST or request_method:GET) and status in [200 299]
Logs of failed GET or POST requests	(request_method:POST or request_method:GET) not status in [200 299]
Logs of non-GET requests	not request_method:GET
Logs of successful GET request and request time is less than 60 seconds	request_method:GET and status in [200 299] not request_time>=60
Logs whose request time is 60 seconds.	<ul style="list-style-type: none"> request_time:60 request_time=60
Logs of requests whose time is greater than or equal to 60 seconds and less than 200 seconds	<ul style="list-style-type: none"> request_time>=60 and request_time<200 request_time in [60 200)
Logs that contain and	content:"and" NOTE Double quotation marks are used to enclose and . In this case, and is a common string, not an operator.

Search Requirement	Search Statement
Logs that do not contain the user field.	not user:*
Logs in which the value of user is empty are searched.	user:'''
Logs in which the value of the week field is not Monday	not week: Monday
Logs in which the value of sec-ch-ua-mobile is ?0 are searched.	sec-ch-ua-mobile:#"?0" NOTE If search is required when log content contains asterisks (*) or question marks (?), use phrases search.

For more complex search examples, see [Table 4-16](#).

Table 4-16 Fuzzy search

Search Requirement	Search Statement
Logs that contain words starting with GE	GE*
Logs that contain words starting with GE and with only one character after GE.	GE?
Logs in which the value of request_method contains a word starting with G.	request_method:G*
Logs in which the value of request_method starts with P, ends with T, and contains a single character in the middle.	request_method:P?T
Logs in which the value of request_method starts with P, ends with T, and contains zero, single, or multiple characters in the middle.	request_method:P*T

Search based on delimiters. For example, the value of the **User-Agent** field is **Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36**.

- If this parameter is left blank, the value of this field is considered as a whole. In this case, when you use **User-Agent:Chrome** to search for logs, no log can be found.
- When the delimiter is set to , "" ; = () [] { } ? @ & < > / : \ n \ t \ r , the value of this field is split into **Mozilla, 5.0, Windows, NT, 10.0, Win64, x64, AppleWebKit, 537.36, KHTML, like, Gecko, Chrome, 113.0.0.0, Safari, and 537.36**.

Then you can use search statements such as **User-Agent:Chrome** for search.

Table 4-17 Delimiter-based search

Search Requirement	Search Statement
Logs in which the value of User-Agent contains Chrome	User-Agent:Chrome
Logs in which the value of User-Agent contains the word starting with Win	User-Agent:Win*
Logs in which the value of User-Agent contains Chrome and Linux	User-Agent:"Chrome Linux"
Logs in which the value of User-Agent contains Firefox or Chrome	User-Agent:Chrome OR User-Agent:Linux
Logs in which the value of User-Agent contains Chrome but not Linux	User-Agent:Chrome NOT User-Agent:Linux

4.4.3 Creating an LTS Quick Analysis Task

Logs contain information such as system performance and business status. For example, the frequency of keyword **ERROR** indicates the system health, and the frequency of keyword **BUY** indicates the business activity. You can use the quick analysis function to query specified log keywords. LTS collects statistics on the keywords and generates metric data, so that you can learn about the system performance and business in real time.

 **NOTE**

- Supports analysis on first 100,000 logs.
The purpose of quick analysis is to quickly return the distribution and change trend of field values. It does not analyze all data, but only samples.
- Logs can be filtered by query time and criteria for analysis.
Quick analysis is to analyze the logs queried by query statements. When the number of queried logs is 0, no result is displayed for quick analysis.
- Quick analysis can be used to generate query statements.
You can click an analysis result to automatically generate a query statement, query logs, and generate a new quick analysis.
- The maximum length of a field for quick analysis is 2 KB.
- The distribution statistics in quick analysis field area displays the first 100 records.
- If quick analysis is not enabled within the analysis time range, a field does not exist, or a field value is null, the analysis result of the field is **null**.
 - When you click **null** to add a string field to the search box, *Field: "null"OR NOT Field: ** will be displayed.
 - When you click **null** to add a float or long field to the search box, **NOT Field: *** will be displayed.
 - If quick analysis is not enabled, the column-store data used for analysis is not stored, and the analysis result is **null**. In this case, log search is meaningless and no log may be matched.

Prerequisites

Quick analysis is conducted on fields extracted from structured logs. **Structure** raw logs before you create a quick analysis task.


Creating a Quick Analysis Task

Quick analysis is performed on a per-log-stream basis. You can create a quick analysis task as follows:

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.




Step 2 Click the target log group or stream. The log stream details page is displayed.

Step 3 You can create a quick analysis task in the following ways:

1. Click  to go to the setting details page. Under **Index Fields**, enable **Quick Analysis** when adding a field.
2. On the **Cloud Structuring Parsing** tab page, enable **Auto Configuration and Analysis**. It is enabled by default. This enables structured fields for configuring indexes and quick analysis.

Step 4 Click **OK**. The quick analysis task is created.

NOTE

-  indicates a field of the **string** type.
-  indicates a field of the **float** type.
-  indicates a field of the **long** type.
- The maximum length of a field for quick analysis is 2000 bytes.
- The quick analysis field area displays the first 100 records.

----End


4.4.4 Saving Conditions for Quick Search

If you need to repeatedly use a keyword to search for logs, you can set and save the keyword as a quick query statement for faster log querying and analysis.

Saving Conditions for Quick Search

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.

Step 2 Click the target log group or stream. The log stream details page is displayed.

Step 3 Click  on the **Log Search** tab page. Enter a name and statement for quick search. By default, quick search and quick search (local cache) are enabled.

- A quick search name is used to distinguish multiple quick search statements. The name can be customized and must meet the following requirements:

- Can contain only letters, digits, hyphens (-), underscores (_), and periods (.).
- Cannot start with a period (.) or underscore (_) or end with a period (.).
- Can contain 1 to 64 characters.
- A quick search statement is used to repeatedly search for logs, for example, **error***.


Step 4 Click **OK**. On the **Quick Search** tab page in the left navigation pane, you can view the statements that are successfully saved.

Click the name of a quick search statement to view log details.

----End

Viewing Context of a Log

You can check the logs generated before and after a log for quick fault locating.

Step 1 On the **Log Search** tab page of the log details page, click  to view the context. Details of several logs generated before and after the log are displayed.

Step 2 On the displayed page, check the log context. For details, see [Table 4-18](#).

Table 4-18 Introduction to log context viewing

Feature	Description
Search Rows	Select the number of lines of logs to be queried as required.
Highlighting	Enter a string to be highlighted and press Enter .
Filter	Enter a string to be filtered and press Enter . When both Highlighting and Filter are configured, the filtered string can also be highlighted.
Fields	The default field for viewing log context is content . Click Fields to view the context of other fields.
Prev	View half the number of Search Rows leading to the current position. For example, if Search Rows is set to 100 and you click Prev , 50 rows prior to the current position are displayed. In this case, the current line number is -50 . If you click Prev again, the line number will become -100 , -150 , -200 , and so on.
Current	Current log position. When Prev or Update is set, you can click Current to return to the position where the context starts (when the line number is 0).

Feature	Description
Update	View half the number of Search Rows following the current position. For example, if Search Rows is set to 100 and you click Update , 50 rows following the current position are displayed. In this case, the current line number is 50. If you click Update again, the line number will become 100, 150, 200 , and so on.
Summary Mode	<ul style="list-style-type: none"> • If this mode is enabled, only the line number and content are displayed. • If this mode is disabled, log details are displayed.
Download	Only content in the content field can be downloaded to the local PC.

----End

4.5 Viewing Real-Time Logs

Logs are reported to LTS about every minute, allowing you to view them on the **Real-Time Logs** page within 1 minute after configuring log ingestion. This enables rapid search and analysis.

NOTE

Log data is usually loaded every 5 seconds. However, if no data is generated in a 5-second interval, no new data will be displayed. Log data will be updated in the next 5 seconds if there is new data coming in that interval.

Prerequisites

- You have created log groups and log streams.
- You have performed operations provided in [Installing ICAgent \(Intra-Region Hosts\)](#).
- You have configured log collection rules.

Viewing Real-Time Logs

Stay on the **Real-Time Logs** tab page to keep updating them in real time. If you leave the **Real-Time Logs** tab page, logs will stop being loaded. The next time you access the tab page, the logs that were shown before you left the tab page will not be displayed.

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.
- Step 2** Click the target log group or stream. The log stream details page is displayed.
- Step 3** Click the **Real-Time Logs** tab to view the real-time logs.

 NOTE

Filter host and K8s logs by source.

- If **Source** is set to **Host**, set the host IP address and file path.
- If **Source** is set to **K8s**, set the instance name, container name, and file path.

Logs are reported to LTS once every minute. You may wait for at most 1 minute before the logs are displayed.

In addition, you can customize log display by clicking **Clear** or **Pause** in the upper right corner.

- **Filter**: Obtain data from the index configuration, structuring configuration, and latest logs.
- **Clear**: Displayed logs will be cleared from the real-time view.
- **Pause**: Loading of new logs to the real-time view will be paused.

After you click **Pause**, the button changes to **Continue**. You can click **Continue** to resume the log loading to the real-time view.

----End

4.6 Analyzing Logs in LTS

After you structure logs, wait 1 to 2 minutes and then you can query and analyze the structured logs using SQL statements and visualize the query results.

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Prerequisites

- Logs have been collected.
- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

 NOTE

If a structured field shares a name with one of the reserved fields for SQL such as **time**, **select**, and **where**, or its name contains hyphens (-), underscores (_), and periods (.), you need to double-quote the field during SQL query.

Analyzing Logs

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**. The **Log Management** page is displayed by default.
- Step 2** Click the target log group or stream. The log stream details page is displayed.
- Step 3** Click the **Log Analysis** tab.
- Step 4** Select a time range, enter a SQL statement by referring to [SQL Analysis Syntax](#), and click **Search**. The search results will be displayed in various charts in the lower part.

There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

- **From now:** queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
- **From last:** queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
- **Specified:** queries log data that is generated in a specified time range.

NOTE

- SQL query constraints are as follows:
 1. A maximum of 100,000 records can be returned for each query.
 2. If there are more than 100,000 aggregation results, they may be inaccurate.
- There are some constraints when you use a string in a WHERE clause.
 1. The value should be enclosed by single quotation marks (') for exact match, and by single or double quotation marks (") for fuzzy search. If the key shares a name with one of the SQL reserved fields, enclose the key with double quotation marks (").
 2. Recommended formats: WHERE "Key"= 'Value' and WHERE "Key" like ' %Value%'
- There are no constraints on **float** and **long** types in WHERE clauses. However, you are still advised to use the formats described above to avoid query exceptions caused by keyword conflicts.

Step 5 If the number of logs generated within the specified time range exceeds 1 billion, iterative query is triggered so you can view all logs in multiple queries. The message **Query status: Results are accurate** is displayed.

Query status: Results are accurate.

Step 6 Select a chart to present the query result. For details, see [Visualizing Logs in Statistical Charts](#).


Step 7 You can perform the following operations on the query result:

- Click **Create**. In the displayed **Create Chart** dialog box, set **Chart Name** and enable **Add to Dashboard** as required, and click **OK** to save the visual chart.
- Click **Save**. In the displayed **Save Chart** dialog box, set **Chart Name** and enable **Add to Dashboard** as required, and click **OK** to save the visual chart. Select a chart and click **Save**, to modify the chart.
- Click **Save As**. In the displayed dialog box, set **Chart Name** and enable **Add to Dashboard** as required, and click **OK** to copy the existing visual chart.

NOTE

You must save a chart before saving it as a visual chart.

- Click **Download** to download the visual data of the current SQL query result. The file is in .csv format.

- Click  . In the displayed **Create Alarm Rule** dialog box, configure **SQL alarm rules** for the selected visual chart.

 **NOTE**

You can create an alarm rule only after saving the chart.

- Click **Show Chart** to expand the visual charts of the current log stream. Click **Show Chart** again to collapse the visual charts of the current log stream.

----End

4.7 SQL Analysis Syntax

4.7.1 Overview

Structured Query Language (SQL) is a programming language used to control database access and manage data in databases. LTS SQL provides statements for querying structured data in log streams. In this document, SQL refers to LTS SQL.

SQL consists of commands and functions that are used to manage databases and database objects. When using this language, comply with the rules for using expressions and texts. In addition to SQL syntax reference, this document also provides information about expressions, functions, and operators. Basic SQL query statements are as follows.

Syntax

```
SELECT [ ALL | DISTINCT ] { * | expr }  
FROM { <subquery> }  
[ WHERE where_condition ]  
[ GROUP BY [ col_name_list ]  
[ HAVING expr ]  
[ ORDER BY expr [ ASC | DESC ], expr [ ASC | DESC ], ... ]  
[ LIMIT limit ]  
[ OFFSET offset ]
```

Data Types

Table 4-19 lists the data types supported by SQL query. The data types of fields can be converted as needed. After the data type of a field is converted, the default value is displayed. For example, after a field of the **string** type is converted to the **long** type, the default value **0** of the **long** type is displayed. Similarly, when a null value is converted to a non-null value, the default value is used. For example, when a null value of the **string** type is converted to a numeric value, the default value **0** is returned.

In the SQL syntax, characters must be enclosed in single quotation marks ('). Fields or table names are either not enclosed or are enclosed in double quotation marks ("). For example, **'msg'** indicates the string **msg**, while **msg** or **"msg"** indicates the structured **msg** field.

Table 4-19 Data types supported in SQL queries

Native Data Type	Default Value	Description
STRING	""	Native string type
FLOAT	0.0	Native float type
LONG	0	Native long type

Query Statements

Table 4-20 SQL query statements

Statement	Description	Example
DISTINCT	Only distinct values are returned.	SELECT DISTINCT visitCount
FROM	Indicates the source data set of the queried data. It can be the structured data of the current log stream or a subset of this data. If FROM is not specified, the structured data of the current log stream is queried by default. If the data source to be queried is a subset, you need to compile a subquery statement.	SELECT visitCount
WHERE	Specifies the filter criteria. Arithmetic operators, relational operators, and logical operators are supported. You can enter the filtering condition in where_condition .	SELECT visitCount WHERE visitCount > 0
GROUP BY	Specifies the grouping field. Single-field grouping and multi-field grouping are supported. You can enter the structured field list in col_name_list .	SELECT host, count(*) AS pv WHERE visitCount > 0 GROUP BY host
HAVING	Used only with GROUP BY. This statement specifies the structured field used to filter the GROUP BY results.	SELECT host, count(*) AS pv GROUP BY host HAVING pv > 10

Statement	Description	Example
ORDER BY	Fields that follow must be used for GROUP BY. The query results of GROUP BY can be sorted by any structured field.	SELECT host, count(*) AS pv GROUP BY host ORDER BY pv
ASC/DESC	ASC (default) sorts from the lowest value to the highest value. DESC sorts from the highest value to the lowest value.	SELECT host, count(*) AS pv GROUP BY host ORDER BY pv DESC
LIMIT	Limits the number of structured logs returned in the query result. A maximum of 100,000 structured logs can be returned for each query. If the LIMIT statement is not used, the latest 100 records in the query result are returned by default.	SELECT host LIMIT 100

Examples

Table 4-21 Examples of common SQL query statements

Query Requirement	Query Statement
Standard query	SELECT "field" WHERE "field" = 'value'
Number of rows	SELECT count(*)
Column alias	SELECT count(*) AS "pv"
Deduplication	SELECT DISTINCT("field")
Pagination	SELECT "field" LIMIT 100
Sorting	SELECT "__time" ORDER BY "__time"
Grouping	SELECT "field" GROUP BY "field"
Statistics by group	SELECT "field",count(*) GROUP BY "field"
Fuzzy search	SELECT "field" LIKE 'value%'
Sum	SELECT sum("field")
Maximum value	SELECT max("field")
Minimum value	SELECT min("field")
Average value	SELECT avg("field")

Query Requirement	Query Statement
SQL nested subquery	SELECT sum(pv) FROM (SELECT "field",count(*) AS "pv" GROUP BY "field")
HAVING clause filtering	SELECT "field",count(*) AS "pv" GROUP BY "field" HAVING "pv" > 10
Query containing GET and POST requests	SELECT * WHERE "request_method" IN ('GET', 'POST')
Query without GET and POST requests	SELECT * WHERE "request_method" NOT IN ('GET', 'POST')
Logs of non-GET requests	SELECT * WHERE "request_method" != 'GET'
Logs of successful GET request with the 200 status code and request time is less than 60 seconds	SELECT * WHERE "request_method" = 'GET' AND "request_time" < 60
Logs of requests whose time is greater than or equal to 60 seconds and less than 200 seconds	SELECT * WHERE "request_time" >=60 and "request_time" < 200
Logs of GET or POST requests	SELECT * WHERE "request_method" = 'GET' OR "request_method" = 'POST'

The following reference statements contain all basic query syntax and are constructed based on structured logs of Elastic Load Balance (ELB).

```
SELECT url AS Url, host AS Host, failure_rate AS FailureRate,
CONCAT(CAST(access_count AS varchar), ' times') AS "All",
CONCAT(CAST(rsp_200_count AS varchar), ' times') AS "COUNT_200"
FROM ( SELECT
CONCAT(host, CASE WHEN STRPOS(router_request_uri, '?') = 0 THEN router_request_uri ELSE
SUBSTR(router_request_uri, 1, 1) END) AS url,
host,count(1) AS access_count,
SUM(CASE WHEN status = 200 THEN 1 ELSE 0 END) AS "rsp_200_count",
(CASE WHEN COUNT(1) < 30 THEN 0 ELSE round(SUM(CASE WHEN status >= 400 THEN 1 ELSE 0 END) *
100.0 / COUNT(1), 2) END) AS failure_rate
WHERE host NOT IN ('monitor-new.olayc.cn')
GROUP BY host,router_request_uri
HAVING router_request_uri NOT IN ('/robots.txt', '/null', '/undefined')
)
ORDER BY FailureRate DESC
LIMIT 100
```

4.7.2 SQL Aggregate Functions

Aggregate functions collect statistics on specified columns of structured logs and return a single value. They are often used with the SELECT and GROUP BY clauses. [Table 4-22](#) lists the aggregate functions supported by LTS.

Pay attention to the following points when using aggregate functions:

- Aggregate functions can be used in the SELECT clause of any query. You can use the syntax such as AGG(expr) FILTER(WHERE whereExpr) to filter columns before aggregation. The aggregate function aggregates only the columns that meet the filter criteria.
- In the same SQL query statement, the result of the corresponding aggregate function varies according to the filter criteria.
- Only COUNT function can be used with DISTINCT function.
- Aggregation operations are not performed in a fixed order. When an SQL statement that contains multiple aggregate functions is executed for query, and the order for executing the functions affects the operation result, the results obtained from different queries may be inconsistent.
- If the data to be aggregated is of the **float** type, the aggregation results may be inconsistent if you perform the same query for multiple times. If you want to obtain the same result from executing the same query, you are advised to use the ROUND function.

Syntax

```
SELECT COUNT(fieldname1)
```

Aggregate Function Statements

Table 4-22 Aggregate function statements

Statement	Description	Example
COUNT(*)	Counts the number of rows.	SELECT COUNT(*)
COUNT(DISTINCT expr)	Counts the number of rows of a field after deduplication. The field value can be a character string or number. The returned value is an estimated value (with a default 2.3% standard error).	SELECT COUNT(DISTINCT host)
SUM(expr)	Returns the sum of the numbers.	SELECT SUM(visitCount)
MIN(expr)	Returns the minimum number.	SELECT MIN(visitCount)
MAX(expr)	Returns the maximum number.	SELECT MAX(visitCount)
AVG(expr)	Returns the average value.	SELECT AVG(visitCount)

Statement	Description	Example
EARLIEST(expr)	The expression must be of the numeric type. The earliest value of expr , which is the first queried value, is returned.	SELECT EARLIEST(visitCount)
LATEST(expr)	The expression must be of the numeric type. The latest value of expr , which is the last queried value, is returned.	SELECT LATEST(visitCount)
APPROX_QUANTILE_DS(expr, probability)	Calculates the approximate quantile of expr . The value of probability must be between 0 and 1.	APPROX_QUANTILE_DS(expr, probability)

4.7.3 SQL Period-over-Period Functions

This section describes the basic syntax and examples of period-over-period functions.

compare Function

This function is used to compare the calculation result in the current period with that in a previous period (*n* seconds ago).

Syntax

- This syntax is used to compare the calculation result in the current period with that *n* seconds ago.
compare(x,n)
- This syntax is used to compare the calculation result in the current period with that *n1*, *n2*, and *n3* seconds ago.
compare(x, n1, n2, n3...)

Description

Table 4-23 Parameters of the year-on-year function

Parameter	Description
x	Name of the target column. The value is of the double or long type.
n	Time window (in seconds). For example, 3600 means 1 hour, 86400 means 1 day, 604800 means 1 week, and 31622400 means 1 year.

Returned Data Type

JSON array. The format is [*Current calculation result, Calculation result n seconds ago, Ratio of the current calculation result to that n seconds ago*].

Example

This example calculates the ratio of the number of visits in the current hour to that in the same period of the previous day.

1. Set the query and analysis time range to 1 hour (full hour), and run the following query and analysis statements: In the preceding command, **86400** indicates the current time minus **86400** seconds (one day).

```
SELECT
compare(PV, 86400)
FROM (SELECT count(*) AS PV )
```

2. Check the query and analyze results.
 - **5994.0** indicates the number of website visits in the current hour (for example, from 2021-01-02 00:00:00 to 2021-01-02 01:00:00).
 - **6000.0** indicates the number of website visits in the same period of the previous day, for example, from 2021-01-01 00:00:00 to 2021-01-01 01:00:00.
 - **0.999** indicates the ratio of the number of website visits in the current hour to that in the same period of the previous day.

Figure 4-2 Query and analysis results

```
EXPRS0
[5994.0,6000.0,0.999]
```

3. Display query and analysis results in columns.

```
SELECT
diff[1] as "today",
diff[2] as "yesterday",
diff[3] as "ratio"
FROM(SELECT compare(pv, 86400) AS diff FROM (SELECT count(*) AS pv ))
```

Figure 4-3 Query and analysis results

today	yesterday	ratio
5993	6029	0.99402887

ts_compare Function

This function is used to compare the calculation result in the current period with that *n* seconds ago.

The `ts_compare` function must be grouped by time column (GROUP BY).

Syntax

- This function is used to compare the calculation result in the current period with that *n* seconds ago.

```
ts_compare(x, n)
```

- This function is used to compare the calculation result in the current period with that *n1*, *n2*, and *n3* seconds ago.

```
ts_compare(x, n1, n2, n3...)
```

Description

Table 4-24 Parameters of the period-over-period function

Parameter	Description
x	The value is of the double or long type.
n	Time window (in seconds). For example, 3600 means 1 hour, 86400 means 1 day, 604800 means 1 week, and 31622400 means 1 year.

Returned Data Type

JSON array. The format is [*Current calculation result, Calculation result n seconds ago, Ratio of the current calculation result to that n seconds ago, UNIX timestamp n seconds ago*].

Example

This example compares the number of website visits in the 3 hours of today with that in the 3 hours of the previous day.

Set the query and analysis time range to 3 hours of the current day and run the following statements: In the preceding command, **86400** indicates that the current time minus **86400** seconds (one day), and **date_trunc('hour', __time)** indicates that the **date_trunc** function is used to truncate the input timestamp to the hour.

- Query and analysis statements

```
SELECT
  t_time,
  ts_compare(PV, 86400) AS data
FROM(
  SELECT
    date_trunc('hour', __time) AS t_time,
    count(*) AS PV
  GROUP BY
    t_time
  ORDER BY
    t_time
)
GROUP BY
  t_time
```

- Query and analysis results

t_time	data
2021-10-26T06:00:00.000Z	[159.0,224.0,0.7098214285714286,1.6351416E9]
2021-10-26T07:00:00.000Z	[100.0,148.0,0.6756756756756757,1.6351452E9]
2021-10-26T08:00:00.000Z	[100.0,100.0,1.0, 1.6016544E9, 1.6351488E9]

4.7.4 SQL JSON Functions

Description

JSON functions are used to parse JSON objects or JSON arrays and extract values from them.

Syntax

```
SELECT json_extract(Results, '$.[0].EndTime')
```

JSON Function Statements

Table 4-25 JSON function statements

Statement	Description	Example	Returned Value Type
json_extract	Extracts a set of JSON values (object or array) from a JSON object or JSON array.	json_extract(x, json_path)	Data of string type in JSON format
json_extract_scalar	Extracts a set of scalar values (string, integer, or Boolean) from a JSON object or JSON array. If the specified JSON path is not a scalar, null is returned.	json_extract_scalar(x,json_path)	varchar type

Examples

- **json_extract function**

Obtain the value of the **EndTime** field in the **Results** field.

- a. Example field

```
Results:[{"EndTime":1626314520},{"FireResult":2}]
```

- b. Query and analysis statement

```
SELECT json_extract(Results, '$.[0].EndTime')
```

- c. Query and analysis result

Table 4-26 Query and analysis results

EXPR\$0
1626314520

- **json_extract_scalar function**

Obtain the values of the **RawResultCount** field from the **Results** field, convert these values to **bigint** type, and sum them up.

- a. Example field

```
Results:[{"EndTime":1626314520},{"RawResultCount":1}]
```


- b. Query and analysis statement
`SELECT sum(cast(json_extract_scalar(Results,'$.[1].RawResultCount') AS bigint))`
- c. Query and analysis result

Table 4-27 Query and analysis result

EXPR\$0
1546

4.7.5 SQL IP Functions

Constraints

The relationships between IP addresses and regions provided by LTS are from third-party libraries. The data is updated periodically (about half a year) and may not reflect the correct relationships. LTS will be optimized to update the libraries more frequently to provide better experience.

A maximum of 5 million data records can be queried using the aggregate query of an IP function. Exceeding this limit will cause the query to time out.

Description

IP functions identify the country, state/province, city, and network carrier to which a target IP address belongs.

Syntax

```
SELECT count(*) AS PV, ip_to_province(client_ip) AS province GROUP BY province
```

IP Function Statements

Table 4-28 IP function statements

Statement	Description	Example
ip_to_province	Identifies the state/province where an IP address is located.	ip_to_province(x)
ip_to_country	Identifies the country/region where an IP address is located.	ip_to_country(x)
ip_to_city	Identifies the city where an IP address is located.	ip_to_city(x)
ip_to_provider	Identifies the network carrier of an IP address.	ip_to_provider(x)

Statement	Description	Example
ip_to_geo	Returns the longitude and latitude of an IP address.	ip_to_geo(x)

Examples

- **ip_to_province**

Obtains the top 3 states/provinces with the most requests.

- a. Query and analysis statement

```
SELECT count(*) AS PV, ip_to_province(client_ip) AS province GROUP BY province ORDER BY PV desc LIMIT 3
```

- b. Query and analysis result

Table 4-29 Query and analysis result

PV	province
101	Guangdong
83	Shanghai
78	Shandong

- **ip_to_country**

Obtains the top 3 countries/regions with the most requests.

- a. Query and analysis statement

```
SELECT count(*) AS PV, ip_to_country(client_ip) AS country GROUP BY country ORDER BY PV desc LIMIT 3
```

- b. Query and analysis result

Table 4-30 Query and analysis result

PV	country
100	China
76	United States
55	Canada

- **ip_to_city**

Obtains the top 3 cities with the most requests.

- a. Query and analysis statement

```
SELECT count(*) AS PV, ip_to_city(client_ip) AS city GROUP BY city ORDER BY PV desc LIMIT 3
```

- b. Query and analysis result

Table 4-31 Query and analysis result

PV	city
109	Guangzhou
89	Shanghai
23	Xi'an

- **ip_to_provider**

Obtains the top 3 carriers with the most requests.

- a. Query and analysis statement

```
SELECT count(*) AS PV, ip_to_provider(client_ip) AS provider GROUP BY provider ORDER BY PV desc LIMIT 3
```

- b. Query and analysis result

Table 4-32 Query and analysis result

PV	provider
115	Telecom
65	att.com
44	Unicom

- **ip_to_geo**

Returns the longitude and latitude of an IP address.

- a. Query and analysis statement

```
SELECT count(*) AS PV, ip_to_geo (client_ip) AS geo GROUP BY geo ORDER BY PV desc LIMIT 3
```

- b. Query and analysis result

Table 4-33 Query and analysis result

PV	geo
101	*, *
83	47.369013, -68.326674
78	32.715891, -117.161588

4.7.6 SQL Mathematical Functions

Description

Mathematical functions are one type of scalar functions. They support only numeric fields and can be used to round values, obtain absolute values, and perform modulo operations. For details, see [Table 4-34](#).

In mathematical operations, if the operands involved in an expression are all integers, SQL uses integer operations. Otherwise, SQL switches to floating-point operations. You can force the switch by converting one of the operands to **float** type, and the runtime SQL extends the 32-bit floats in most expressions to 64-bit floats.

Syntax

```
SELECT ABS(fieldname1) AS fieldname1_abs
```

Mathematical Function Statements

Table 4-34 Mathematical function statements

Statement	Description	Example
ABS(expr)	Returns the absolute value.	SELECT ABS(fieldname1)
CEIL(expr)	Rounds up to the nearest integer.	SELECT CEIL(fieldname1)
FLOOR(expr)	Rounds down to the nearest integer.	SELECT FLOOR(fieldname1)
TRUNCATE(expr, digits)	Truncates expr to a specified number of digits . If the number is negative, many positions to the left of the decimal point are truncated. If not specified, the number is zero by default.	SELECT TRUNCATE(fieldname1, 2)

Statement	Description	Example
ROUND(expr, digits)	Rounds <i>expr</i> to a specified number of decimal places. The result is of the same type as that of <i>expr</i> . <i>expr</i> can be an integer or a floating point number, but <i>digits</i> must be an integer. If digits is not specified, the default value 0 is used. If digits is a negative number, the value of expr is rounded off. If expr is a non-numeric value, it will be converted to 0. If expr is an infinite number, it is converted to a number with a finite number of digits of the closest double type.	SELECT ROUND(fieldname1, 2)
x + y	Addition.	SELECT fieldname1 + fieldname2
x - y	Subtraction.	SELECT fieldname1 - fieldname2
x * y	Multiplication.	SELECT fieldname1 * fieldname2
x / y	Division.	SELECT fieldname1 / fieldname2
MOD(x, y)	Modulo.	SELECT MOD(fieldname1, fieldname2)
LN(expr)	Logarithm (base: e).	SELECT ln(expr)
LOG10(expr)	Logarithm (base 10).	SELECT LOG10(expr)
POWER(expr,power)	Power of expr .	SELECT POWER(expr ,2)
SQRT(expr)	Square root of expr .	SELECT SQRT(expr)
SIN(expr)	Sine.	SELECT SIN(expr)
COS(expr)	Cosine.	SELECT COS(expr)
TAN(expr)	Tangent.	SELECT TAN(expr)
COT(expr)	Cotangent.	SELECT COT(expr)
ASIN(expr)	Arcsine.	SELECT ASIN(expr)

Statement	Description	Example
ACOS(expr)	Arc cosine.	SELECT ACOS(expr)
ATAN(expr)	Inverse tangent.	SELECT ATAN(expr)

Examples

ACOS(expr)

Calculates the arccosine of a parameter value. $y = \arccos x$, where x is within $[-1,1]$.

1. Example field
x:0.5
2. Query and analysis statement

```
select ACOS(x)
```
3. Query and analysis result

Table 4-35 Query and analysis result

x	EXPR\$1
0.5	1.0471975511965979

ATAN(expr) function

Calculates the arctangent of a parameter value. $y = \arctan x$, where x is R.

1. Example field
x:0.5
2. Query and analysis statement

```
select ATAN(X)
```
3. Query and analysis result

Table 4-36 Query and analysis result

x	EXPR\$1
0.5	1.0471975511965979

ATAN2(expr)

Returns the angle θ from rectangular coordinates (x, y) to polar coordinates (r, θ) .

1. Example field
x:3; y:4
2. Query and analysis statement

```
SELECT x, y, ATAN2(x,y)
```
3. Query and analysis result

Table 4-37 Query and analysis result

x	y	EXPR\$0
3	4	0.6435011087932844

4.7.7 SQL Time Functions

Description

Time functions can be used with **__time**. Any column stored as a millisecond timestamp can use the **MILLIS_TO_TIMESTAMP** function, or any column stored as a string timestamp can use the **TIME_PARSE** function. By default, the UTC time is used for time operations. You can change the time zone by setting the **timezone** parameter to the name of another time zone (for example, **Asia/Shanghai**) or to an offset (for example, **+08:00**).

Syntax

Statement	Description	Example
CURRENT_DATE	Current date in the connected time zone.	SELECT CURRENT_DATE
CURRENT_TIMESTAMP	Current timestamp in the connected time zone.	SELECT CURRENT_TIMESTAMP
DATE_TRUNC(<unit>,<expr >)	Truncates the timestamp and returns it as a new timestamp.	SELECT DATE_TRUNC('minute',__time)

Statement	Description	Example
<p>TIME_FORMAT(<expr>,<pattern>,<timezone>))</p>	<p>Parses a character string into a timestamp using the specified pattern or ISO 8601 (for example, 2000-01-02T03:04:05Z). timezone (if specified) should be a time zone name (such as America/Los_Angeles) or an offset (such as +08:00), and will be used as the time zone of the string that does not include a time zone offset. The mode and time zone must be literal. If a character string cannot be parsed as a timestamp, null is returned.</p>	<pre>SELECT TIME_FORMAT(__time,'yy-MM-dd HH:mm:ss','+08:00')</pre>
<p>TIME_PARSE(<expr>,<pattern>,<timezone>)</p>	<p>Parses a character string into a timestamp using the specified pattern or ISO 8601 (for example, 2000-01-02T03:04:05Z). timezone (if specified) should be a time zone name (such as America/Los_Angeles) or an offset (such as +08:00), and will be used as the time zone of the string that does not include a time zone offset. The mode and time zone must be literal. If a character string cannot be parsed as a timestamp, null is returned.</p>	<pre>SELECT TIME_PARSE("timestamp ",'yyyy-MM-dd HH:mm:ss','+08:00')</pre>

Statement	Description	Example
MILLIS_TO_TIMESTAMP(expr)	Converts the timestamp to the time format.	SELECT MILLIS_TO_TIMESTAMP(expr)
TIMESTAMP_TO_MILLIS(expr)	Converts the time to the timestamp format.	SELECT TIMESTAMP_TO_MILLIS(expr)
EXTRACT(<extract_unit>FROM expr)	Extracts the time part from expr and returns it as a number.	SELECT EXTRACT(MINUTE FROM __time)
TIMESTAMPDIFF(<unit>,<expr1>,<expr2>)	Returns the unit between expr1 and expr2 .	SELECT TIMESTAMPDIFF (minute, expr1, expr2)
TIME_SERIES	Completes the missing data in the query time window.	TIME_SERIES(__time, period, time_format, [padding_value],<timezone>)

TIME_SERIES

The TIME_SERIES function is used to supplement the missing data in the query time window.

- This function must be used together with ORDER BY and must be the first parameter of ORDER BY.
- The OFFSET statement cannot be used in the query statement.
- The TIME_SERIES function cannot be used as a subquery.

Syntax

```
time_series(__time, period, time_format, [padding_value], <timezone>)
```

Parameter description

Table 4-38

Parameter	Description
__time	Time sequence.
period	Time window size (ISO 8601 standard). For example, P1M (1 month), P1D (1 day), PT1H (1 hour), PT1M (1 minute), and PT1S (1 second).
time_format	Time format of the returned result. (For details, see Joda DateTimeFormat .)

Parameter	Description
padding_value	Supplemented content. Options: <ul style="list-style-type: none"> ● 0 or zero: By default, the missing value is set to 0. ● null: The missing value is set to null. ● last: The missing value is set to the value corresponding to the previous time point. ● next: The missing value is set to the value corresponding to the next time point. ● avg: The missing value is set to the average value of two adjacent time points.
timezone	Time zone, for example, UTC+08:00 (Beijing).

Returned Data Type

The value is of the **bigint** type.

Example

Supplement data based on the time granularity of a day, set the missing value to **0**, and add a time zone.

- Query and analysis statement

```
select time_series(_time, 'P1D', 'yyyy-MM-dd HH:mm:ss', '0', '+08:00') as t_time, count(*) as num
group by t_time order by t_time
```
- Query and analysis results

t_time	num
2021-10-01 08:00:00	5
2021-10-02 08:00:00	0
2021-10-03 08:00:00	0
2021-10-04 08:00:00	21
2021-10-05 08:00:00	17
2021-10-06 08:00:00	0
2021-10-07 08:00:00	34

CURRENT_DATE/CURRENT_TIMESTAMP

CURRENT_DATE returns the ISO 8601 time (UTC time) of 00:00 on the current day. This function can be used to calculate timestamps.

CURRENT_TIMESTAMP returns the ISO 8601 time (UTC time) of the current time. This function can be used to calculate timestamps.

1. Example field
__time: 2023-02-14T02:35:56.706Z
2. Query and analysis statement
select __time,CURRENT_DATE, CURRENT_TIMESTAMP,CURRENT_TIMESTAMP
3. Query and analysis result

Table 4-39 Query and analysis result

__time	CURRENT_DATE	CURRENT_TIMESTAMP
2023-02-14T02:35:56.706Z	2023-02-14T00:00:00.000Z	2023-02-14T14:35:57.000Z

DATE_TRUNC (<unit>, <timestamp_expr>)

Truncates <timestamp_expr> to the precision specified in <unit>, fills in 0, and returns the value as a new timestamp. The unit is case-insensitive, and can be milliseconds, second, minute, hour, day, week, month, quarter, year, decade, century, or millennium.

1. Example field
__time: 2023-02-14T02:35:56.706Z
2. Query and analysis statement
SELECT __time,DATE_TRUNC('minute', __time),DATE_TRUNC('day', __time),DATE_TRUNC('year', __time)
3. Query and analysis result

Table 4-40 Query and analysis result

__time	EXPR\$1	EXPR\$2	EXPR\$3
2023-02-14T02:35:56.706Z	2023-02-15T08:50:00.000Z	2023-02-15T00:00:00.000Z	2023-01-01T00:00:00.000Z

TIME_PARSE(<string_expr>, [<pattern>, [<timezone>]])/ TIME_FORMAT(<timestamp_expr>, [<pattern>, [<timezone>]])

Parses the given character string <timestamp_expr> into a timestamp according to the custom <pattern> parameter and **Joda DateTimeFormat** mode. If <pattern> is not specified, the character string is parsed according to ISO 8601 by default. <timezone> is optional.

Parses the given timestamp <timestamp_expr> into a character string according to the custom <pattern> parameter and **Joda DateTimeFormat** mode. If <pattern> is not specified, the timestamp is parsed according to ISO 8601 by default. <timezone> is optional.

1. Example field
__time: 2023-02-16T07:38:25.306Z
start_time:2023-02-14 02:35:56

2. Query and analysis statement
SELECT __time,TIME_PARSE(start_time,'yyyy-MM-dd HH:mm:ss'),TIME_FORMAT(__time,'yyyy-MM-dd HH:mm:ss')
3. Query and analysis result

Table 4-41 Query and analysis result

__time	EXPR\$1	EXPR\$2
2023-02-16T07:38:25.306Z	2023-02-14T02:35:56.000Z	2023-02-16 07:38:25

MILLIS_TO_TIMESTAMP(millis_expr)/ TIMESTAMP_TO_MILLIS(timestamp_expr)

MILLIS_TO_TIMESTAMP converts a number of milliseconds into a timestamp in ISO 8601 format. The converted parameter can be used to calculate timestamps. TIMESTAMP_TO_MILLIS converts a timestamp to a number of milliseconds.

1. Example field
__time: 2023-02-16T07:54:15.106Z,start_time: 1676534055106
2. Query and analysis statement
SELECT __time,MILLIS_TO_TIMESTAMP(start_time),TIMESTAMP_TO_MILLIS(__time)
3. Query and analysis result

Table 4-42 Query and analysis result

__time	EXPR\$1	EXPR\$2
2023-02-16T07:54:15.106Z	2023-02-16T07:54:05.000Z	1676534055106

TIME_EXTRACT(<timestamp_expr>,[<unit>],[<timezone>])/EXTRACT(<unit> FROM timestamp_expr)

TIME_EXTRACT extracts time from **timestamp_expr** and returns it as a number. The unit can be EPOCH (Unix time in seconds since the epoch), SECOND (second of the current minute), MINUTE (minute of the current hour), HOUR (hour of the current day), DAY (day of the current month), DOW (day of the current week), DOY (day of the current year), WEEK (week of the current year), MONTH (month of the current year), QUARTER (quarter of the current year), or YEAR (current year). <timezone> is optional. EXTRACT is short for TIME_EXTRACT.

1. Example field
__time: 2023-02-16T07:54:15.106Z,start_time: 1676534055106
2. Query and analysis statement
SELECT __time,MILLIS_TO_TIMESTAMP(start_time),TIMESTAMP_TO_MILLIS(__time)
3. Query and analysis result

Table 4-43 Query and analysis result

__time	EXPR\$1	EXPR\$2
2023-02-16T07:54:15.106Z	2023-02-16T07:54:05.000Z	1676534055106

Reference Information

- **Unit**

unit	Description
second	Second
minute	Minute
hour	Hour
day	Day
week	Week
month	Month
quarter	Quarter
year	Year

- **extract_unit**

extract_unit	Description
SECOND	Second
MINUTE	Minute
HOUR	Hour
DAY	Day of a month
DOW	Day of a week
DOY	Day of a year
WEEK	Week of a year
MONTH	Month
QUARTER	Quarter
YEAR	Year

4.7.8 SQL Extrema Functions

Description

Extrema functions return the maximum or minimum value after evaluating zero or more specified fields. For details, see [Table 4-44](#).

Pay attention to the following points when using the extrema functions:

- If no field is set, null is returned. Fields must be able to be converted to common data types.
- If all fields are null, null is returned. If only some fields are null, these fields will be ignored.
- If a field contains both numbers and strings, the function compares them as strings.
- If all columns are integers, the function compares them as **long** values.
- If all fields are numbers and at least one is a **float** value, the function compares them as **float** values.

Syntax

```
SELECT GREATEST(fieldname1,fieldname2) AS the_greatest_field
```

Extrema Function Statements

Table 4-44 Extrema function statements

Statement	Description	Example
GREATEST([expr1, ...])	Returns the maximum value among zero or multiple fields.	SELECT GREATEST(fieldname1,fieldname2)
LEAST([expr1, ...])	Returns the minimum value among zero or multiple fields.	SELECT LEAST(fieldname1,fieldname2)

4.7.9 SQL String Functions

Description

SQL provides the string function for concatenating strings and converting between uppercase and lowercase letters. For details, see [Table 4-45](#).

In the SQL syntax, characters must be enclosed in single quotation marks ('). Fields or table names are either not enclosed or are enclosed in double quotation marks ("). For example, 'msg' indicates the string msg, while msg or "msg" indicates the structured msg field.

Syntax

```
SELECT (fieldname1 || fieldname2) AS fieldname1_fieldname2
```

String Function Statements

Table 4-45 String function statements

Statement	Description	Example
CONCAT(expr1, expr2...)	Concatenates all listed strings.	SELECT str1, str2, str3, CONCAT(str1, str2, str3) WHERE str1 IS NOT NULL
TEXTCAT(expr, expr)	Concatenates two strings.	SELECT str1, str2, TEXTCAT(str1, str2) WHERE str1 IS NOT NULL
STRING_FORMAT(pattern[, args...])	Formats strings based on the Java string format.	SELECT str1, STRING_FORMAT(str1, '%s') WHERE str1 IS NOT NULL
LENGTH(expr)	Returns the length of a string, which is the number of UTF-16 characters in the string.	SELECT LENGTH(str1) WHERE str1 IS NOT NULL
LOWER(expr)	Converts a string into lowercase letters.	SELECT LOWER(str1) WHERE str1 IS NOT NULL
POSITION(string1 IN string2 [FROM fromIndex])	Returns the index of the position where string1 appears in string2 for the first time. The search starts from the specified index. If no index is specified, the search starts from index1 . If string1 does not exist in string2 , 0 is returned.	SELECT POSITION(str1 IN str2 FROM 5)
REGEXP_EXTRACT(expr, pattern, [index])	Extracts a string that matches a specified regular expression in <i>expr</i> . The index starts from 1. If no match is found, null is returned. If no index is specified or the index is 0, the first matched substring is returned. For exact match, add a caret (^) before the regular expression and a dollar sign (\$) after the regular expression.	SELECT REGEXP_EXTRACT(str1, '[A-Za-z]+://[A-Za-z0-9.-]+(/[^\]*)', 5)

Statement	Description	Example
REGEXP_LIKE(expr, pattern)	Checks whether a string matches a specified regular expression. For exact match, add a caret (^) before the regular expression and a dollar sign (\$) after the regular expression. The usage of this function is similar to that of the LIKE statement. The difference is that the LIKE statement searches for content with a specified pattern.	SELECT REGEXP_LIKE(str1, '\.(jpg jpeg png gif)\$')
REPLACE(expr, pattern, replacement)	Uses replacement to replace the substring same as pattern in expr .	SELECT REPLACE(expr,pattern, replacement)
STRPOS(string1, string2)	Returns the index of the position where string2 appears in string1 for the first time. The search starts from index1 . If no result is found, 0 is returned.	SELECT STRPOS(str1, str2) WHERE str1 IS NOT NULL AND str2 IS NOT NULL
SUBSTRING(expr, index, [length])	Subtracts a string. <i>length</i> indicates how many UTF-16 characters will be extracted.	SELECT SUBSTRING(str1, 3, 10) WHERE str1 IS NOT NULL
RIGHT(expr, [length])	Truncates a specified length from the right to the left of a string.	SELECT RIGHT(str1, 5) WHERE str1 IS NOT NULL
LEFT(expr, [length])	Truncates a specified length from the left to the right of a string.	SELECT LEFT(str1, 5) WHERE str1 IS NOT NULL
SUBSTR(expr, index, [length])	Same as SUBSTRING.	SELECT SUBSTR(str1, 3, 10) WHERE str1 IS NOT NULL
UPPER(expr)	Converts a string into uppercase letters.	SELECT UPPER(str1) WHERE str1 IS NOT NULL
REVERSE(expr)	Reverses the character string.	SELECT REVERSE(str1) WHERE str1 IS NOT NULL

Statement	Description	Example
LPAD(expr, length, chars)	Adds specified characters to the left of a string until the string reaches the specified length. If the specified length is less than the actual string length, the string is truncated based on the specified length. If the string or the specified character is null, null is returned. If the specified character is empty, no padding is performed. Characters may be deleted if necessary.	SELECT LPAD(str1, 50, 'testStr') WHERE str1 IS NOT NULL
RPAD(expr, length, chars)	Adds specified characters to the right of a string until the string reaches the specified length. If the specified length is less than the actual string length, the string is truncated based on the specified length. If the string or the specified character is null, null is returned. If the specified character is empty, no padding is performed. Characters may be deleted if necessary.	SELECT RPAD(str1, 50, 'testStr') WHERE str1 IS NOT NULL
CONTAINS_STRING(<expr>, str)	Checks whether expr contains the str string.	SELECT CONTAINS_STRING(log_level, 'warn')
ICONTAINS_STRING(<expr>, str)	Checks whether expr contains the str string. The string is not case sensitive.	SELECT ICONTAINS_STRING(log_level, 'WARN')

Examples

REPEAT

REPEAT(expr, [N]) repeats **expr** for N times.

1. Example field
field4:is
2. Query and analysis statement
select field4, REPEAT(field4, 3)
3. Query and analysis result

Table 4-46 Query and analysis result

field4	EXPR\$1
Is	isis

4.7.10 SQL SPLIT Functions

Description

SPLIT functions split a character string using a delimiter, and return the substrings.

Syntax

```
SELECT split_to_map(x, delimiter01, delimiter02)
```

SPLIT Function Statements

State ment	Description	Example	Parameter
split	Splits a character string using a delimiter and returns the substrings.	split(x, delimiter,[limit])	<ul style="list-style-type: none"> • x: varchar parameter value. • delimiter: delimiter. • limit: an integer greater than 0, specifying the maximum number into which the character string can be split.
split_part	Splits a character string using a specified delimiter and returns the indexed content.	split_part(x, delimiter, part)	<ul style="list-style-type: none"> • x: varchar parameter value. • delimiter: delimiter. • part: index of the field to return.
split_to_map	Splits a character string using the first delimiter, and then splits the string again using the other delimiter.	split_to_map(x, delimiter01, delimiter02)	<ul style="list-style-type: none"> • x: varchar parameter value. • delimiter01: delimiter 1. • delimiter02: delimiter 2.

Examples

- **split**

Splits a target character string according to a specified string. **limit** specifies the maximum number of words to split into. If this parameter is not specified, all words are split by default.

- a. Example field

Id: dc1dab7e-b045-4e77-bda4-914d083d1bf7

- b. Query and analysis statement

```
SELECT split(Id,'-'), split(Id,'-',2)
```

- c. Query and analysis result

Table 4-47 Query and analysis result of the split function

EXPR\$0	EXPR\$1
["dc1dab7e","b045","4e77","bda4","914d083d1bf7"]	["dc1dab7e","b045-4e77-bda4-914d083d1bf7"]

- **split_part**

Splits a character string and returns the indexed content. The index starts from 0. If it is negative or greater than the number of substrings, an empty string is returned.

- a. Example field

Id: dc1dab7e-b045-4e77-bda4-914d083d1bf7

- b. Query and analysis statement

```
SELECT split_part(Id,'-',1)
```

- c. Query and analysis result

Table 4-48 Query and analysis result of the split_part function

EXPR\$0
b045

- **split_to_map**

Splits a character string using the first delimiter, and then splits the string again using the other delimiter. The result is displayed in format `{"KEY1":"VALUE1","KEY2":"VALUE2"}`. If a value cannot be split again, it is left blank.

- a. Example field

Request: request_id:"e3ac4b70c7d244f080d434e300d8065a";
request_time: "1674965051000"

- b. Query and analysis statement

```
SELECT split_to_map(Request,';',':')
```

- c. Query and analysis result

Table 4-49 Query and analysis result of the split_to_map function

EXPR\$0
{"request_id ":"e3ac4b70c7d244f080d434e300d8065a", "request_time":"1674965051000"}

4.7.11 SQL Comparison Operators

Description

Comparison operators are used to compare two values and return **true** or **false**. Values of the numeric type and contents of the string type can be compared. For example, comparison operators can be used to check whether the numeric field *num1* < *num2* is true and whether *str1* of the string type exists in string *strs*. For details, see [Table 4-50](#).

Syntax

```
SELECT fieldname1 WHERE fieldname1 > fieldname2
```

Comparison Operator Statements

Table 4-50 Comparison operator statements

Statement	Description	Example
<code>x = y</code>	Equal to	<code>SELECT num1 < num2</code>
<code>x <> y</code>	Not equal to	<code>SELECT num1 <> num2</code>
<code>x > y</code>	Greater than	<code>SELECT num1 > num2</code>
<code>x >= y</code>	Greater than or equal to	<code>SELECT num1 >= num2</code>
<code>x < y</code>	Less than	<code>SELECT num1 < num2</code>
<code>x <= y</code>	Less than or equal to	<code>SELECT num1 <= num2</code>
<code>x BETWEEN y AND z</code>	Equivalent to <code>x >= y AND x <= z</code>	<code>SELECT num1 BETWEEN num2 AND num3</code>
<code>x NOT BETWEEN y AND z</code>	Equivalent to <code>x < y OR x > z</code>	<code>SELECT num1 NOT BETWEEN num2 AND num3</code>

Statement	Description	Example
x LIKE pattern	Returns true if x matches the SQL LIKE mode.	SELECT str1 LIKE '*'
x NOT LIKE pattern	Returns true if x does not match the SQL LIKE mode.	SELECT str1 NOT LIKE '*'
x IS NULL	Returns true if x is null or an empty string.	SELECT str1 IS NULL
x IS NOT NULL	Returns true if x is neither null nor an empty string.	SELECT str1 IS NOT NULL
x IN (values)	If x is one of the listed values, true is returned.	SELECT str1 IN ('testStr1', 'testStr2')
x NOT IN (values)	If x is not listed, true is returned.	SELECT str1 NOT IN ('testStr1', 'testStr2')
x IN (subquery)	Returns true if x is returned through a specified subquery.	SELECT str1 WHERE str2 IN (SELECT DISTINCT str2 LIMIT 100)
x NOT IN (subquery)	If x is not returned through the specified subquery, true is returned.	SELECT str1 NOT IN (SELECT str2 LIMIT 100)

4.7.12 SQL IP Address Functions

Description

The IPv4 function can use an IPv4 address in dotted decimal notation (example: **192.168.0.1**) or an IP address in integer format (example: **3232235521**). The **subnet** parameter is an IPv4 subnet in CIDR notation (example: **192.168.0.0/16**).

IP Function Statements

Statement	Description	Example
IPV4_MATCH(address,subnet)	If subnet belongs to address , true is returned. Otherwise, false is returned. If address is not a valid IPv4 address, false is returned. This function is more efficient if address is an integer rather than a string.	SELECT IPV4_MATCH (address,subnet)
IPV4_PARSE(address)	Parses address into an integer IPv4 address. If address is a valid IPv4 address, it can be parsed. If address is not a valid IPv4 address, null is returned.	SELECT IPV4_PARSE(address)
IPV4_STRINGIFY(address)	Converts an integer address into an IPv4 address in dotted notation. If address is a valid integer IPv4 address, it can be parsed. If address is not an IPv4 address, null is returned.	SELECT IPV4_STRINGIFY(address)

Examples

IPV4_MATCH(address, subnet)

If **address** belongs to **subnet**, **true** is returned. Otherwise, **false** is returned. If **address** is not a valid IPv4 address, **false** is returned. This function is more efficient if **address** is an integer rather than a string.

1. Example field
Ipv4: 192.168.1.18
2. Query and analysis statement

```
select IPV4,IPV4_MATCH(Ipv4, '192.168.0.0/16')
```
3. Query and analysis result

Table 4-51 Query and analysis result

IPV4	EXPR\$1
192.168.1.18	true

IPV4_PARSE(address)/IPV4_STRINGIFY(address)

Parses **address** into an integer IPv4 address. If **address** is a valid IPv4 address, it can be parsed. If **address** is not a valid IPv4 address, **null** is returned.

1. Example field
Ipv4: 192.168.0.1
Num: 3232235521
2. Query and analysis statement

```
select IPV4_PARSE(Ipv4), IPV4_STRINGIFY(Num)
```
3. Query and analysis result

Table 4-52 Query and analysis result

EXPR\$0	EXPR\$1
-1062731775	192.168.0.1

4.7.13 SQL Reduction Functions

Description

Reduction functions operate on zero or a list of expressions and return a single expression. If no expression is passed as a parameter, the result is **NULL**. All expressions must be converted to public data types.

- If all parameters are **NULL**, the result is **NULL**. Otherwise, **NULL** parameters are ignored.
- If all parameters contain both numbers and strings, they are interpreted as strings.
- If all parameters are integers, they are interpreted as long integers.
- If all parameters are numeric and at least one of them is double, all parameters are interpreted as double.

Syntax

```
GREATEST([expr1, ...]) / LEAST([expr1, ...])
```

Examples

GREATEST([expr1, ...])/LEAST([expr1, ...])

The GREATEST function returns the maximum value in zero or a list of expressions.

The LEAST function returns the minimum value in zero or a list of expressions.

1. Example field
Num: 11785730
2. Query and analysis statement

```
select Num,GREATEST( "Num"/10,(select count(1)) ),LEAST("Num"/10,(select count(1)))
```
3. Query and analysis result

Table 4-53 Query and analysis result of reduction functions

Num	EXPR\$1	EXPR\$2
11785730	1178573	1

4.7.14 Other SQL Functions

Description

SQL functions also support some conversion types and logical operations such as CASE and WHEN. For details, see [Table 4-54](#).

Syntax

```
SELECT CAST(fieldname1 AS VARCHAR) AS fieldname1_str
```

Other Function Statements

Table 4-54 Other function statements

Keyword	Description	Example
CAST(value AS TYPE)	Converts the data type. Data types can be converted only to varchar or float .	SELECT fieldname1, CAST(fieldname1 AS VARCHAR)
CASE WHEN boolean_expr1 THEN result1 \[WHEN boolean_expr2 THEN result2 ... \] \[ELSE resultN \] END	Simple CASE function.	SELECT CASE WHEN httpStatus = 200 THEN 1 ELSE 0 END
NULLIF(value1, value2)	If <i>value1</i> is equal to <i>value2</i> , null is returned. Otherwise, <i>value1</i> is returned.	SELECT fieldname1, fieldname2, NULLIF(fieldname1, fieldname2)

Keyword	Description	Example
NVL(expr,expr-for-null)	If expr is null or an empty string, this function returns expr-for-null .	SELECT NVL(str1, 'expr-for-null')

4.7.15 SQL JOIN Syntax

The JOIN clause queries data in two or more tables. This section describes how to use the JOIN clause.

Syntax

```
select key
from t1
LEFT|RIGHT|INNER JOIN t2
on t1.key=t2.key
```

Currently, LTS supports LEFT JOIN, RIGHT JOIN, and INNER JOIN.

Table 4-55

JOIN Mode	Description
LEFT JOIN	Join the right table (t2) based on the result of the left table (t1). If the table name contains only digits, add double quotation marks to the table name to convert it into a string. For example, if the table name is 123 , enter " 123 " in the JOIN statement.
RIGHT JOIN	Join the left table (t1) based on the result of the right table (t2).
INNER JOIN	Intersection data of the results (elb1 and elb2) of two tables

Examples

There are two tables. **access** indicates the host ingestion metrics, including path, latency, and status code. **host** indicates the host metrics, including CPU and memory. You can use JOIN to associate ingestion and host metrics and view different dimensions of metrics of the same host.

- **LEFT JOIN**
 - a. Query statements

```
SELECT
  "access"._time,
  "access".host_ip,
  "access".cost,
```

```
"host".cpu,
"host".memory
FROM
log "access"
LEFT JOIN (select memory,cpu,host_ip from log) host ON "access".host_ip = "host".host_ip
```

b. 60 data records returned

- **RIGHT JOIN**

a. Query statements

```
SELECT
"access"._time,
"host".host_ip,
"access".cost,
"host".cpu,
"host".memory
FROM
log "access"
RIGHT JOIN (select memory,cpu,host_ip from log) host ON "access".host_ip = "host".host_ip
```

b. 60 data records returned

- **INNER JOIN**

a. Query statements

```
SELECT
"access"._time,
"host".host_ip,
"access".cost,
"host".cpu,
"host".memory
FROM
log "access"
INNER JOIN (select memory,cpu,host_ip from log) host ON "access".host_ip = "host".host_ip
```

b. 45 data records returned


4.7.16 SQL Query Example

This section describes how to query ELB raw logs in LTS.

Step 1 Log in to the LTS console.

Step 2 Choose **Log Management** in the navigation pane, and click a log group or stream to access the log details page.

Step 3 View logs on the **Log Search** tab page. The system obtains ELB raw logs.

Step 4 Click  in the upper right corner. On the page that is displayed, select **Cloud Structuring Parsing**.

Step 5 Select **Structuring Template**, and configure log structuring based on the ELB template. For other logs, you can select other structuring templates.

Step 6 On the **Visualization** tab page, enter a SQL statement in the search box.

----End

Display of Query Results

Table

The following statement queries the host, the number of logs of each **request_uri**, the request body size in MB, and the ratios of 2xx, 3xx, 4xx, and 5xx responses.

Then the results are displayed in a table, and sorted in descending order by the number of logs.

```
SELECT "router_request_uri" as "request_uri", "host", COUNT(*) as pv,
round(sum(body_bytes_sent) / 1024.0 , 5) as "body_bytes_sent(MB)",
round(sum(case when status >= 200 and status < 300 then 1 else 0 end ) * 100.0 / COUNT(1), 6) as "2xx
ratio(%)",
round(sum(case when status >= 300 and status < 400 then 1 else 0 end ) * 100.0 / count(1), 6) as "3xx
ratio(%)",
round(sum(case when status >= 400 and status < 500 then 1 else 0 end ) * 100.0 / count(1), 6) as "4xx
ratio(%)",
round(sum(case when status >= 500 and status < 600 then 1 else 0 end ) * 100.0 / count(1), 6) as "5xx
ratio(%)",
GROUP BY "host", "router_request_uri"
ORDER BY pv DESC
LIMIT 100
```

Bar chart

The following statement queries the number of requests for each request URL and displays the results in a bar chart, with **request_uri** set as the X axis and **pv** as the Y axis.

```
SELECT "router_request_uri" as "request_uri", "host", COUNT(*) as pv,
round(sum(body_bytes_sent) / 1024.0 , 5) as "body_bytes_sent(MB)",
round(sum(case when status >= 200 and status < 300 then 1 else 0 end ) * 100.0 / COUNT(1), 6) as "2xx
ratio(%)",
round(sum(case when status >= 300 and status < 400 then 1 else 0 end ) * 100.0 / count(1), 6) as "3xx
ratio(%)",
round(sum(case when status >= 400 and status < 500 then 1 else 0 end ) * 100.0 / count(1), 6) as "4xx
ratio(%)",
round(sum(case when status >= 500 and status < 600 then 1 else 0 end ) * 100.0 / count(1), 6) as "5xx
ratio(%)",
GROUP BY "host", "router_request_uri"
ORDER BY pv DESC
LIMIT 100
```

Line chart

The following statement displays a line chart based on the query result, with **_time_** set as the X axis and **QPS** as the Y axis. The chart shows the queries per second (QPS) change at an interval of 5s in the query period.

```
select TIME_FORMAT(TIME_CEIL(TIME_PARSE(SUBSTRING(time_iso8601, 2, 25) , 'yyyy-MM-
dd"TT"HH:mm:ssZZ'),'PT5S'),'yyyy-MM-dd HH:mm:ss','+08:00') AS _time_ , COUNT(*) as QPS from log group
by _time_
```

Pie chart

The following statement queries the percentage of different values of status in the query period, and displays the results in a bar chart, with **Category** set to **status** and **Value** set to **rm**.

```
SELECT status, COUNT(1) AS rm GROUP BY status
```

Number chart

The following statement queries the total number of normal requests in the last hour and displays the results in a number chart.

```
SELECT count(*) AS normalRequest WHERE status = 200
```

5 Log Visualization

5.1 Overview

LTS can analyze structured logs with [SQL query syntax](#) and display analysis results in various charts, catering to visualization needs of different O&M and operations analysis scenarios.

With more than 40 out-of-the-box dashboard templates and preset sample data that are easy to use and configure, LTS lowers the usage threshold and reduces repetitive development work. Cloud service logs ingested to LTS can be directly displayed in dashboard templates. This helps enterprises realize digital operations based on log data and facilitates digital transformation.

The log visualization function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Table 5-1 Visualization methods

Method	Description
Statistical charts	Statistical charts, such as tables, bar charts, and line charts, are rendered by LTS based on SQL query syntax . For details, see Visualizing Logs in Statistical Charts .
Dashboards	<ul style="list-style-type: none">LTS provides dashboards for real-time data analysis. A dashboard consolidates and synchronously saves various statistical charts derived from SQL query and analysis. For details, see Visualizing Logs in Dashboards.

5.2 Visualizing Logs in Statistical Charts

5.2.1 Statistical Charts

After logs are reported to LTS, you can search for key log data using [SQL analysis syntax](#) and view the results in charts. This helps you track log data trends effectively. You can also save chart analysis results to the dashboard for long-term monitoring.

Constraints

- Up to 100 charts can be created for a log stream.
- Up to 50 charts can be added to a dashboard.

Statistical Chart Types

Data in different scenarios can be displayed in table charts, bar charts, and line charts. For details, see [Table 5-2](#).

Table 5-2 Chart types

Chart Type	Scenario
Table chart	A table chart is the most common way for data display. Data can be collected and compared through data structuring. This chart type is applicable to most scenarios.
Bar chart	A bar chart describes the classified data and intuitively compares the size of each classification item. This chart type is applicable to classified statistics, such as the number of occurrences of each error code type in the past day.
Line chart	A line chart requires the time sequence field to organize and aggregate metrics based on the time sequence. This chart type intuitively reflects the changes of metrics over time.
Pie chart	A pie chart shows the proportion of each category. The proportion of each category is measured by the sector size. This chart type is applicable to scenarios such as error code proportion analysis.
Number chart	A number chart describes a single metric. Generally, key metrics with business value are selected. This chart type is applicable to single-metric statistics such as daily, weekly, monthly PVs and UVs.
Digital line	A digital line chart combines a line chart and a number chart. The line chart displays data trends and changes, and the number chart displays key metrics. This chart type is applicable when trends and key data points need to be displayed at the same time.

Chart Type	Scenario
Map	A map displays the geographical location of data based on the location of a chart. Generally, a map displays the distribution of data in different geographical areas. This chart type is applicable to geographical location statistics, such as geographical distribution of attack IP addresses.
Funnel chart	A funnel chart is applicable to the service procedure with a single flow direction and a single path. The funnel chart collects statistics on each phase and uses a trapezoidal area to indicate the difference between the service volume of a phase and that of the previous phase.

Operation Description

You can create and save a chart, or save a chart as another file. For details, see [Table 5-3](#).

Table 5-3 Operation description

Operation	Description
Create	Selecting a chart type and saving the chart analysis result to the dashboard
Save	Saving the current visualized chart
Save As	Copying the existing visualized chart
Download	Downloading the chart analysis result to an Excel file
Show Chart	Expanding the visualized chart of the current log stream
Hide Chart	Hiding the visualized chart of the current log stream

5.2.2 Table

Table charts are one of the most common data display types and the most basic method for organizing and sorting data. By sorting data, you can quickly reference and analyze the data. By default, the data obtained through the query and analysis syntax is displayed in a table chart.

Checking Tables

- Step 1** Log in to the LTS console.
- Step 2** In the navigation pane, choose **Log Management**.
- Step 3** On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.

Step 4 Choose **Log Analysis**.

Step 5 Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.

Step 6 By default, the query results are displayed in a table below the log. Set parameters in the **General Settings** right pane by referring to [Table 5-4](#).

Figure 5-1 Table chart

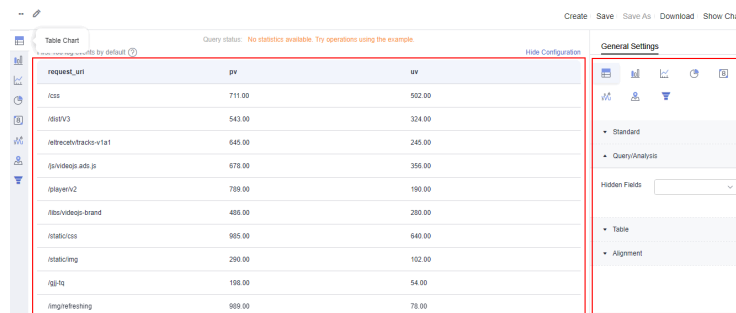


Table 5-4 Table parameters

Type	Parameter	Description
Standard	Format	Displays the table chart data in the specified format.
	Unit	Specifies the unit of the data in the customized table chart.
	Decimal Places	Sets the number of decimal places to be displayed.
	Title Font Size	Sets the font size of the chart title.
Query/Analysis	Hidden Fields	Select a target field to hide it in the table chart.
Table	Records per Page	Number of data records displayed on each page.
	Display Total	Displays the total number of entries in the table.
Column	Alignment	Alignment mode of table data. The options are left alignment, right alignment, and center alignment.
	Filtering	After this function is enabled, you can search for data in the table column.
	Sorting	After this function is enabled, you can sort data in the table.

Type	Parameter	Description
	Font Size	Size of the table font. The value ranges from 12 px to 24 px.

----End

5.2.3 Bar Chart

A bar chart uses vertical or horizontal bars to compare values between categories. It shows the data of different categories and counts the number of elements in each category.

The bar chart provided by LTS consists of vertical bars and horizontal bars. The width of the rectangle block is fixed, and the height indicates the value. When multiple columns of data are mapped to the Y axis, the data is displayed in group columns.

By default, vertical bars are used. You can select either vertical or horizontal bars. A bar chart contains the following elements:

- X axis (horizontal axis)
- Y axis (vertical axis)
- Rectangular block
- Legend

Checking Bar Charts

Step 1 Log in to the LTS console.

Step 2 In the navigation pane, choose **Log Management**.

Step 3 On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.

Step 4 Choose **Log Analysis**.

Step 5 Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.


Step 6 Click  to display the queried data in a bar chart. Set parameters in the **General Settings** right pane by referring to [Table 5-5](#).

Figure 5-2 Bar chart

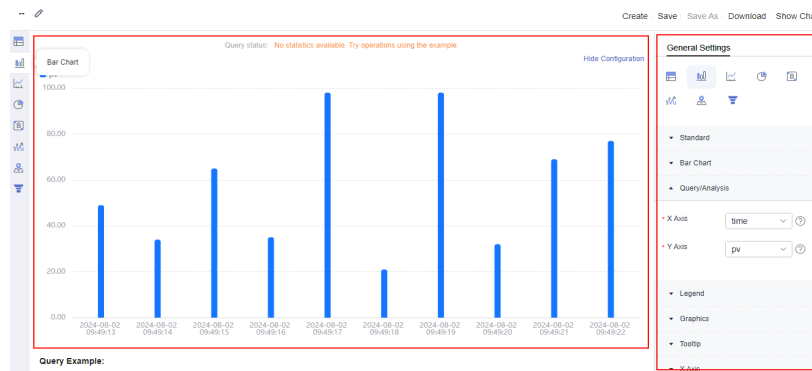


Table 5-5 Bar chart parameters

Type	Parameter	Description
Standard	Format	Displays the Y axis in the specified format.
	Unit	Customize the unit of the Y axis.
	Decimal Places	Sets the number of decimal places to be displayed.
	Title Font Size	Sets the font size of the chart title.
Bar Chart	Direction	Select Bar chart or Horizontal bar chart .
	Column Width	Sets the column width.
	Display Value	After this function is enabled, the value indicated by each bar is displayed.
	Font Size	Sets the font size of each bar.
	Stacked	After this function is enabled, the Y axis data is displayed in stack mode.
Query/ Analysis	X Axis	Numeric or string data is supported.
	Y Axis	Numeric or string data is supported. You can select multiple data records.
Legend	Hide Legend	After this function is enabled, you can hide the legend and comparison results.
	Legend Position	Position of the legend in the chart. Select Top or Right .
	Comparison Value	Indicates whether to display the maximum value, minimum value, average value, and sum value. You can select multiple options.

Type	Parameter	Description
Graphics	Top Margin	Specifies the distance between the axis and the upper boundary of the graph.
	Bottom Margin	Specifies the distance between the axis and the lower boundary of the graph.
	Left Margin	Specifies the distance between the axis and the left boundary of the graph.
	Right Margin	Specifies the distance between the axis and the right boundary of the graph.
Tooltip	None, Ascending, and Descending	Dialog box configuration. When multiple Y-axis data records are selected, they can be sorted and displayed.
X Axis	Show	After this function is enabled, data on the X axis is displayed.
	X Axis Title	Title of the X axis.
Y Axis	Show	After this function is enabled, the Y axis data is displayed.
	Y Axis Title	Title of the Y axis.
	Position	Position of the Y axis. Select Left or Right .

----End

5.2.4 Line Chart

A line chart is used to analyze the data change trend in a certain period, which is reflected in the following aspects:

- Increment or decrement
- Increase/Decrease rate
- Regularity of increase/decrease (such as periodic change)
- Peak and valley values

Therefore, the line chart is the best choice for analyzing the data change trend over time. In addition, you can draw multiple lines to analyze the change trend of multiple groups of data in the same period, and then analyze the interaction and impact between data (such as in direct or inverse proportion).

Checking Line Charts

Step 1 Log in to the LTS console.

Step 2 In the navigation pane, choose **Log Management**.

Step 3 On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.

Step 4 Choose **Log Analysis**.

Step 5 Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.


Step 6 Click  to display the queried data in a line chart. Set parameters in the **General Settings** right pane by referring to [Table 5-6](#).

Figure 5-3 Line chart

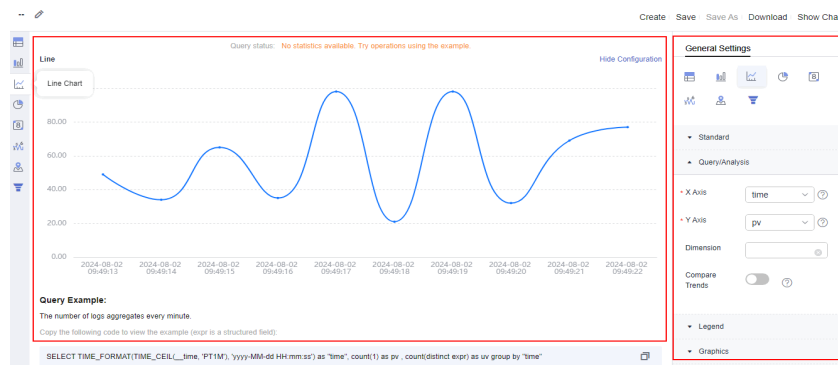


Table 5-6 Line chart parameters

Type	Parameter	Description
Standard	Format	Select K,Mil,Bil, 1,000,000, or Byte,KB,MB from the drop-down list to specify the format of the Y axis.
	Unit	Customize the unit of the Y axis.
	Decimal Places	Sets the number of decimal places to be displayed.
	Title Font Size	Sets the font size of the chart title.
Query/ Analysis	X Axis	Numeric or string data is supported.
	Y Axis	Numeric or string data is supported. You can select multiple data records.
	Dimension	Select a value from the drop-down list. Generally, it is an ordinal variable.
	Compare Trends	This function can be enabled when the X axis shows time data and Dimension is not specified. After this function is enabled, set Comparison to a duration less than or equal to 24 hours. After the setting is complete, compare the data of the current time with the object time data.

Type	Parameter	Description
Legend	Hide Legend	After this function is enabled, you can hide the legend and comparison results.
	Legend Position	Select Top or Right .
	Comparison Value	Indicates whether to display the maximum value, minimum value, average value, and sum value. You can select multiple options.
Graphics	Line Shape	Line type. Select Straight or Curved .
	Line Width	Width of a polyline.
	Show Data Markers	Whether to display the connection points.
	Top Margin	Specifies the distance between the axis and the upper boundary of the graph.
	Bottom Margin	Specifies the distance between the axis and the lower boundary of the graph.
	Left Margin	Specifies the distance between the axis and the left boundary of the graph.
	Right Margin	Specifies the distance between the axis and the right boundary of the graph.
Tooltip	Sort By	Dialog box configuration. When multiple Y-axis data records are selected, they can be sorted and displayed.
X Axis	Show	After this function is enabled, data on the X axis is displayed.
	X Axis Title	Title of the X axis.
Y Axis	Show	After this function is enabled, the Y axis data is displayed.
	Y Axis Title	Title of the Y axis.
	Position	Position of the Y axis. Select Left or Right .

----End

5.2.5 Pie Chart

A pie chart shows the proportion of different categories. Different categories are compared by radian. A pie is divided into multiple sectors based on the proportion of each category. The entire pie indicates the total volume. Each sector indicates the proportion of the category to the total. The sum of all sectors is 100%. A pie chart contains the following elements:

- Sector
- Text percentage
- Legend

Checking Pie Charts

Step 1 Log in to the LTS console.

Step 2 In the navigation pane, choose **Log Management**.

Step 3 On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.

Step 4 Choose **Log Analysis**.

Step 5 Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.


Step 6 Click  to display the queried data in a pie chart. Set parameters in the **General Settings** right pane by referring to [Table 5-7](#).

Figure 5-4 Pie chart

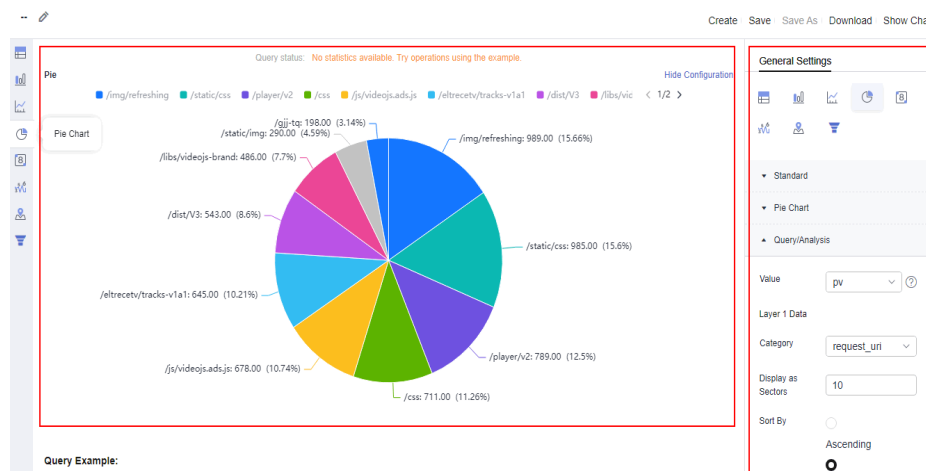


Table 5-7 Pie chart parameters

Type	Parameter	Description
Standard	Format	Select K,Mil,Bil, 1,000,000 , or Byte,KB,MB from the drop-down list to specify the format of the Y axis.
	Unit	Customize the unit of the Y axis.
	Decimal Places	Sets the number of decimal places to be displayed.
	Title Font Size	Sets the font size of the chart title.

Type	Parameter	Description
Pie Chart	Pie Chart Type	<p>Includes pie, cyclic, and Coxcomb charts.</p> <ul style="list-style-type: none"> Pie A pie chart displays the percentage of each part in the whole. It divides a circle into different sectors. The area (or arc length and center angle) of each sector corresponds to the proportion of the data represented by the sector. In this way, the relationship between each part and the whole is intuitively displayed. Cyclic Essentially, a cyclic chart hollows out the center of a pie chart. Cyclic charts are better than pie charts in the following aspects: <ul style="list-style-type: none"> Cyclic charts display more information such as the total number. It is not intuitive to compare two pie charts directly. You can compare two cyclic charts by the cyclic bar length. Coxcomb A Coxcomb chart is not a cyclic chart in essence. Instead, it is a bar chart drawn in the polar coordinate system. Each category is evenly divided by an arc. The radius of the arc indicates the data size. Coxcomb charts are better than pie charts in the following aspects: <ul style="list-style-type: none"> A pie chart can contain a maximum of 10 categorized data records, while a Coxcomb chart can contain 10 to 30 data records. Because the radius and area are squared, a Coxcomb chart magnifies the difference between the values of each category. It is suitable for comparing the values of similar sizes. Due to the periodicity of a circle, a Coxcomb chart is also suitable for displaying data by time period, such as by week and month.
	Show Scale	After this function is enabled, text labels are displayed on the pie chart to describe data details, such as the value and name.
	Scale Text	This parameter can be set to Category, Percent, Category: % , or Category: Value (%) .

Type	Parameter	Description
	Label Position	After Show Scale is enabled, you can set this parameter to adjust the position of the label in the chart.
Query/ Analysis	Value	Specifies the value corresponding to the categorized data.
	Layer 1 Data	
	Category	Specifies the categorized data.
	Display as Sectors	Specifies the number of pie sectors to be displayed.
	Sort By	Specifies the ascending or descending order.
	Display Rest as Others	Specifies whether to display the data of other types as Other .
	Add Layer	Click Add Layer and set the data of the second layer. The data of each layer includes the category, display sectors, sorting mode, and displaying rest sectors as others.
Legend	Hide Legend	After this function is enabled, you can hide the legend and its content.
	Legend	Select Value or Percent , or both.
	Legend Position	Position of the legend in the chart. Select Top or Right .
Graphics	Outer Radius	Specifies the outer radius of the pie chart. The value ranges from 40 to 100.
	Inside Radius	Specifies the inside radius of the pie chart. The value ranges from 0 to 100.
	Top Margin	Specifies the distance between the axis and the upper boundary of the graph.
	Bottom Margin	Specifies the distance between the axis and the lower boundary of the graph.
	Left Margin	Specifies the distance between the axis and the left boundary of the graph.
	Right Margin	Specifies the distance between the axis and the right boundary of the graph.

----End

5.2.6 Number Chart

A number chart is usually used to represent a single data point or key metric. It can better display the relative size of information and data of the same type. The number chart is ideal for displaying key information and data that needs to be highlighted. It allows users to quickly and intuitively understand data trends and key metrics.

Checking Number Charts


- Step 1** Log in to the LTS console.
- Step 2** In the navigation pane, choose **Log Management**.
- Step 3** On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.
- Step 4** Choose **Log Analysis**.
- Step 5** Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.
- Step 6** Click  to display the queried data in a number chart. Set parameters in the **General Settings** right pane by referring to [Table 5-8](#).

Figure 5-5 Number chart

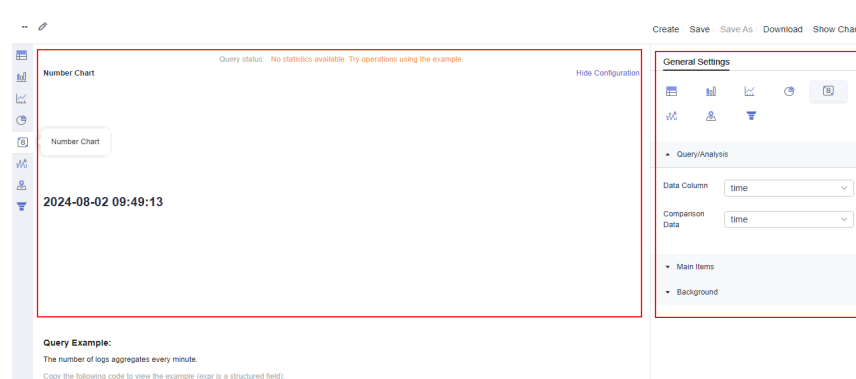


Table 5-8 Number chart parameters

Type	Parameter	Description
Query/ Analysis	Data Column	Numeric or string data is supported.
	Comparison Data	Select a field to compare. The value of the field is displayed in the chart.
Main Items	Title Font Size	Sets the font size of the chart title.
	Format	Displays data in the specified format.
	Data Text Size	Font size of the displayed value. The value ranges from 12 px to 80 px.

Type	Parameter	Description
	Data Unit	Unit of the displayed value.
	Unit Text Size	Font size of the displayed value unit. The value ranges from 12 px to 50 px.
	Decimal Places	Sets the number of decimal places to be displayed.
	Add Comparison Data	After this function is enabled, the value of the field to compare is displayed.
	Comparison Formatting	Displays the data to compare in the specified format.
	Comparison Data Text Size	Font size of the value to compare. The value ranges from 12 px to 50 px.
	Comparison Data Unit	Unit to compare.
	Comparison Unit Text Size	Font size of the value unit to compare. The value ranges from 12 px to 50 px.
	Description	Description of the displayed value and comparison trend. The description is displayed below the value.
Background	Background	Background color of a chart, which can be dark or light.

----End

5.2.7 Digital Line Chart

A digital line chart combines a line chart and a number chart to display the trend and key data points at the same time. This helps users better understand data and its change trend to make better service decisions.

Checking Digital Line Charts

- Step 1** Log in to the LTS console.
- Step 2** In the navigation pane, choose **Log Management**.
- Step 3** On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.
- Step 4** Choose **Log Analysis**.
- Step 5** Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.


Step 6 Click  to display the queried data in a digital line chart. Set parameters in the **General Settings** right pane by referring to [Table 5-9](#).

Figure 5-6 Digital line chart

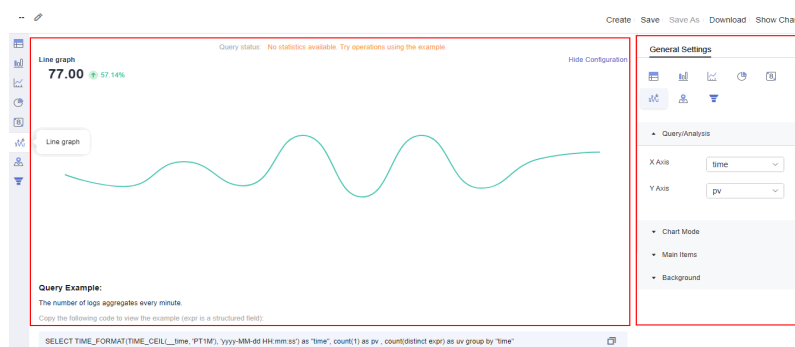


Table 5-9 Digital line chart parameters

Type	Parameter	Description
Query/ Analysis	X Axis	Numeric or string data is supported.
	Y Axis	Numeric or string data is supported. You can select multiple data records.
Chart Mode	Line Shape	Line type. Select Straight or Curved .
Main Items	Title Font Size	Sets the font size of the chart title.
	Number Format	Displays data in the specified format.
	Data Text Size	Font size of the displayed value. The value ranges from 12 px to 80 px.
	Data Unit	Unit of the displayed value.
	Unit Text Size	Font size of the displayed value unit. The value ranges from 12 px to 50 px.
	Decimal Places	Sets the number of decimal places to be displayed.
Background	Background	Background color of a chart, which can be dark or light.

----End

5.2.8 Map

A map is used as the background to display geographic data in different colors and marks. Maps provided by LTS include the China map and the world map. After you use specific functions (**ip_to_province** for the China map and **ip_to_country** for the world map) in query and analysis statements, LTS displays analysis results on a map.

A map contains the following elements:

- Map canvas
- Color block

Checking Maps

Step 1 Log in to the LTS console.

Step 2 In the navigation pane, choose **Log Management**.

Step 3 On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.

Step 4 Choose **Log Analysis**.

Step 5 Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.


Step 6 Click  to display the queried data in a map. Set parameters in the **General Settings** right pane by referring to [Table 5-10](#).

Figure 5-7 Map



Table 5-10 Map parameters

Parameter	Description
Map Type	Sets the region range displayed on the map, including the provincial map of China and the world map.
Province	If the map type is a provincial map of China, this field indicates a specific province, for example, Zhejiang province.
Country/Region	If the map type is a world map, the value of this field is a specific country, for example, China.
Data Column	Select the field whose value needs to be displayed.
Title Font Size	Sets the font size of the chart title.

----End

5.2.9 Funnel Chart

A funnel chart is applicable to unidirectional analysis of a single process with standard service procedure, long period, and many phases. By comparing service data in each phase of the procedure, you can intuitively find and describe the phase where a problem occurs and make decisions to solve the problem. The funnel chart uses a trapezoidal area to indicate the difference between the service volume of a phase and that of the previous phase.

Checking Funnel Charts


- Step 1** Log in to the LTS console.
- Step 2** In the navigation pane, choose **Log Management**.
- Step 3** On the **Log Management** page, select the target log group and log stream. The log stream details page is displayed.
- Step 4** Choose **Log Analysis**.
- Step 5** Enter a query and analysis statement by referring to [SQL Analysis Syntax](#), set the time range, and click **Search**.
- Step 6** Click  to display the queried data in a funnel chart. Set parameters in the **General Settings** right pane by referring to [Table 5-11](#).

Figure 5-8 Funnel chart

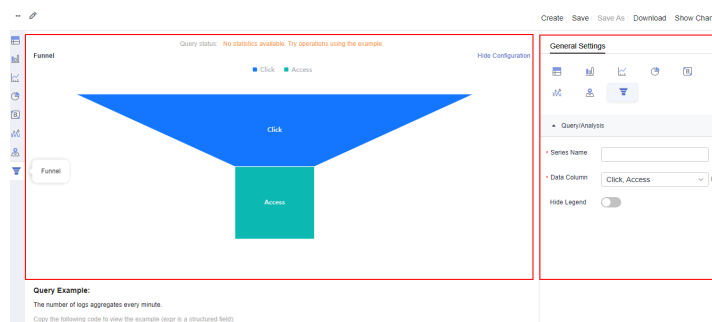


Table 5-11 Funnel chart parameters

Parameter	Description
Series Name	Name of a funnel chart.
Data Column	Select a numeric field. The larger the value of a field, the higher the position of the field.
Hide Legend	After this function is enabled, you can hide the legend names above the funnel chart.

----End

5.3 Visualizing Logs in Dashboards

5.3.1 Creating a Dashboard

Dashboards serve as data visualization tools that summarize and display key performance metrics, important data, and analysis results. They offer a comprehensive overview of service or system statuses.

LTS provides various dashboard templates to visualize log data or retain statistical charts derived from queries and analyses.

Prerequisites

- Logs have been collected.
- Logs have been structured. For details, see [Log Structuring](#).

Constraints

- Up to 100 dashboards can be created in an account.
- Up to 100 charts can be created for a log stream.
- Up to 50 charts can be added to a dashboard.

Creating a Dashboard

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Click . In the **Add Dashboard Group** dialog box, enter a group name.

 **NOTE**

A group name can contain up to 64 characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Do not start or end with a period or underscore (_).

Step 3 Click **OK**. The group is created.

Step 4 Click **Add Dashboard**. On the displayed page, set parameters by referring to [Table 5-12](#).

Table 5-12 Parameters for creating a dashboard

Parameter	Description
Dashboard Name	Enter a unique name to differentiate your dashboard from other dashboards in the log stream. The value can contain up to 255 characters and cannot start or end with a period (.). Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.

Parameter	Description
Enterprise Project Name	Select the required enterprise project. The default value is default . You can click View Enterprise Projects to view all enterprise projects.
Add to Dashboard Group	<p>Add the created dashboard to a group for management.</p> <p>If this parameter is disabled, the created dashboard will be added to the system's default group.</p> <p>If this parameter is enabled, the created dashboard will be added to a group based on the group type you selected.</p> <ul style="list-style-type: none"> • Existing: Select an existing dashboard group. • Create: Enter a dashboard group name to create one.
Lite Mode	<p>Set whether to display the dashboard page in lite mode.</p> <ul style="list-style-type: none"> • If you enable this mode, the buttons for editing and deleting the dashboard and adding filters will not be displayed on the dashboard. • If you disable this mode, the buttons for editing and deleting the dashboard and adding filters will be displayed on the dashboard.
Add Chart	<p>Custom: Select log streams' charts to add to the dashboard.</p> <ol style="list-style-type: none"> 1. Click Add Chart under Custom. 2. Select one or more log streams, select one or more charts by clicking their check box (<input type="checkbox"/>), and click OK. On the dashboard details page displayed, adjust the charts as required and click Save. 3. If there are no suitable charts or no charts at all in a log stream, click Create Chart to create one. <p>Using Dashboard Templates: You can select a custom template (extracted from a created dashboard) or a system template (LTS's built-in template, which cannot be modified).</p> <ol style="list-style-type: none"> 1. In the Add Chart area, click Use Dashboard Templates. 2. Select a dashboard template and click Next. Select one or more log streams by clicking <input type="checkbox"/> , and click OK.

Step 5 The created dashboard will be displayed in the dashboard list.

- Click **Edit** in the **Operation** column of the dashboard to change the dashboard name and lite mode.
- Click **Move** in the **Operation** column of the dashboard to change the group.
- Click **Delete** in the **Operation** column of the dashboard to delete it.

----End

Creating a Chart for a Dashboard


- Step 1** In the lower part of the dashboard directory, select a dashboard group and click the name of the target dashboard to access its details page.
- Step 2** Click  in the upper right corner. On the displayed page, select a log stream. Then click **Create Chart**.
- Step 3** On the **Add Chart** page displayed, click **Create**, set parameters by referring to [Table 5-13](#), and click **OK**.

Table 5-13 Creating a chart

Parameter	Description
Chart Name	Enter a unique name to differentiate your chart from other charts in the log stream. Enter 1 to 64 characters, including only letters, digits, hyphens (-), underscores (_), spaces, brackets, and periods (.). Do not start or end with a period or space.
SQL Statement	<ul style="list-style-type: none"> The default statement is SELECT *, which is used to query structured log data in the log stream. For details about SQL statements, see SQL Analysis Syntax.
Chart Type	LTS provides various chart types, including tables, bar charts, and line charts.
Add to Dashboard	<ul style="list-style-type: none"> If you enable this option, select one or more dashboards to add the chart to them. If you disable this option, the chart will not be displayed on any dashboard.


- Step 4** Click **OK**.











----End












Related Operations



After a dashboard is created, click the dashboard name to go to the details page. You can edit and remove its chart, and adjust chart positions. For details, see [Table 5-14](#).

Table 5-14 Related operations

Operation	Description
Edit chart	Move the cursor to the upper right corner of a chart, click  , and select Edit from the drop-down list to edit the chart. For details, see Analyzing Logs in LTS .

Operation	Description
Remove chart	Move the cursor to the upper right corner of a chart, click  , select Remove from the drop-down list, and click Save .
Relocate chart	Move the cursor to a chart, drag the chart to the desired location on the dashboard, and click Save .
Resize chart	Move the cursor to the lower right corner of a chart, drag the corner to adjust the chart size, and click Save .
Modify filter	Move the cursor to the upper right corner of a filter, click  , and select Modify from the drop-down list. On the displayed page, edit the filter by referring to Adding a Filter .
Copy filter	Move the cursor to the upper right corner of a filter, click  , and select Copy from the drop-down list. On the displayed page, click OK to copy the filter.
Delete filter	Move the cursor to the upper right corner of a filter, click  , and select Delete from the drop-down list. In the displayed Delete Filter dialog box, click OK to delete the filter.
Resize filter	Move the cursor to the lower right corner of a filter, drag the corner to adjust the filter size, and click Save .
Enable auto refresh	Click  in the upper right corner, and enable auto refresh with a specified interval to refresh the data in all charts. The interval can be 1, 5, and 15 minutes.
Manually refresh	Select a dashboard and click  to manually refresh it.
View dashboard in full screen	Select a dashboard and click  to view the dashboard in full screen. In the full-screen mode, you can select Stay Logged In so that your account will not be logged out after a period of inactivity.
Exit full screen of dashboard	Move the cursor to the upper part of the screen and click  or  , or press Esc on the keyboard.
View chart in full screen	Select a dashboard and click Cancel to exit the editing mode. Move the cursor to the upper right corner of a chart, click  , and select Full Screen from the drop-down list.

Operation	Description
Exit full screen of chart	Move the cursor to the upper part of the screen and click  , or click  and select Exit Full Screen from the drop-down list, or press Esc on the keyboard.
Manually refresh chart	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select Refresh from the drop-down list. Alternatively, in full-screen mode, click  and select Refresh from the drop-down list.
Set time range	<p>Select a dashboard and click the  1 hour(From now) .</p> <p>drop-down list box before .</p> <p>There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.</p> <p>NOTE</p> <ul style="list-style-type: none"> • From now: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31. • From last: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00. • Specified: queries log data that is generated in a specified time range.
View chart details	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select View Details from the drop-down list.
Add alarm rules	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select Add Alarm Rule from the drop-down list.
Copy	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select Copy from the drop-down list.
Copy to other dashboards	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select Copy to Other Dashboards from the drop-down list.

Operation	Description
Copy statement	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select Copy Statement from the drop-down list.
Export chart data	Select a dashboard, move the cursor to the upper right corner of a chart, click  , and select Export Chart Data from the drop-down list.

5.3.2 Adding a Dashboard Filter

Add filters to a dashboard so that you can filter data or replace data with variables.

Filters are used to modify query criteria for all statistical charts in a dashboard in batches. Each statistical chart is actually the results of a query and analysis statement.

- Filter type: Filter logs by log field key and value. To apply a filter, add it as a filtering condition before your query and analysis statement, and use **AND** or **NOT** to link the condition and statement. For example, *Key: Value AND [search query] / [SQL query]* indicates that logs containing *Key:Value* are searched for within the results of the original query and analysis statement. For this filter type, you can select or enter one or more values, which are connected by an **OR** relationship.
- Time series filter type: Labels and values are dynamically added for filtering. When adding a filter, you can add filter criteria, which are logically connected by **AND**.


Prerequisites

- Logs have been collected.
- Charts have been added to a dashboard.
- A log structuring rule has been configured for the log stream. For details, see [Setting Cloud Structuring Parsing](#).

Constraints

Up to 10 filters can be added for a dashboard.


Adding a Filter

- Step 1** Log in to the LTS console, choose **Dashboards** in the navigation pane, and select a dashboard.
- Step 2** Click the dashboard name to go to the details page.
- Step 3** Click . On the **Filter** page, set the filter parameters by referring to [Adding a Dashboard Filter](#).

 **NOTE**

Fields of the numeric type cannot be filtered.

Table 5-15 Filter parameters

Parameter	Description
Filter Name	Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed. It cannot start with a period or underscore, or end with a period.
Operator	Specify the relationship between filter criteria and chart query statements. The value can be AND (default) or NOT .
Key	The field to be filtered. Use only letters, digits, hyphens (-), underscores (_), and periods (.). Do not start with a period or underscore, end with a period, or use only digits.
Key Alias	Alias of the key, which is used to distinguish fields. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed. It cannot start with a period or underscore, or end with a period.
Static Values	<p>Set the value corresponding to the key. You can click Add multiple times to add more values.</p> <p>Set the following parameters for adding a value:</p> <ul style="list-style-type: none"> ● Value: name of a value. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed. It cannot start with a period or underscore, or end with a period. ● Alias: alias of the value. ● Default Value: If you enable this parameter, the value will be set as a default value for the key. ● Operation: Click  to delete the value.

Parameter	Description
Add Dynamic Value	<p>The dynamic values are the query results of the corresponding query and analysis statements. The query results change dynamically in different time ranges.</p> <ul style="list-style-type: none"> • If Add Dynamic Value is disabled, the dynamic change of the query results cannot be set. • Enable this parameter to add a dynamic value for the key, displaying the dynamic changes of the query results. Set the following parameters: <ul style="list-style-type: none"> Log Group: Select the log group to be queried. Log stream: Select the log stream to be queried. Dynamic Value Source: You can choose fuzzy match or SQL query. Fuzzy match: Select structured fields configured for the current log stream. SQL query: Enter a SQL query statement. Click Search to preview the dynamic value.

Step 4 Click **OK**.

----End

5.3.3 Dashboard Templates

5.3.3.1 APIG Dashboard Templates

APIG is your fully managed API hosting service with high performance, availability, and security. With APIG, you can build, manage, and deploy APIs at any scale to package your capabilities and sell them at KooGallery. With just a few clicks, you can integrate your internal systems and selectively expose and monetize your service capabilities with minimal cost and risk. APIG helps you lower R&D costs and improve operational efficiency, freeing you to focus on your core business.

APIG dashboard templates support [Viewing APIG Access Center](#), [Viewing APIG Monitoring Center](#), and [Viewing APIG Monitoring by the Second](#).

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing APIG Access Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **APIG dashboard templates** under **Dashboard Templates** and click **APIG access center** to view the chart details.

- Filter by requested domain name. The associated query and analysis statement is:

```
select distinct(host)
```
- Filter by application ID. The associated query and analysis statement is:

```
select distinct(app_id)
```
- **PV Distribution (Global).** The associated query and analysis statement is:

```
SELECT ip_to_country(my_remote_addr) as country,sum(ori_pv) as PV from (select my_remote_addr, count(1) as ori_pv group by my_remote_addr ORDER BY ori_pv desc LIMIT 10000) GROUP BY country HAVING country not in ('Reserved address','*')
```
- **Average Latency Distribution (China).** The associated query and analysis statement is:

```
SELECT province,round( CASE WHEN "Average latency (ms)" > 0 THEN "Average latency (ms)" ELSE 0 END, 3 ) AS "Average latency (ms)"FROM (SELECT ip_to_province(my_remote_addr) as province,sum(rt)/sum(ori_pv) * 1000 AS "Average latency (ms)" from (select my_remote_addr, sum(request_time) as rt,count(1) as ori_pv group by my_remote_addr ORDER BY ori_pv desc LIMIT 10000) WHERE IP_TO_COUNTRY (my_remote_addr) = 'China' GROUP BY province ) where province not in ('Reserved address','*')
```
- **Average Latency Distribution (Global).** The associated query and analysis statement is:

```
SELECT country,round( CASE WHEN "Average latency (ms)" > 0 THEN "Average latency (ms)" ELSE 0 END, 2 ) AS "Average latency (ms)"FROM (SELECT ip_to_country(my_remote_addr) as country,sum(rt)/sum(ori_pv) * 1000 AS "Average latency (ms)" from (select my_remote_addr, sum(request_time) as rt,count(1) as ori_pv group by my_remote_addr ORDER BY ori_pv desc LIMIT 10000) GROUP BY country ) where country not in ('Reserved address','*')
```
- **PV/UV Today.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss' ) as _time_PV,UV FROM (select TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT600S') AS _time_ , count(1) as PV, APPROX_COUNT_DISTINCT(my_remote_addr) as UV from log WHERE __time <= CURRENT_TIMESTAMP and __time >= DATE_TRUNC( 'DAY',(CURRENT_TIMESTAMP + INTERVAL '8' HOUR)) - INTERVAL '8' HOUR group by _time_ order by _time_)
```
- **Top 10 Provinces by Visits.** The associated query and analysis statement is:

```
select ip_to_province(my_remote_addr) as "province", sum(ori_pv) as "Visits" from(select my_remote_addr, count(1) as ori_pv group by my_remote_addr ORDER BY ori_pv desc LIMIT 10000)group by "province" HAVING "province" <> '-1' order by "Visits" desc limit 10
```
- **Top 10 Cities by Visits.** The associated query and analysis statement is:

```
select ip_to_city(my_remote_addr) as "city", sum(ori_pv) as "Visits" from(select my_remote_addr, count(1) as ori_pv group by my_remote_addr ORDER BY ori_pv desc LIMIT 10000) group by "city" HAVING "city" <> '-1' order by "Visits" desc limit 10
```
- **Top 10 Hosts by Visits.** The associated query and analysis statement is:

```
select host as "Host", count(1) as "PV" group by "Host" order by "PV" desc limit 10
```
- **Top 10 UserAgents by Visits.** The associated query and analysis statement is:

```
select http_user_agent as "UserAgent", count(1) as "PV" group by "UserAgent" order by "PV" desc limit 10
```
- **Device Distribution by Type.** The associated query and analysis statement is:

```
select case when regexp_like(lower(http_user_agent), 'iphone|ipod|android|ios') then 'Mobile' else 'PC' end as type , count(1) as total group by type
```
- **Device Distribution by System.** The associated query and analysis statement is:

```
select case when regexp_like(lower(http_user_agent), 'iphone|ipod|ios') then 'IOS' when
regexp_like(lower(http_user_agent), 'android') then 'Android' else 'other' end as type , count(1) as
total group by type HAVING type != 'other'
```

- **TOP URL.** The associated query and analysis statement is:
select router_uri , count(1) as pv, APPROX_COUNT_DISTINCT(my_remote_addr) as UV,
round(sum(case when status < 400 then 1 else 0 end) * 100.0 / count(1), 2) as "Access Success
Rate" group by router_uri ORDER by pv desc
- **Top IP Addresses by Visits.** The associated query and analysis statement is:
select my_remote_addr as "Source IP Address",ip_to_country(my_remote_addr) as "Country/
Region",ip_to_province(my_remote_addr) as "Province",ip_to_city(my_remote_addr) as
"City",ip_to_provider(my_remote_addr) as "Carrier",count(1) as "PV" group by my_remote_addr
ORDER by "PV" desc limit 100

----End

Viewing APIG Monitoring Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **APIG dashboard templates** under **Dashboard Templates** and click **APIG monitoring center** to view the chart details.

- Filter by requested domain name. The associated query and analysis statement is:
select distinct(host)
- Filter by application ID. The associated query and analysis statement is:
select distinct(app_id)
- **PV.** The associated query and analysis statement is:
SELECT TIME_FORMAT(_time_ , 'yyyy-MM-dd HH:mm:ss') as _time_ , PV FROM (SELECT TIME_CEIL
(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ') , 'PT300S') AS _time_ , count(1) AS PV
FROM log GROUP BY _time_)
- **Request Success Rate.** The associated query and analysis statement is:
select ROUND(sum(case when status < 400 then 1 else 0 end) * 100.0 / count(1),2) as cnt
- **Average Latency.** The associated query and analysis statement is:
select round(avg(request_time) * 1000, 3) as cnt
- **4xx Requests.** The associated query and analysis statement is:
SELECT COUNT(1) as cnt WHERE "status" >= 400 and "status" < 500
- **404 Requests.** The associated query and analysis statement is:
SELECT COUNT(1) as cnt WHERE "status" = 404
- **429 Requests.** The associated query and analysis statement is:
SELECT COUNT(1) as cnt WHERE "status" = 429
- **504 Requests.** The associated query and analysis statement is:
SELECT COUNT(1) as cnt WHERE "status" = 504
- **5xx Requests.** The associated query and analysis statement is:
SELECT TIME_FORMAT(_time_ , 'yyyy-MM-dd HH:mm:ss') as _time_ , cnt FROM (SELECT TIME_CEIL
(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ') , 'PT300S') AS _time_ , count(1) AS cnt
FROM log where "status" >= 500 GROUP BY _time_)
- **Status Code Distribution.** The associated query and analysis statement is:
SELECT status, COUNT(1) AS rm GROUP BY status
- **UV.** The associated query and analysis statement is:
SELECT TIME_FORMAT(_time_ , 'yyyy-MM-dd HH:mm:ss') as _time_ , UV FROM (select
TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ') , 'PT600S') AS _time_ ,
APPROX_COUNT_DISTINCT(my_remote_addr) as UV from log group by _time_)
- **Traffic.** The associated query and analysis statement is:
select TIME_FORMAT(_time_ , 'yyyy-MM-dd HH:mm:ss') AS _time_ , round(CASE WHEN "Inbound" > 0
THEN "Inbound" ELSE 0 END, 2) AS "Inbound" , round(CASE WHEN "Outbound" > 0 THEN
"Outbound" ELSE 0 END, 2) AS "Outbound" FROM (SELECT TIME_CEIL(TIME_PARSE(time_local,

```
'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT600S') AS _time_,sum(request_length) / 1024.0 AS
"Inbound",sum(bytes_sent) / 1024.0 AS "Outbound" group by _time_)
```

- Access Failure Rate.** The associated query and analysis statement is:


```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss') as _time_,round( CASE WHEN "Failure rate"
> 0 THEN "Failure rate" ELSE 0 END, 2 ) AS "Failure rate",round( CASE WHEN "5xx Requests" > 0
THEN "5xx Requests" ELSE 0 END, 2 ) AS "5xx Requests" from (select
TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT600S') AS _time_,sum(case
when status >= 400 then 1 else 0 end) * 100.0 / count(1) as 'Failure rate' , sum(case when status
>=500 THEN 1 ELSE 0 END)*100.0/COUNT(1) as '5xx Requests' group by _time_)
```
- Latency.** The associated query and analysis statement is:


```
select TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss') as _time_,round( CASE WHEN "Avg." > 0
THEN "Avg." ELSE 0 END, 2 ) AS "Avg.",round( CASE WHEN "P50" > 0 THEN "P50" ELSE 0 END, 2 )
AS "P50",round( CASE WHEN "P90" > 0 THEN "P90" ELSE 0 END, 2 ) AS "P90",round( CASE WHEN
"P99" > 0 THEN "P99" ELSE 0 END, 2 ) AS "P99",round( CASE WHEN "P9999" > 0 THEN "P9999" ELSE
0 END, 2 ) AS "P9999" from (select TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss
ZZ'),'PT600S') as _time_,avg(request_time) * 1000 as "Avg.", APPROX_QUANTILE_DS("request_time",
0.50)*1000 as "P50", APPROX_QUANTILE_DS("request_time", 0.90)*1000 as
"P90",_APPROX_QUANTILE_DS("request_time", 0.99)*1000 as
"P99",APPROX_QUANTILE_DS("request_time", 0.9999)*1000 as 'P9999' group by _time_)
```
- Top Host Requests.** The associated query and analysis statement is:


```
SELECT "host", pv, uv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate
(%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN
"Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN
"Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT
"host", count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( my_remote_addr ) AS uv, sum( CASE WHEN
"status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)",
avg( request_time ) * 1000 AS "Average Latency (ms)", sum( request_length ) / 1024.0 AS "Inbound
(KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "host" != " " GROUP BY "host" )
ORDER BY pv DESC
```
- Top Host Latencies.** The associated query and analysis statement is:


```
SELECT "host", pv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate
(%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90
Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE
WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)"
FROM ( SELECT "host", count( 1 ) AS pv, sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END ) *
100.0 / count( 1 ) AS "Access Success Rate (%)", avg( request_time ) * 1000 AS "Average Latency
(ms)",APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",
APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " "
GROUP BY "host" ) ORDER BY "Average Latency (ms)" desc
```
- Top Host Failure Rates.** The associated query and analysis statement is:


```
SELECT "host", pv,round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)"
ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN
"Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90
Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE
WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)"
FROM ( SELECT "host", count( 1 ) AS pv, sum( CASE WHEN "status" >= 400 THEN 1 ELSE 0 END ) *
100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( request_time ) * 1000 AS "Average Latency
(ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",
APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " "
GROUP BY "host" ) ORDER BY "Access Failure Rate (%)" desc
```
- Top URL Requests.** The associated query and analysis statement is:


```
SELECT upstream_uri, pv,uv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success
Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)"
> 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN
"Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN
"Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT
upstream_uri, count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( my_remote_addr ) AS uv, sum( CASE
WHEN "status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)",
avg( request_time ) * 1000 AS "Average Latency (ms)", sum( request_length ) / 1024.0 AS "Inbound
(KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "host" != " " GROUP BY
upstream_uri ) ORDER BY pv desc
```
- Top URL Failure Rates.** The associated query and analysis statement is:

```
SELECT upstream_uri, pv, round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)" ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM( SELECT upstream_uri, count( 1 ) AS pv, sum( CASE WHEN "status" >= 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( request_time ) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " GROUP BY upstream_uri ) ORDER BY "Access Failure Rate (%)" desc
```

- Top Backend Requests.** The associated query and analysis statement is:


```
SELECT addr, pv, uv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN "Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT my_remote_addr as addr, count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( my_remote_addr ) AS uv, sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)", avg( request_time ) * 1000 AS "Average Latency (ms)", sum( request_length ) / 1024.0 AS "Inbound (KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "host" != " GROUP BY addr having length(my_remote_addr) > 2) ORDER BY "pv" desc
```
- Top Backend Latencies.** The associated query and analysis statement is:


```
SELECT addr,pv,round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)",round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)",round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)",round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM (SELECT my_remote_addr as addr,count( 1 ) AS pv,sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)",avg( request_time ) * 1000 AS "Average Latency (ms)",APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " and "my_remote_addr" != '-' GROUP BY addr ) ORDER BY "Average Latency (ms)" desc
```
- Top Backend Failure Rates.** The associated query and analysis statement is:


```
SELECT addr, pv, round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)" ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM ( SELECT my_remote_addr as addr, count( 1 ) AS pv, sum( CASE WHEN "status" >= 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( request_time ) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " and "my_remote_addr" != '-' GROUP BY addr) ORDER BY "Access Failure Rate (%)" desc
```
- Top URL Latencies.** The associated query and analysis statement is:


```
SELECT upstream_uri, pv,round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)",round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)",round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)",round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM (SELECT upstream_uri, count( 1 ) AS pv, sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)", avg( request_time ) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " GROUP BY upstream_uri ) ORDER BY "Average Latency (ms)" desc
```

----End

Viewing APIG Monitoring by the Second

- Step 1** Log in to the LTS console. In the navigation pane, choose **Dashboards**.
- Step 2** Choose **APIG dashboard templates** under **Dashboard Templates** and click **APIG monitoring by the second** to view the chart details.
 - Filter by requested domain name. The associated query and analysis statement is:


```
select distinct(host)
```

- **Filter by application ID.** The associated query and analysis statement is:

```
select distinct(app_id)
```
- **QPS.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT(TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT1S'),'yyyy-MM-dd HH:mm:ss') AS __time_, COUNT(*) as QPS from log group by __time_
```
- **Success Rate.** The associated query and analysis statement is:

```
select __time,round(CASE WHEN "Success rate" > 0 THEN "Success rate" else 0 end,2) as "Success rate" from (select TIME_FORMAT(TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT5S'),'yyyy-MM-dd HH:mm:ss') as __time, sum(case when status < 400 then 1 else 0 end) * 100.0 / count(1) as 'Success rate' from log group by __time)
```
- **Latency.** The associated query and analysis statement is:

```
select __time,round(CASE WHEN "Access latency" > 0 THEN "Access latency" else 0 end,2) as "Access latency",round(CASE WHEN "Upstream latency" > 0 THEN "Upstream latency" else 0 end,2) as "Upstream latency" from (select TIME_FORMAT(TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT5S'),'yyyy-MM-dd HH:mm:ss') as __time, avg(request_time)* 1000 as 'Access latency',avg(upstream_response_time)* 1000 as 'Upstream latency' from log group by __time)
```
- **Traffic.** The associated query and analysis statement is:

```
select __time,round( CASE WHEN "Incoming" > 0 THEN "Incoming" ELSE 0 END, 3 ) AS "Incoming",round( CASE WHEN "Outgoing body" > 0 THEN "Outgoing body" ELSE 0 END, 3 ) AS "Outgoing body" from (select TIME_FORMAT(TIME_CEIL(TIME_PARSE(time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ'),'PT5S'),'yyyy-MM-dd HH:mm:ss') as __time , sum("request_length") / 1024.0 as "Incoming", sum("body_bytes_sent") / 1024.0 as "Outgoing body" group by __time)
```
- **Status Codes.** The associated query and analysis statement is:

```
SELECT TIME_CEIL ( TIME_PARSE ( time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ' ), 'PT5S' ) AS "time", SUM( CASE WHEN "status" >= 200 AND "status" < 300 THEN 1 ELSE 0 END ) AS "2XX", SUM( CASE WHEN "status" >= 300 AND "status" < 400 THEN 1 ELSE 0 END ) AS "3XX", SUM( CASE WHEN "status" >= 400 AND "status" < 500 THEN 1 ELSE 0 END ) AS "4XX", SUM( CASE WHEN "status" >= 500 AND "status" < 600 THEN 1 ELSE 0 END ) AS "5XX", SUM( CASE WHEN "status" < 200 OR "status" >= 600 THEN 1 ELSE 0 END ) AS "Other" FROM log WHERE TIME_PARSE ( time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ' ) IS NOT NULL GROUP BY "time" ORDER BY "time" ASC LIMIT 100000
```
- **Backend Response Codes.** The associated query and analysis statement is:

```
SELECT TIME_CEIL ( TIME_PARSE ( time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ' ), 'PT5S' ) AS "time", SUM( CASE WHEN "upstream_status" >= 200 AND "upstream_status" < 300 THEN 1 ELSE 0 END ) AS "2XX", SUM( CASE WHEN "upstream_status" >= 300 AND "upstream_status" < 400 THEN 1 ELSE 0 END ) AS "3XX", SUM( CASE WHEN "upstream_status" >= 400 AND "upstream_status" < 500 THEN 1 ELSE 0 END ) AS "4XX", SUM( CASE WHEN "upstream_status" >= 500 AND "upstream_status" < 600 THEN 1 ELSE 0 END ) AS "5XX", SUM( CASE WHEN "upstream_status" < 200 OR "upstream_status" >= 600 THEN 1 ELSE 0 END ) AS "Other" FROM log WHERE TIME_PARSE ( time_local, 'dd/MMM/yyyy:HH:mm:ss ZZ' ) IS NOT NULL GROUP BY "time" ORDER BY "time" ASC LIMIT 100000
```

----End

5.3.3.2 CCE Dashboard Templates

CCE provides highly scalable, high-performance, enterprise-class Kubernetes clusters.

CCE dashboard templates support [Viewing CCE Logs: Node Operations](#), [Viewing CCE Logs Kubernetes Object Operations](#), [Viewing CCE Logs Kubernetes Event Queries](#), [Viewing CCE Logs Kubernetes Event Center](#), [Viewing CCE Logs Aggregation Search](#), [Viewing CCE Logs Account Operations](#), and [Viewing CCE Audit Log Center](#).

Prerequisites

- Logs have been collected from CCE. For details, see [Ingesting CCE Application Logs to LTS](#).
- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing CCE Logs: Node Operations

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Logs: Node Operations** to view the chart details.

- Filter by node name. The associated query and analysis statement is:
`select distinct("objectRef.name")`
- Filter by user. The associated query and analysis statement is:
`select distinct("user.username")`
- Filter by status code. The associated query and analysis statement is:
`select distinct("responseStatus.code")`
- Filter by operation type. The associated query and analysis statement is:
`select distinct("verb")`
- **Node Quantity Trend.** The associated query and analysis statement is:
`SELECT time_series(TIME_PARSE(LEFT(requestReceivedTimestamp, 23),yyyy-MM-dd"T"HH:mm:ss.SSS'), 'PT1H', 'yyyy-MM-dd HH', '0') as "dt", count(DISTINCT("objectRef.name")) as "Nodes" where "objectRef.resource" = 'nodes' and "objectRef.subresource" = 'status' and "verb" in ('update', 'patch') and "user.username" = 'system:node' group by "dt" order by "dt" desc limit 10000`
- **Non-System User Operation Trend.** The associated query and analysis statement is:
`SELECT time_series(TIME_PARSE(LEFT(requestReceivedTimestamp, 23),yyyy-MM-dd"T"HH:mm:ss.SSS'), 'PT1H', 'yyyy-MM-dd HH', '0') as "dt", count(*) as "Requests", "user.username" where "objectRef.resource" = 'nodes' and "user.username" not in ('kube-controller-manager', 'kube-apiserver-kubelet-client', 'apiserver') and "user.username" not like 'system:%' and "verb" in ('create', 'delete', 'update', 'patch') group by "dt", "user.username" order by "dt", "Requests" desc limit 10000`
- **create Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.resource" = 'nodes' and "verb" = 'create' group by "Status code"`
- **delete Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.resource" = 'nodes' and "verb" = 'delete' group by "Status code"`
- **patch Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.resource" = 'nodes' and "verb" = 'patch' group by "Status code"`
- **update Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.resource" = 'nodes' and "verb" = 'update' group by "Status code"`
- **Node Blocking/Unblocking Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "requestObject" in ('{"spec":{"unschedulable":false}}', '{"spec":{"unschedulable":true}}') group by "Status code"`
- **Label Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.resource" = 'nodes' and "verb" in ('patch', 'update') and "requestObject" = 'labels' and "requestObject" = 'metadata' group by "Status code"`
- **Taint Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.resource" = 'nodes' and "verb" in ('patch', 'update') and "requestObject" = 'taints' group by "Status code"`
- **Eviction Codes.** The associated query and analysis statement is:
`select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "objectRef.subresource" = 'eviction' and "objectRef.resource" = 'pods' and "verb" = 'create' group by "Status code"`
- **Node Addition/Deletions.** The associated query and analysis statement is:

```
select "auditID" AS "Audit ID", "objectRef.name" AS "Node", "verb" AS "Operation",  
"stageTimestamp" AS "Occurred", "user.username" AS "Operator", "responseStatus.code" AS "Status  
Code" where "objectRef.resource" = 'nodes' and "verb" in ('create','delete')
```

- **Taint Operations.** The associated query and analysis statement is:

```
select "auditID" AS "Audit ID", "objectRef.name" AS "Node", "requestObject" AS "Taints",  
"requestReceivedTimestamp" AS "Occurred", "user.username" AS "Operator", "responseStatus.code"  
AS "Status Code" where "objectRef.resource" = 'nodes' and "verb" = 'patch' and "requestObject" =  
'taints'
```
- **Eviction Operations.** The associated query and analysis statement is:

```
select "auditID" AS "Audit ID", "objectRef.name" AS "pod", "sourceIPs" AS "Source IP",  
"requestReceivedTimestamp" AS "Occurred", "user.username" AS "Operator", "responseStatus.code"  
AS "Status Code" where "objectRef.resource" = 'pods' and "verb" = 'create' and  
"objectRef.subresource" = 'eviction'
```
- **Label Operations.** The associated query and analysis statement is:

```
select "auditID" AS "Audit ID", "objectRef.name" AS "Node", "requestObject" AS "Label",  
"requestReceivedTimestamp" AS "Occurred", "user.username" AS "Operator", "responseStatus.code"  
AS "Status Code" where "objectRef.resource" = 'nodes' and "verb" = 'patch' and "requestObject" =  
'labels'
```
- **Blocking Operations.** The associated query and analysis statement is:

```
select "auditID" AS "Audit ID", "objectRef.name" AS "Node", "requestReceivedTimestamp" AS  
"Occurred", "user.username" AS "Operator", "responseStatus.code" AS "Status Code" where "verb" =  
'patch' and "objectRef.resource" = 'nodes' and "requestObject" = 'true' and "requestObject" =  
'unschedulable'
```
- **Unblocking Operations.** The associated query and analysis statement is:

```
select "auditID" AS "Audit ID", "objectRef.name" AS "Node", "requestReceivedTimestamp" AS  
"Occurred", "user.username" AS "Operator", "responseStatus.code" AS "Status Code" where "verb" =  
'patch' and "objectRef.resource" = 'nodes' and "requestObject" not in ('true','taints','unschedulable')
```

----End

Viewing CCE Logs Kubernetes Object Operations

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Logs Kubernetes Object Operations** to view the chart details.

- Filter by namespace. The associated query and analysis statement is:

```
select distinct("objectRef.namespace")
```
- Filter by operation type. The associated query and analysis statement is:

```
select distinct("verb")
```
- Filter by status code. The associated query and analysis statement is:

```
select distinct("responseStatus.code")
```
- Filter by resource object. The associated query and analysis statement is:

```
select distinct("objectRef.name")
```
- Filter by resource type. The associated query and analysis statement is:

```
select distinct("objectRef.resource")
```
- Filter by user. The associated query and analysis statement is:

```
select distinct("user.username")
```
- **Major Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(requestReceivedTimestamp, 16),'T',' ') AS "dt", "verb" as "Operation Type",  
count(*) as "count" where "verb" in ('create','delete','update','patch') and "objectRef.resource" in  
( 'deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config  
maps','persistentvolumeclaims') group by "dt","Operation Type" order by "dt" limit 10000
```
- **Non-System User Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(requestReceivedTimestamp, 16),'T',' ') AS "dt", count(*) as "Requests  
", "user.username" WHERE "user.username" not in ('kube-controller-manager','kube-apiserver-kubelet-  
client','apiserver') and "user.username" not like 'system:%' and "verb" in
```

```
('create','delete','update','patch') and "objectRef.resource" in  
( 'deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','secrets',  
'pvcs') group by "dt", "user.username" limit 10000
```

- **create Resources.** The associated query and analysis statement is:
select "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'create' and
"objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "objectRef.resource"
- **delete Resources.** The associated query and analysis statement is:
select "objectRef.resource" as "Resource Type", count(*) as "count" where "verb" = 'delete' and
"objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "objectRef.resource"
- **update Resources.** The associated query and analysis statement is:
select "objectRef.resource" as "Resource Type", count(*) as "count" where "verb" = 'update' and
"objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "objectRef.resource"
- **patch Resources.** The associated query and analysis statement is:
select "objectRef.resource" as "Resource Type", count(*) as "count" where "verb" = 'patch' and
"objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "objectRef.resource"
- **create Users.** The associated query and analysis statement is:
select "user.username" as "User", count(*) as "count" where "verb" = 'create' and "objectRef.resource"
in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "user.username"
- **delete Users.** The associated query and analysis statement is:
select "user.username" as "User", count(*) as "count" where "verb" = 'delete' and "objectRef.resource"
in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "user.username"
- **update Users.** The associated query and analysis statement is:
select "user.username" as "User", count(*) as "count" where "verb" = 'update' and
"objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "user.username"select "user.username" as "User", count(*)
as "count" where "verb" = 'update' and "objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "user.username"
- **patch Users.** The associated query and analysis statement is:
select "user.username" as "User", count(*) as "count" where "verb" = 'patch' and
"objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "user.username"
- **create Codes.** The associated query and analysis statement is:
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" =
'create' and "objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "responseStatus.code"
- **delete Codes.** The associated query and analysis statement is:
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" =
'delete' and "objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "responseStatus.code"
- **update Codes.** The associated query and analysis statement is:
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" =
'update' and "objectRef.resource" in
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config
maps','persistentvolumeclaims') group by "responseStatus.code"

- **patch Codes.** The associated query and analysis statement is:

```
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" = 'patch' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by "responseStatus.code"
```
- **create Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'create' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
```
- **delete Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'delete' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
```
- **update Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'update' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
```
- **patch Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'patch' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
```

----End

Viewing CCE Logs Kubernetes Event Queries

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Logs Kubernetes Event Queries** to view the chart details.

- **Filter by namespace.** The associated query and analysis statement is:

```
select distinct("objectRef.namespace")
```
- **Filter by operation type.** The associated query and analysis statement is:

```
select distinct("verb")
```
- **Filter by status code.** The associated query and analysis statement is:

```
select distinct("responseStatus.code")
```
- **Filter by resource object.** The associated query and analysis statement is:

```
select distinct("objectRef.name")
```
- **Filter by resource type.** The associated query and analysis statement is:

```
select distinct("objectRef.resource")
```
- **Filter by user.** The associated query and analysis statement is:

```
select distinct("user.username")
```
- **Major Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(requestReceivedTimestamp, 16),'T',' ') AS "dt", "verb" as "Operation Type", count(*) as "count" where "verb" in ('create','delete','update','patch') and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by "dt","Operation Type" order by "dt" limit 10000
```
- **Non-System User Operation Trend.** The associated query and analysis statement is:

```
SELECT REPLACE(LEFT(requestReceivedTimestamp, 16),'T',' ') AS "dt", count(*) as "Requests", "user.username" WHERE "user.username" not in ('kube-controller-manager','kube-apiserver-kubelet-client','apiserver') and "user.username" not like 'system:%' and "verb" in ('create','delete','update','patch') and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','secrets','pvcs') group by "dt", "user.username" limit 10000
```

- **create Resources.** The associated query and analysis statement is:

```
select "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'create' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "objectRef.resource"
```
- **delete Resources.** The associated query and analysis statement is:

```
select "objectRef.resource" as "Resource Type", count(*) as "count" where "verb" = 'delete' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "objectRef.resource"
```
- **update Resources.** The associated query and analysis statement is:

```
select "objectRef.resource" as "Resource Type", count(*) as "count" where "verb" = 'update' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "objectRef.resource"
```
- **patch Resources.** The associated query and analysis statement is:

```
select "objectRef.resource" as "Resource Type", count(*) as "count" where "verb" = 'patch' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "objectRef.resource"
```
- **create Users.** The associated query and analysis statement is:

```
select "user.username" as "User", count(*) as "count" where "verb" = 'create' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "user.username"
```
- **delete Users.** The associated query and analysis statement is:

```
select "user.username" as "User", count(*) as "count" where "verb" = 'delete' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "user.username"
```
- **update Users.** The associated query and analysis statement is:

```
select "user.username" as "User", count(*) as "count" where "verb" = 'update' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "user.username"select "user.username" as "User", count(*) as "count" where "verb" = 'update' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "user.username"
```
- **patch Users.** The associated query and analysis statement is:

```
select "user.username" as "User", count(*) as "count" where "verb" = 'patch' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "user.username"
```
- **create Codes.** The associated query and analysis statement is:

```
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" = 'create' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "responseStatus.code"
```
- **delete Codes.** The associated query and analysis statement is:

```
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" = 'delete' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "responseStatus.code"
```
- **update Codes.** The associated query and analysis statement is:

```
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" = 'update' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','configmaps','persistentvolumeclaims') group by "responseStatus.code"
```
- **patch Codes.** The associated query and analysis statement is:

```
select cast("responseStatus.code" as varchar) as "Status code", count(*) as "count" where "verb" = 'patch' and "objectRef.resource" in
```

```
('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by "responseStatus.code"
```

- **create Operation Trend.** The associated query and analysis statement is:
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'create' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
- **delete Operation Trend.** The associated query and analysis statement is:
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'delete' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
- **update Operation Trend.** The associated query and analysis statement is:
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'update' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000
- **patch Operation Trend.** The associated query and analysis statement is:
SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" where "verb" = 'patch' and "objectRef.resource" in ('deployments','statefulsets','cronjobs','daemonsets','jobs','pods','services','ingresses','configmaps','config maps','persistentvolumeclaims') group by dt, "objectRef.resource" order by dt limit 10000

----End

Viewing CCE Logs Kubernetes Event Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Logs Kubernetes Event Center** to view the chart details.

- **Event Severity:** The options are **Warning** and **Normal**.
- **Filter by event type.** The associated query and analysis statement is:
select distinct("name")
- **Filter by cluster ID.** The associated query and analysis statement is:
select distinct("cluster_id")
- **Filter by namespace.** The associated query and analysis statement is:
select distinct("namespace")
- **Filter by name.** The associated query and analysis statement is:
select distinct("resource_name")
- **Contrack Full.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100 , 2) as "inc" from (select compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'ContrackFull'))
- **Event Sync Error.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100 , 2) as "inc" from (select compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'NTPIsDown'))
- **Insufficient Node PIDs.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100 , 2) as "inc" from (select compare("total", 3600) as diff from(select count(1) as "total" from log where "name" in ('PIDPressure','NodeHasPIDPressure')))
- **Insufficient Node FDs.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100 , 2) as "inc" from (select compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'NodeHasFDPressure'))
- **Insufficient Node Disk Space.** The associated query and analysis statement is:

```
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare( "total", 3600) as diff from( select count(1) as "total" from log where "name"=
'NodeHasDiskPressure' ) )
```

- **Pod OOM.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare("total", 3600) as diff from(select count(1) as "total" from log where "reason" in
('OOMKilling','PodOOMKilling'))
- **DockerHung.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'Failed' and
"reason" = 'DockerHung'))
- **Node Restart.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'NodeRebooted'))
- **Pull Image Failed.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'Failed' and
"reason" = 'ImagePullBackOff'))
- **Node OOM.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare("total", 3600) as diff from(select count(1) as "total" from log where "name" = 'SystemOOM'))
- **Start Pod Failed.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select
compare("total", 3600) as diff from(select count(1) as "total" from log where "name"= 'Failed' and
"resource_kind" = 'Pod' and "reason" = 'ImagePullBackOff'))
- **Event Distribution.** The associated query and analysis statement is:
select "type", count(*) as "Events" group by "type"
- **Warning Event Trend.** The associated query and analysis statement is:
select time_series(__time, 'PT1H', 'yyyy-MM-dd HH', '0') as "dt",count(1) as "count" from log where
"type" = 'Warning' group by "dt" order by "dt"
- **Error Event Trend.** The associated query and analysis statement is:
select time_series(__time, 'PT1H', 'yyyy-MM-dd HH', '0') as "dt",count(1) as "count" from log where
"type" = 'Error' group by "dt" order by "dt"
- **Pod OOM Events.** The associated query and analysis statement is:
select TIME_FORMAT(__time, 'yyyy-MM-dd HH:mm:ss', '+08:00') as "Time", "resource_kind" as "Event
Target", "name" as "Type", "resource_name" as "Target Name", "reason" as "Details" from log where
"name" in ('OOMKilling','PodOOMKilling') order by __time desc limit 100
- **Pod Drive Event List.** The associated query and analysis statement is:
select TIME_FORMAT(__time, 'yyyy-MM-dd HH:mm:ss', '+08:00') as "Time", "resource_kind" as
"Event Target", "name" as "Type", "resource_name" as "Target Name", "reason" as "Details" from log
where "name" = 'NodeControllerEviction' order by __time desc limit 100
- **Major Events.** The associated query and analysis statement is:
select TIME_FORMAT(__time, 'yyyy-MM-dd HH:mm:ss', '+08:00') as "Time", "type" as "Level",
"resource_kind" as "Event Target", "name" as "Type", "resource_name" as "Target Name", "reason" as
"Details" from log where "type" in ('Warning','Error') order by __time desc limit 100

----End

Viewing CCE Logs Aggregation Search

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Logs Aggregation Search** to view the chart details.

- **Filter by namespace.** The associated query and analysis statement is:
select distinct("objectRef.namespace")

- **Filter by user.** The associated query and analysis statement is:
`select distinct("user.username")`
- **Filter by status code.** The associated query and analysis statement is:
`select distinct("responseStatus.code")`
- **Filter by operation type.** The associated query and analysis statement is:
`select distinct("verb")`
- **Filter by resource object.** The associated query and analysis statement is:
`select distinct("objectRef.name")`
- **Filter by resource type.** The associated query and analysis statement is:
`select distinct("objectRef.resource")`
- **Filter by request URL.** The associated query and analysis statement is:
`select distinct("requestURI")`
- **Filter by userAgent.** The associated query and analysis statement is:
`select distinct("userAgent")`
- **User Distribution Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "user.username" as "User", count(*) as "count" group by dt, "user.username" order by dt limit 10000`
- **Namespace Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.namespace" as "Namespace", count(*) as "count" group by dt, "objectRef.namespace" order by dt limit 10000`
- **Operation Type Distribution Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.namespace" as "Namespace", count(*) as "count" group by dt, "objectRef.namespace" order by dt limit 10000`
- **Status Code Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, cast("responseStatus.code" as varchar) as "Return code", count(*) as "count" group by dt, "Return code" order by dt limit 10000`
- **Resource Type Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" group by dt, "objectRef.resource" order by dt limit 10000 SELECT REPLACE(LEFT(stageTimestamp, 16),'T',' ') AS dt, "objectRef.resource" as "Resource type", count(*) as "count" group by dt, "objectRef.resource" order by dt limit 10000`
- **Major Operations.** The associated query and analysis statement is:
`select "auditID" AS "Audit ID", "verb" AS "Operation Type", "requestReceivedTimestamp" AS "Started", "stageTimestamp" AS "Ended", "user.username" AS "Operator", "sourceIPs" AS "Operation Source", "userAgent", "objectRef.namespace" AS "Namespace", CONCAT(CONCAT("objectRef.resource", '/'), "objectRef.subresource") AS "Operation Object", "objectRef.name" AS "Resource Name", "responseStatus.code" AS "Return Code"`

----End

Viewing CCE Logs Account Operations

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Logs Account Operations** to view the chart details.

- **Filter by username.** The associated query and analysis statement is:
`select distinct("user.username")`
- **Filter by namespace.** The associated query and analysis statement is:
`select distinct("objectRef.namespace")`
- **Filter by status code.** The associated query and analysis statement is:
`select distinct("responseStatus.code")`
- **Resources Created.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from (select count(1) as total from log where "verb" = 'create'))`

- **Resources Modified.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100 , 2) as "inc" from (select compare(total , 86400) as diff from(select count(*) as "total" from log where "verb" in ('update','patch')))`
- **Resources Deleted.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100 , 2) as "inc" from (select compare(total , 86400) as diff from(select count(*) as "total" from log where "verb" = 'delete'))`
- **Affected Namespaces.** The associated query and analysis statement is:
`select case when "objectRef.namespace" is null then '_all_' else "objectRef.namespace" end as ns, count(1) as total group by ns limit 10000`
- **Deleted Resource.** The associated query and analysis statement is:
`SELECT "objectRef.resource" as "resource", count(1) as "count" where "verb" = 'delete' group by "resource"`
- **Operation History.** The associated query and analysis statement is:
`select case when "Operation" is null then 'None' else "Operation" end as "Operation", "Time", v from (select concat(CASE WHEN "objectRef.subresource" is null then "objectRef.resource" else "objectRef.subresource" end, '[', verb, ']') as "Operation", time_series(__time, 'PT1H', 'yyyy-MM-dd HH', '0') as "Time", count(1) as v from log where "verb" in ('create', 'patch', 'update', 'delete') group by "Operation", "Time" order by "Time" desc limit 10000)`
- **Resource Operations.** The associated query and analysis statement is:
`select CASE WHEN "objectRef.subresource" is null then "objectRef.resource" else "objectRef.subresource" end as "Resource", verb as "Operation", count(1) as total where "verb" in ('create','update','patch','delete') group by "Resource", "Operation" limit 10000`
- **Created Resources.** The associated query and analysis statement is:
`SELECT "auditID" as "Event ID", time_format("__time", 'yyyy-MM-dd HH:mm:ss') as "Occurred", "requestURI" as "Resource", "objectRef.name" as "Resource Name", "responseStatus.code" as "Status Code", "sourceIPs" as "Source IP", "requestObject" as "Details" where "verb" = 'create' order by __time desc limit 1000`
- **Modified Resources.** The associated query and analysis statement is:
`SELECT auditID as "Event ID", time_format("__time", 'yyyy-MM-dd HH:mm:ss') as "Occurred", "requestURI" as "Resource", "objectRef.name" as "Resource Name", "responseStatus.code" as "Status Code", "sourceIPs" as "Source IP", requestObject as "Details" where "verb" in ('update','patch') order by __time desc limit 1000`
- **Accessed Resources.** The associated query and analysis statement is:
`SELECT auditID as "Event ID", time_format("__time", 'yyyy-MM-dd HH:mm:ss') as "Occurred", "requestURI" as "Resource", "objectRef.name" as "Resource Name", "responseStatus.code" as "Status Code", "sourceIPs" as "Source IP", requestObject as "Details" where "verb" in ('get','list') order by __time desc limit 1000`
- **Deleted Resources.** The associated query and analysis statement is:
`SELECT auditID as "Event ID", time_format("__time", 'yyyy-MM-dd HH:mm:ss') as "Occurred", "requestURI" as "Resource", "objectRef.name" as "Resource Name", "responseStatus.code" as "Status Code", "sourceIPs" as "Source IP", requestObject as "Details" where "verb" = 'delete' order by __time desc limit 1000`

----End

Viewing CCE Audit Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CCE dashboard templates** under **Dashboard Templates** and click **CCE Audit Log Center** to view the chart details.

- **Filter by namespace.** The associated query and analysis statement is:
`select distinct("objectRef.namespace")`
- **Filter by user.** The associated query and analysis statement is:
`select distinct("user.username")`
- **Filter by operation type.** The associated query and analysis statement is:
`select distinct("verb")`

- **Filter by status code.** The associated query and analysis statement is:
`select distinct("responseStatus.code")`
- **Filter by resource object.** The associated query and analysis statement is:
`select distinct("objectRef.name")`
- **Filter by resource type.** The associated query and analysis statement is:
`select distinct("objectRef.resource")`
- **Filter by request URL.** The associated query and analysis statement is:
`select distinct("requestURI")`
- **Filter by UserAgent.** The associated query and analysis statement is:
`select distinct("userAgent")`
- **Total Audit Records.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(1) as total from log))`
- **The query analysis statements associated with the Number of Operators chart are as follows:**
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(distinct("user.username")) as total from log))`
- **Active Nodes.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(DISTINCT "user.username") as total from log where "objectRef.resource" = 'nodes' and "objectRef.subresource" = 'status' and "verb" in ('update','put','patch') and "user.username" in ('node','system')))`
- **Abnormal Visits.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(1) as total from log where "responseStatus.code" >= 400))`
- **Sensitive Operations.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(1) as "total" from log where ("verb" = 'create' AND "objectRef.subresource" = 'exec') OR ("verb" = 'create' AND "objectRef.subresource" = 'attach' AND "objectRef.resource" = 'pods') OR ("objectRef.resource" = 'secrets' AND "verb" = 'get' AND ("user.username" != 'apiserver') AND ("user.username" not like 'system:node:%')) OR ("verb" = 'delete' AND ("user.username" not like 'system:node:%') AND ("user.username" not like 'system:serviceaccount:kube-system:%') AND ("user.username" != 'system:apiserve') AND ("user.username" != 'system:apiserve') AND ("user.username" != 'system:kube-scheduler') AND ("user.username" != 'system:kube-controller-manager')))`
- **Creation Operations.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(1) as total from log where verb = 'create'))`
- **Update Operations.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(1) as total from log where verb in ('update','patch')))`
- **Deletion Operations.** The associated query and analysis statement is:
`select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as "inc" from (select compare(total , 86400) as diff from(select count(1) as total from log where verb = 'delete'))`
- **Operators.** The associated query and analysis statement is:
`select "user.username" as "Username", count(*) as "count" group by "Username" order by "count" desc`
- **Namespaces.** The associated query and analysis statement is:
`select "objectRef.namespace" as "Namespace", count(*) as "count" group by "Namespace"`
- **Resource Types.** The associated query and analysis statement is:
`select "objectRef.resource" as "Resource type", count(*) as "count" group by "Resource type" order by "count" desc limit 20`
- **Operation Types.** The associated query and analysis statement is:
`select verb as "Operation type", count(*) as "count" group by "Operation type" order by "count" desc`

- **Node Operations.** The associated query and analysis statement is:
`select "verb" as "Operation type", count(*) as "count" where "objectRef.resource" = 'nodes' AND ("verb" in ('create','delete')) group by "Operation type" order by "count" desc`
 - **Workload Operations.** The associated query and analysis statement is:
`select "verb" as "Operation type", count(*) as "count" where "verb" in ('create', 'delete') and "objectRef.resource" in ('deployments','statefulsets','daemonsets','jobs','cronjobs') group by "Operation type" order by "count" desc`
 - **Service/Ingress Operations.** The associated query and analysis statement is:
`select "verb" as "Operation type", count(*) as "count" where "verb" in ('create', 'delete') and "objectRef.resource" in ('ingresses','services') group by "verb" order by "count" desc`
 - **Major Operation Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT("stageTimestamp", 16),'T',' ') AS "dt", "verb", count(*) as "count" where "verb" in ('create','delete','update','patch') group by "dt", "verb" order by "dt" limit 10000`
 - **Non-System User Operation Trend.** The associated query and analysis statement is:
`SELECT REPLACE(LEFT("stageTimestamp", 16),'T',' ') AS "dt", count(*) as "count", "user.username" as "Username" where "user.username" not in ('kube-controller-manager','kube-apiserver-kubelet-client','system','apiserver') group by "dt", "Username" order by "dt" limit 10000`
- End

5.3.3.3 CDN Dashboard Templates

CDN logs the requests to all domain names including those deleted. You can ingest these logs to LTS to analyze the access to your service resources in detail. If you have enabled the enterprise project function, log management is not available for deleted domain names.

CDN dashboard templates support [Viewing CDN Error Analysis](#), [Viewing CDN Basic Data](#), [Viewing CDN User Analysis](#), and [Viewing Popular CDN Resources](#).

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing CDN Error Analysis

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CDN dashboard templates** under **Dashboard Templates** and click **CDN Error Analysis** to view the chart details.

- **Top 5 Domain Name Errors.** The associated query and analysis statement is:
`select domain , count(*) as c where http_code > 400 group by domain order by c desc limit 5`
- **Top 5 URI Errors.** The associated query and analysis statement is:
`select uri , count(*) as c where http_code > 400 group by uri order by c desc limit 5`
- **Status Codes.** The associated query and analysis statement is:
`select http_code , count(*) as c where http_code > 400 group by http_code order by c desc`
- **Errors by Carrier.** The associated query and analysis statement is:
`select ip_to_provider(client_ip) as isp , count(*) as c where http_code > 400 group by isp having ip_to_provider(client_ip) != '' order by c desc limit 10`
- **Errors by Client.** The associated query and analysis statement is:
`select user_agent as "Client version", count(*) as "Errors" where http_code > 400 group by user_agent order by "Errors" desc limit 10`
- **Errors by Province.** The associated query and analysis statement is:

```
select ip_to_province(client_ip) as province , count(*) as c where http_code > 400 and
IP_TO_COUNTRY (client_ip) = 'China' group by province order by c desc limit 50
```

- **4XX Errors.** The associated query and analysis statement is:

```
SELECT
  province AS "Province",
  isp AS "Carrier",
  c AS "Errors",
  round( c * 100.0 / sum( c ), 2 ) AS "Rate (%)"
FROM
  (
  SELECT
    ip_to_province ( client_ip ) AS province,
    ip_to_provider ( client_ip ) AS isp,
    count(*) AS c
  FROM
    log
  WHERE
    http_code >= 400
    AND http_code < 500
  GROUP BY
    province,
    isp
  HAVING
    (
    ip_to_provider ( client_ip )) != ""
  ORDER BY
    c DESC
  )
GROUP BY
  province,
  isp,
  c
```

- **5XX Errors.** The associated query and analysis statement is:

```
SELECT
  province AS "Province",
  isp AS "Carrier",
  c AS "Errors",
  round( c * 100.0 / sum( c ), 2 ) AS "Rate (%)"
FROM
  (
  SELECT
    ip_to_province ( client_ip ) AS province,
    ip_to_provider ( client_ip ) AS isp,
    count(*) AS c
  FROM
    log
  WHERE
    http_code >= 500
  GROUP BY
    province,
    isp
  HAVING
    (
    ip_to_provider ( client_ip )) != ""
  ORDER BY
    c DESC
  )
GROUP BY
  province,
  isp,
  c
```

- **Errors by Country.** The associated query and analysis statement is:

```
select ip_to_country(client_ip) as country , count(*) as c where http_code > 400 group by country
order by c desc limit 50
```

----End

Viewing CDN Basic Data

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CDN dashboard templates** under **Dashboard Templates** and click **CDN Basic Data** to view the chart details.

- **Cache Hit Ratio.** The associated query and analysis statement is:

```
select round(diff[1],2) as Hit_ratio, round(diff[2],2) as diff, round((diff[3]-1)*100, 2) from (select compare(Hit_ratio, 86400) as diff from (select sum(s) * 100.0/count(*) as Hit_ratio from (select case when hit_info = 'HIT' then 1 else 0 end as s from log)))
```
- **Download Speed.** The associated query and analysis statement is:

```
select round(diff[1],2) as speed, round(diff[2],2) as diff, round((diff[3]-1)*100, 2) from (select compare(speed, 86400) as diff from (select sum(response_size) * 1.0 /sum(response_time) as speed from log ))
```
- **Status Codes.** The associated query and analysis statement is:

```
select http_code , count(*) as c group by http_code order by c desc
```
- **Latency Distribution.** The associated query and analysis statement is:

```
select  
  case when response_time < 100 then '~100ms'  
  when response_time < 500 then '100~500ms'  
  when response_time < 1000 then '500ms~1s'  
  when response_time < 5000 then '1~5s'  
  when response_time < 6000 then '5~6s'  
  when response_time < 7000 then '6~7s'  
  when response_time < 8000 then '7~8s'  
  when response_time < 10000 then '8~10s'  
  when response_time < 15000 then '10~15s'  
  else '15s~' end as latency ,  
  count(*) as cnt  
  group by latency  
  order by cnt
```
- **Request Bandwidth.** The associated query and analysis statement is:

```
select TIME_FORMAT (TIME_FLOOR(__time,'PT1M'), 'HH:mm', '+08:00') as thisdate,  
  sum(response_size) * 8/1000000000.0 as "Bandwidth (Gbit/min)"  
  group by TIME_FLOOR(__time,'PT1M')  
  order by TIME_FLOOR(__time,'PT1M')
```
- **Visits and Users.** The associated query and analysis statement is:

```
select TIME_FORMAT (TIME_FLOOR(__time,'PT1M'), 'HH:mm', '+08:00') as thisdate,  
  count(*) as pv, APPROX_COUNT_DISTINCT(client_ip) as uv group by TIME_FLOOR(__time,'PT1M')  
  order by TIME_FLOOR(__time,'PT1M')
```
- **Average Latency.** The associated query and analysis statement is:

```
select TIME_FORMAT (TIME_FLOOR(__time,'PT1M'), 'HH:mm', '+08:00') as thisdate,  
  avg(response_time) as "Average Latency (ms)" group by TIME_FLOOR(__time,'PT1M') order by  
  TIME_FLOOR(__time,'PT1M')
```
- **Request Hit Ratio.** The associated query and analysis statement is:

```
select  
  TIME_FORMAT (TIME_FLOOR(m_time,'PT1M'), 'HH:mm', '+08:00' ) as thisdate ,  
  sum(is_hit)*100.0/count(*) as hit_ratio  
  from (select TIME_FLOOR(__time,'PT1M') as m_time , case when hit_info = 'HIT'  
  then 1 else 0 end as is_hit from log ) group by m_time order by m_time
```

----End

Viewing CDN User Analysis

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CDN dashboard templates** under **Dashboard Templates** and click **CDN User Analysis** to view the chart details.

- **Visits.** The associated query and analysis statement is:
select diff[1] as pv,diff[2] as diff, round(100*(diff[3]-1), 2) from (select compare(pv, 86400) as diff from (select count(*) as pv from log))
- **Visitors.** The associated query and analysis statement is:
select diff[1] as uv, diff[2] as diff, round((diff[3]-1)*100, 2) from (select compare(uv , 86400) as diff from (select APPROX_COUNT_DISTINCT(client_ip) as uv from log))
- **Visits by Client.** The associated query and analysis statement is:
select ua as "Client" , sum(c) as "Visits" from (select case when strpos(ua, 'iphone') > 1 then 'iphone' when strpos(ua, 'ipad') > 1 then 'ipad' when strpos(ua, 'android') > 1 then 'android' when strpos(ua, 'windows') > 1 then 'windows' when strpos(ua, 'mac') > 1 then 'mac' when strpos(ua, 'linux') > 1 then 'linux' else ua end as ua , c from (select count(*) as c , lower(user_agent) as ua from log group by ua order by c desc limit 2000)) group by "Client" order by "Visits" desc limit 100
- **Visits by Carrier.** The associated query and analysis statement is:
select ip_to_provider(client_ip) as isp ,count(*) as "Visits" group by isp order by "Visits" desc limit 100
- **Visits by Geography.** The associated query and analysis statement is:
select ip_to_province(client_ip) as province , count(*) as cnt where IP_TO_COUNTRY (client_ip) = 'China' group by province HAVING province not in ('','Reserved address','*') order by cnt desc limit 100
- **Top Valid Visitors.** The associated query and analysis statement is:
SELECT CASE WHEN ip_to_country(client_ip) = 'Shanghai' THEN concat(client_ip, ' (shanghai)') WHEN ip_to_province(client_ip) = '' THEN concat(client_ip, ' (Unknown IP)') WHEN ip_to_provider(client_ip) = 'Private IP' THEN concat(client_ip, ' (Private IP)') ELSE concat(client_ip, ' (', ip_to_country(client_ip), ', ', ip_to_province(client_ip), ', ', CASE WHEN ip_to_city(client_ip) = '-1' THEN 'Unknown city' ELSE ip_to_city(client_ip) END, ' ', ip_to_provider(client_ip), ')') END AS client, pv as "Total Visits", (pv - success_count) as "Errors", round(CASE WHEN "throughput" > 0 THEN "throughput" ELSE 0 END, 1) AS "Total Downloads (GB)" from (select client_ip, count(*) as pv, sum(response_size) / 1024.0 / 1024 / 1024.0 AS throughput, sum(CASE WHEN http_code < 400 THEN 1 ELSE 0 END) AS success_count from log group by client_ip order by success_count desc limit 100)
- **Top Visitors by Downloads.** The associated query and analysis statement is:
SELECT CASE WHEN ip_to_country(client_ip)='Shanghai' THEN concat(client_ip, ' (shanghai)') WHEN ip_to_province(client_ip)='' THEN concat(client_ip, ' (Unknown IP)') WHEN ip_to_provider(client_ip)='Private IP' THEN concat(client_ip, ' (Private IP)') ELSE concat(client_ip, ' (', ip_to_country(client_ip), ', ', ip_to_province(client_ip), ', ', CASE WHEN ip_to_city(client_ip)='-1' THEN 'Unknown city' ELSE ip_to_city(client_ip) END, ' ',ip_to_provider(client_ip), ')') END AS client, pv as "Total Visits", error_count as "Errors", round(CASE WHEN "throughput" > 0 THEN "throughput" ELSE 0 END, 1) AS "Total Downloads (GB)" from (select client_ip , count(*) as pv, sum(response_size)/1024.0/1024/1024.0 AS throughput , sum(CASE WHEN http_code > 400 THEN 1 ELSE 0 END) AS error_count from log group by client_ip order by throughput desc limit 100)

----End

Viewing Popular CDN Resources

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CDN dashboard templates** under **Dashboard Templates** and click **Popular CDN Resources** to view the chart details.

- **5 Most Visited Domain Names.** The associated query and analysis statement is:
select domain ,count(*) as cnt group by domain order by cnt desc limit 5
- **5 Most Downloaded Domain Names.** The associated query and analysis statement is:
select domain , sum(response_size) as "Total Downloads" group by domain order by "Total Downloads" desc limit 5
- **Popular Visits (URI).** The associated query and analysis statement is:
select uri as URI, "Visits", "Visitors", round(CASE WHEN "Total Downloads (GB)" > 0 THEN "Total Downloads (GB)" ELSE 0 END, 2) AS "Total Downloads (GB)" from (select uri ,count(*) as "Visits" , APPROX_COUNT_DISTINCT(client_ip) as "Visitors", sum(response_size)/1024.0/1024.0/1024.0 as "Total Downloads (GB)" where http_code < 400 group by uri order by "Visits" desc limit 100)

- **Popular Visits (Source).** The associated query and analysis statement is:

```
select refer_domain as "Referer",c as "Times",uv as "Users", round(c * 100.0 / sum(c), 2) as "Percentage" from (select refer_domain as refer_domain,count(*) as c,APPROX_COUNT_DISTINCT(client_ip) as uv from log where refer_domain != " group by refer_domain order by c desc limit 100 ) GROUP BY refer_domain, c, uv
```
- **Visits by Geography.** The associated query and analysis statement is:

```
select ip_to_province(client_ip) as province , count(*) as cnt where IP_TO_COUNTRY (client_ip) = 'China' group by province HAVING province not in ('','Reserved address','*') order by cnt desc limit 1000
```
- **Download Speed by Geography.** The associated query and analysis statement is:

```
select province, round( CASE WHEN "speed" > 0 THEN "speed" ELSE 0 END, 3 ) AS "speed" from (select ip_to_province(client_ip) as province , sum(response_size)* 1.0 /(sum(response_time)+1) as "speed" , count(*) as c where IP_TO_COUNTRY (client_ip) = 'China' group by province HAVING province not in ('','Reserved address','*') order by c desc limit 40)
```
- **Statistics by Province.** The associated query and analysis statement is:

```
select ip_to_province(client_ip) as "Province" ,count(*) as "Visits", sum(response_size)/ 1024.0/1024.0/1024.0 as "Total Downloads (GB)" , sum(response_size) * 1.0 /sum(response_time) as "Download Speed (KB/s)" group by "Province" having ip_to_province(client_ip) != " order by "Total Downloads (GB)" desc limit 200
```
- **Carrier Traffic and Speed.** The associated query and analysis statement is:

```
select ip_to_provider(client_ip) as isp , sum(response_size)* 1.0 /(sum(response_time)+1) as "Download Speed (KB/s)" , sum(response_size)/1024.0/1024.0/1024.0 as "Total Downloads (GB)", count(*) as c group by isp having ip_to_provider(client_ip) != " order by c desc limit 10
```
- **Statistics by Carrier.** The associated query and analysis statement is:

```
select "Carrier", "Visits", round( CASE WHEN "Total Downloads (GB)" > 0 THEN "Total Downloads (GB)" ELSE 0 END, 2 ) AS "Total Downloads (GB)", round( CASE WHEN "Download Speed (KB/s)" > 0 THEN "Download Speed (KB/s)" ELSE 0 END, 2 ) AS "Download Speed (KB/s)" from ( select ip_to_provider(client_ip) as "Carrier" ,count(*) as "Visits", sum(response_size)/ 1024.0/1024.0/1024.0 as "Total Downloads (GB)" , sum(response_size) * 1.0 /sum(response_time) as "Download Speed (KB/s)" group by "Carrier" having ip_to_provider(client_ip) != " and "Carrier" not in (*) order by "Total Downloads (GB)" desc limit 200)
```

----End

5.3.3.4 CFW Dashboard Templates

CFW is a next-generation cloud-native firewall. It protects Internet and VPC borders on the cloud by real-time intrusion detection and prevention, global unified access control, full traffic analysis, log audit, and tracing. CFW employs AI for intelligent defense, and can be elastically scaled to meet changing business needs, helping you easily handle security threats. It allows you to view the risk levels, affected ports, matched rules, and attack event types of detected dangerous traffic in attack event logs. You can also view all traffic allowed or blocked in access control logs to better adjust access control policies.

CFW dashboard templates support [Viewing CFW Access Log Center](#), [Viewing CFW Traffic Log Center](#), and [Viewing CFW Attack Log Center](#).

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing CFW Access Log Center

- Step 1** Log in to the LTS console. In the navigation pane, choose **Log Management**.
- Step 2** In the **Log Applications** area, click **CFW Log Center** and choose **Go to the Dashboard**.

Step 3 In the dashboard list, choose **CFW dashboard templates** under **Dashboard Templates** and click **CFW access log center** to view the chart details.

- The **Blockage Trends (Internet Access)** chart displays the blockage trend of Internet access. The associated query and analysis statement is:

```
select time_series(MILLIS_TO_TIMESTAMP(hit_time), 'PT1M', 'yyyy-MM-dd HH:mm:ss', '0') as t_time,COUNT(*) as frequency WHERE action='deny' AND direction='out2in' group by t_time order by t_time
```
- The **Blockage Trends (Server-Originated Access)** chart displays the blockage trend of server-originated access. The associated query and analysis statement is:

```
select time_series(MILLIS_TO_TIMESTAMP(hit_time), 'PT1M', 'yyyy-MM-dd HH:mm:ss', '0') as t_time,COUNT(*) as frequency WHERE action='deny' AND direction='in2out' group by t_time order by t_time
```
- The **5 Most Blocked Applications** chart displays the top 5 applications with the most Internet access blocks. The associated query and analysis statement is:

```
SELECT app, COUNT(*) as frequency WHERE action='deny' AND direction='out2in' GROUP BY app ORDER BY frequency DESC LIMIT 5
```
- The **5 Most Blocked Destinations** chart displays the top 5 destinations with the most Internet access blocks. The associated query and analysis statement is:

```
SELECT dst_ip, COUNT(*) as frequency WHERE action='deny' AND direction='out2in' GROUP BY dst_ip ORDER BY frequency DESC LIMIT 5
```
- The **5 Most Blocked Sources** chart displays the top 5 sources with the most Internet access blocks. The associated query and analysis statement is:

```
SELECT src_ip, COUNT(*) as frequency WHERE action='deny' AND direction='out2in' GROUP BY src_ip ORDER BY frequency DESC LIMIT 5
```
- The **5 Most Blocked Applications (Server-Originated Access)** chart displays the top 5 applications with the most server-originated access blocks. The associated query and analysis statement is:

```
SELECT app, COUNT(*) as frequency WHERE action='deny' AND direction='in2out' GROUP BY app ORDER BY frequency DESC LIMIT 5
```
- The **5 Most Blocked Destinations (Server-Originated Access)** chart displays the top 5 destinations with the most server-originated access blocks. The associated query and analysis statement is:

```
SELECT dst_ip, COUNT(*) as frequency WHERE action='deny' AND direction='in2out' GROUP BY dst_ip ORDER BY frequency DESC LIMIT 5
```
- The **5 Most Blocked Sources (Server-Originated Access)** chart displays the top 5 sources with the most server-originated access blocks. The associated query and analysis statement is:

```
SELECT src_ip, COUNT(*) as frequency WHERE action='deny' AND direction='in2out' GROUP BY src_ip ORDER BY frequency DESC LIMIT 5
```

----End

Viewing CFW Traffic Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Log Management**.

Step 2 In the **Log Applications** area, click **CFW Log Center** and choose **Go to the Dashboard**.

Step 3 In the dashboard list, choose **CFW dashboard templates** under **Dashboard Templates** and click **CFW traffic log center** to view the chart details.

- The **Traffic Trends (Internet Access)** chart displays the traffic trend of Internet access. The associated query and analysis statement is:

```
select time_series(MILLIS_TO_TIMESTAMP(start_time), 'PT1M', 'yyyy-MM-dd HH:mm:ss', '0') as t_time,
SUM(to_s_bytes) AS 'Inbound', SUM(to_c_bytes) AS 'Outbound' WHERE direction='out2in' group by
t_time order by t_time
```

- The **Region Distribution of Inbound Internet Access (China)** chart displays the distribution of inbound Internet access by region in China. The associated query and analysis statement is:

```
SELECT count(*) AS PV, ip_to_province(src_ip) AS province WHERE direction='out2in' and
IP_TO_COUNTRY (src_ip) = 'China' GROUP BY province HAVING province not in ('Reserved
address',*) ORDER BY PV DESC
```
- The **Region Distribution of Inbound Internet Access (Global)** chart displays the distribution of inbound Internet access by region in the world. The associated query and analysis statement is:

```
SELECT count(*) AS PV, ip_to_country(src_ip) AS country WHERE direction='out2in' GROUP BY country
HAVING country not in ('Reserved address',*) ORDER BY PV DESC
```
- The **Application Distribution of Internet Access** chart displays the application distribution of Internet access. The associated query and analysis statement is:

```
SELECT app, COUNT(*) AS num WHERE direction='out2in' GROUP BY app ORDER BY num DESC
```
- The **Top 5 Source IP Addresses** chart displays the top 5 source IP addresses of Internet access. The associated query and analysis statement is:

```
select src_ip, SUM(bytes)/1024 as sum_bytes WHERE direction='out2in' GROUP BY src_ip ORDER BY
sum_bytes DESC LIMIT 5
```
- The **Top 5 Destination IP Addresses** chart displays the top 5 destination IP addresses of Internet access. The associated query and analysis statement is:

```
select dst_ip, SUM(bytes)/1024 as sum_bytes WHERE direction='out2in' GROUP BY dst_ip ORDER BY
sum_bytes DESC LIMIT 5
```
- The **Traffic Trends (Server-Originated Access)** chart displays the traffic trend of server-originated access. The associated query and analysis statement is:

```
select time_series(MILLIS_TO_TIMESTAMP(start_time), 'PT1M', 'yyyy-MM-dd HH:mm:ss', '0') as t_time,
SUM(to_c_bytes) AS 'Inbound', SUM(to_s_bytes) AS 'Outbound' WHERE direction='in2out' group by
t_time order by t_time
```
- The **Destination Region Distribution of Server Originated Access (China)** chart displays the destination region distribution of server-originated access in China. The associated query and analysis statement is:

```
SELECT count(*) AS PV, ip_to_province(dst_ip) AS province WHERE direction='in2out' and
IP_TO_COUNTRY (dst_ip) = 'China' GROUP BY province HAVING province not in ('Reserved
address',*) ORDER BY PV DESC
```
- The **Destination Region Distribution (Global)** chart displays the destination region distribution in the world. The associated query and analysis statement is:

```
SELECT count(*) AS PV, ip_to_country(dst_ip) AS country WHERE direction='in2out' GROUP BY country
HAVING country not in ('Reserved address',*) ORDER BY PV DESC
```
- The **Application Distribution (Server-Originated Access)** chart displays the application distribution of server-originated access. The associated query and analysis statement is:

```
SELECT app, COUNT(*) AS num WHERE direction='in2out' GROUP BY app ORDER BY num DESC
```
- The **Top 5 Source IP Addresses (Server-Originated Access)** chart displays the top 5 source IP addresses of server-originated access. The associated query and analysis statement is:

```
select src_ip, SUM(bytes)/1024 as sum_bytes WHERE direction='in2out' GROUP BY src_ip ORDER BY
sum_bytes DESC LIMIT 5
```
- The **Top 5 Destination IP Addresses (Server-Originated Access)** chart displays the top 5 destination IP addresses of server-originated access. The associated query and analysis statement is:

```
select dst_ip, SUM(bytes)/1024 as sum_bytes WHERE direction='in2out' GROUP BY dst_ip ORDER BY  
sum_bytes DESC LIMIT 5
```

----End

Viewing CFW Attack Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Log Management**.

Step 2 In the **Log Applications** area, click **CFW Log Center** and choose **Go to the Dashboard**.

Step 3 In the dashboard list, choose **CFW dashboard templates** under **Dashboard Templates** and click **CFW attack log center** to view the chart details.

- **Attack Trends.** The associated query and analysis statement is:

```
select time_series(MILLIS_TO_TIMESTAMP(event_time), 'PT1M', 'yyyy-MM-dd HH:mm:ss', '0') as  
t_time, count(*) as frequency group by t_time order by t_time
```
- **Sources (China).** The associated query and analysis statement is:

```
SELECT count(*) as PV,ip_to_province(src_ip) as province WHERE IP_TO_COUNTRY (src_ip) = 'China'  
GROUP BY province HAVING province not in ('','Reserved address','*')
```
- **Sources (Global).** The associated query and analysis statement is:

```
SELECT count(*) AS PV,ip_to_country(src_ip) AS country GROUP BY country HAVING country not in  
('','Reserved address','*')
```
- **Types.** The associated query and analysis statement is:

```
SELECT attack_type, COUNT(*) as num GROUP BY attack_type ORDER BY num
```
- **Top 5 Destinations.** The associated query and analysis statement is:

```
SELECT dst_ip, COUNT(*) as frequency GROUP BY dst_ip ORDER BY frequency DESC LIMIT 5
```
- **Top 5 Sources.** The associated query and analysis statement is:

```
SELECT src_ip, COUNT(*) as frequency GROUP BY src_ip ORDER BY frequency DESC LIMIT 5
```

----End

5.3.3.5 CSE Dashboard Templates

CSE is a cloud middleware used for microservice applications. You can also use it with other cloud services to quickly build a cloud-native microservice system for quick development and high-availability O&M of microservice applications.

CSE dashboard templates support [Viewing CSE Layer Access Center](#), [Viewing CSE Layer Monitoring Center](#), and [Viewing CSE Layer Monitoring in Seconds](#).

Prerequisites

- CSE logs have been collected to LTS.
- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing CSE Layer Access Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CSE dashboard templates** under **Dashboard Templates** and click **CSE Layer Access Center** to view the chart details.

- **upstream_host** allows you to filter upstream IP addresses. The associated query and analysis statement is:

```
select distinct(upstream_host)
```

- **trace_id** allows you to filter traces. The associated query and analysis statement is:

```
select distinct(trace_id)
```
- **Day-over-day PV Change.** The associated query and analysis statement is:

```
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare( "pv" , 86400) as diff from (select count(1) as "pv" from log))
```
- **Week-on-week PV Change.** The associated query and analysis statement is:

```
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare( "pv" , 604800) as diff from (select count(1) as "pv" from log))
```
- **Day-over-day UV Change.** The associated query and analysis statement is:

```
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare( "uv" , 86400) as diff from (select APPROX_COUNT_DISTINCT(authority) as "uv" from log))
```
- **Week-on-week UV Change.** The associated query and analysis statement is:

```
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare( "uv" , 604800) as diff from (select APPROX_COUNT_DISTINCT(authority) as "uv" from log))
```
- **PV Distribution (China).** The associated query and analysis statement is:

```
select ip_to_province(authority) as province, sum(ori_pv) as pv from (select authority, count(1) as ori_pv group by authority ORDER BY ori_pv desc LIMIT 10000) where IP_TO_COUNTRY (authority) = 'China' group by province HAVING province not in ('', 'Reserved address', '*')
```
- **PV Distribution (Global).** The associated query and analysis statement is:

```
SELECT ip_to_country(authority) as country, sum(ori_pv) as PV from (select authority, count(1) as ori_pv group by authority ORDER BY ori_pv desc LIMIT 10000) GROUP BY country HAVING country not in ('', 'Reserved address', '*')
```
- **Average Latency Distribution (China).** The associated query and analysis statement is:

```
SELECT province, round( CASE WHEN "Average latency (ms)" > 0 THEN "Average latency (ms)" ELSE 0 END, 3 ) AS "Average latency (ms)" FROM (SELECT ip_to_province(authority) as province, sum(rt)/ sum(ori_pv) * 1000 AS "Average latency (ms)" from (select authority, sum(duration) as rt, count(1) as ori_pv group by authority ORDER BY ori_pv desc LIMIT 10000) WHERE IP_TO_COUNTRY (authority) = 'China' GROUP BY province ) where province not in ('', 'Reserved address', '*')
```
- **Average Latency Distribution (Global).** The associated query and analysis statement is:

```
SELECT country, round( CASE WHEN "Average latency (ms)" > 0 THEN "Average latency (ms)" ELSE 0 END, 2 ) AS "Average latency (ms)" FROM (SELECT ip_to_country(authority) as country, sum(rt)/ sum(ori_pv) * 1000 AS "Average latency (ms)" from (select authority, sum(duration) as rt, count(1) as ori_pv group by authority ORDER BY ori_pv desc LIMIT 10000) GROUP BY country ) where country not in ('', 'Reserved address', '*')
```
- **PV/UV Today.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss') as _time_ , count(1) as PV, APPROX_COUNT_DISTINCT(authority) as UV from log WHERE __time_ <= CURRENT_TIMESTAMP and __time_ >= DATE_TRUNC( 'DAY', (CURRENT_TIMESTAMP + INTERVAL '8' HOUR)) - INTERVAL '8' HOUR group by _time_ order by _time_)
```
- **Top 10 Provinces by Visits.** The associated query and analysis statement is:

```
select ip_to_province(authority) as "province", sum(ori_pv) as "Visits" from (select authority, count(1) as ori_pv group by authority ORDER BY ori_pv desc LIMIT 10000) group by "province" HAVING "province" <> '-1' order by "Visits" desc limit 10
```
- **Top 10 Cities by Visits.** The associated query and analysis statement is:

```
select ip_to_city(authority) as "city", sum(ori_pv) as "Visits" from (select authority, count(1) as ori_pv group by authority ORDER BY ori_pv desc LIMIT 10000) group by "city" HAVING "city" <> '-1' order by "Visits" desc limit 10
```
- **Top 10 Hosts by Visits.** The associated query and analysis statement is:

```
select upstream_host as "Host", count(1) as "PV" group by "Host" order by "PV" desc limit 10
```
- **Top 10 UserAgents by Visits.** The associated query and analysis statement is:

```
select user_agent as "UserAgent", count(1) as "PV" group by "UserAgent" order by "PV" desc limit 10
```
- **Device Distribution by Type.** The associated query and analysis statement is:

```
select case when regexp_like(lower(user_agent), 'iphone|ipod|android|ios') then 'Mobile' else 'PC' end as type , count(1) as total group by type
```

- **Device Distribution by System.** The associated query and analysis statement is:

```
select case when regexp_like(lower(user_agent), 'iphone|ipod|ios') then 'IOS' when  
regexp_like(lower(user_agent), 'android') then 'Android' else 'other' end as type , count(1) as total  
group by type HAVING type != 'other'
```
- **TOP URL.** The associated query and analysis statement is:

```
select path , count(1) as pv, APPROX_COUNT_DISTINCT(authority) as UV, round(sum( case when  
response_code < 400 then 1 else 0 end ) * 100.0 / count(1), 2) as "Access Success Rate" group by  
path ORDER by pv desc
```
- **Top IP Addresses by Visits.** The associated query and analysis statement is:

```
select authority as "Source IP Address",ip_to_country(authority) as "Country/  
Region",ip_to_province(authority) as "Province",ip_to_city(authority) as  
"City",ip_to_provider(authority) as "Carrier",count(1) as "PV" group by authority ORDER by "PV" desc  
limit 100
```

----End

Viewing CSE Layer Monitoring Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **CSE dashboard templates** under **Dashboard Templates** and click **CSE Layer Monitoring Center** to view the chart details.

- **upstream_host** allows you to filter upstream IP addresses. The associated query and analysis statement is:

```
select distinct(upstream_host)
```
- **trace_id** allows you to filter traces. The associated query and analysis statement is:

```
select distinct(trace_id)
```
- **PV.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss') as _time_PV FROM ( SELECT TIME_CEIL  
( TIME_PARSE(start_time), 'PT300S' ) AS _time_ , count( 1 ) AS PV FROM log GROUP BY _time_ )
```
- **Request Success Rate.** The associated query and analysis statement is:

```
select ROUND(sum(case when response_code < 400 then 1 else 0 end) * 100.0 / count(1),2) as cnt
```
- **Average Latency.** The associated query and analysis statement is:

```
select round(avg(duration) * 1000, 3) as cnt
```
- **4xx Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "response_code" >= 400 and "response_code" < 500
```
- **404 Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "response_code" = 404
```
- **429 Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "response_code" = 429
```
- **504 Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "response_code" = 504
```
- **5xx Requests.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss') as _time_cnt FROM ( SELECT TIME_CEIL  
( TIME_PARSE(start_time), 'PT300S' ) AS _time_ , count( 1 ) AS cnt FROM log where "response_code"  
>= 500 GROUP BY _time_ )
```
- **Status Code Distribution.** The associated query and analysis statement is:

```
SELECT response_code, COUNT(1) AS rm GROUP BY response_code
```
- **UV.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss') as _time_UV FROM (select  
TIME_CEIL(TIME_PARSE(start_time),'PT600S') AS _time_ , APPROX_COUNT_DISTINCT(authority) as  
UV from log group by _time_)
```
- **Traffic.** The associated query and analysis statement is:

```
select TIME_FORMAT(_time_,'yyyy-MM-dd HH:mm:ss') AS _time_,round( CASE WHEN "Inbound" > 0 THEN "Inbound" ELSE 0 END, 2 ) AS "Inbound",round( CASE WHEN "Outbound" > 0 THEN "Outbound" ELSE 0 END, 2 ) AS "Outbound" FROM (SELECT TIME_CEIL(TIME_PARSE(start_time),'PT600S') AS _time_,sum(bytes_received) / 1024.0 AS "Inbound",sum(bytes_sent) / 1024.0 AS "Outbound" group by _time_)
```

- **Access Failure Rate.** The associated query and analysis statement is:
SELECT TIME_FORMAT(_time_,'yyyy-MM-dd HH:mm:ss') as _time_,round(CASE WHEN "Failure rate" > 0 THEN "Failure rate" ELSE 0 END, 2) AS "Failure rate",round(CASE WHEN "5xx ratio" > 0 THEN "5xx ratio" ELSE 0 END, 2) AS "5xx ratio" from (select TIME_CEIL(TIME_PARSE(start_time),'PT600S') AS _time_,sum(case when response_code >= 400 then 1 else 0 end) * 100.0 / count(1) as 'Failure rate' , sum(case when response_code >=500 THEN 1 ELSE 0 END)*100.0/COUNT(1) as '5xx ratio' group by _time_)
- **Latency.** The associated query and analysis statement is:
select TIME_FORMAT(_time_,'yyyy-MM-dd HH:mm:ss') as _time_,round(CASE WHEN "Avg" > 0 THEN "Avg" ELSE 0 END, 2) AS "Avg",round(CASE WHEN "P50" > 0 THEN "P50" ELSE 0 END, 2) AS "P50",round(CASE WHEN "P90" > 0 THEN "P90" ELSE 0 END, 2) AS "P90",round(CASE WHEN "P99" > 0 THEN "P99" ELSE 0 END, 2) AS "P99",round(CASE WHEN "P9999" > 0 THEN "P9999" ELSE 0 END, 2) AS "P9999" from (select TIME_CEIL(TIME_PARSE(start_time),'PT600S') as _time_,avg(duration) * 1000 as "Avg", APPROX_QUANTILE_DS("duration", 0.50)*1000 as "P50", APPROX_QUANTILE_DS("duration", 0.90)*1000 as "P90",APPROX_QUANTILE_DS("duration", 0.99)*1000 as "P99",APPROX_QUANTILE_DS("duration", 0.9999)*1000 as "P9999" group by _time_)
- **Top Host Requests.** The associated query and analysis statement is:
SELECT "host", pv, uv, round(CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2) AS "Access Success Rate (%)", round(CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3) AS "Average Latency (ms)", round(CASE WHEN "Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3) AS "Inbound (KB)", round(CASE WHEN "Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3) AS "Outbound (KB)" FROM (SELECT "host", count(1) AS pv, APPROX_COUNT_DISTINCT (my_remote_addr) AS uv, sum(CASE WHEN "status" < 400 THEN 1 ELSE 0 END) * 100.0 / count(1) AS "Access Success Rate (%)", avg(request_time) * 1000 AS "Average Latency (ms)", sum(request_length) / 1024.0 AS "Inbound (KB)", sum(bytes_sent) / 1024.0 AS "Outbound (KB)" WHERE "host" != " " GROUP BY "host") ORDER BY pv DESC
- **Top Host Latencies.** The associated query and analysis statement is:
SELECT "upstream_host", pv, round(CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2) AS "Access Success Rate (%)", round(CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3) AS "Average Latency (ms)", round(CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3) AS "P90 Latency (ms)", round(CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3) AS "P99 Latency (ms)" FROM (SELECT "upstream_host", count(1) AS pv, sum(CASE WHEN "response_code" < 400 THEN 1 ELSE 0 END) * 100.0 / count(1) AS "Access Success Rate (%)", avg(duration) * 1000 AS "Average Latency (ms)",APPROX_QUANTILE_DS(duration, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(duration, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "upstream_host" != " " GROUP BY "upstream_host") ORDER BY "Average Latency (ms)" desc
- **Top Host Failure Rates.** The associated query and analysis statement is:
SELECT "upstream_host", pv,round(CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)" ELSE 0 END, 2) AS "Access Failure Rate (%)", round(CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3) AS "Average Latency (ms)", round(CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3) AS "P90 Latency (ms)", round(CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3) AS "P99 Latency (ms)" FROM (SELECT "upstream_host", count(1) AS pv, sum(CASE WHEN "response_code" >= 400 THEN 1 ELSE 0 END) * 100.0 / count(1) AS "Access Failure Rate (%)", avg(duration) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(duration, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(duration, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "upstream_host" != " " GROUP BY "upstream_host") ORDER BY "Access Failure Rate (%)" desc
- **Top URL Requests.** The associated query and analysis statement is:
SELECT path, pv,uv, round(CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2) AS "Access Success Rate (%)", round(CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3) AS "Average Latency (ms)", round(CASE WHEN "Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3) AS "Inbound (KB)", round(CASE WHEN "Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3) AS "Outbound (KB)" FROM (SELECT path, count(1) AS pv, APPROX_COUNT_DISTINCT (authority) AS uv, sum(CASE WHEN "response_code" < 400 THEN 1 ELSE 0 END) * 100.0 / count(1) AS "Access Success Rate (%)", avg(duration) * 1000 AS "Average Latency (ms)", sum(bytes_received) / 1024.0 AS "Inbound (KB)", sum(bytes_sent) / 1024.0 AS "Outbound (KB)" WHERE "upstream_host" != " " GROUP BY path) ORDER BY pv desc

- **Top URL Failure Rates.** The associated query and analysis statement is:

```
SELECT path, pv, round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)" ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM( SELECT path, count( 1 ) AS pv, sum( CASE WHEN "response_code" >= 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( duration ) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(duration, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(duration, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "upstream_host" != " GROUP BY path ) ORDER BY "Access Failure Rate (%)" desc
```
- **Top Backend Requests.** The associated query and analysis statement is:

```
SELECT addr, pv, uv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN "Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT authority as addr, count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( authority ) AS uv, sum( CASE WHEN "response_code" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)", avg( duration ) * 1000 AS "Average Latency (ms)", sum( bytes_received ) / 1024.0 AS "Inbound (KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "upstream_host" != " GROUP BY addr having length(authority) > 2) ORDER BY "pv" desc
```
- **Top Backend Latencies.** The associated query and analysis statement is:

```
SELECT addr,pv,round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)",round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)",round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)",round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM (SELECT authority as addr,count( 1 ) AS pv,sum( CASE WHEN "response_code" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)",avg( duration ) * 1000 AS "Average Latency (ms)",APPROX_QUANTILE_DS(duration, 0.9) * 1000 AS "P90 Latency (ms)",APPROX_QUANTILE_DS(duration, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "upstream_host" != " and "authority" != '-' GROUP BY addr ) ORDER BY "Average Latency (ms)" desc
```
- **Top Backend Failure Rates.** The associated query and analysis statement is:

```
SELECT addr, pv, round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)" ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM ( SELECT authority as addr, count( 1 ) AS pv, sum( CASE WHEN "response_code" >= 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( duration ) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(duration, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(duration, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "upstream_host" != " and "authority" != '-' GROUP BY addr) ORDER BY "Access Failure Rate (%)" desc
```
- **Top URL Latencies.** The associated query and analysis statement is:

```
SELECT path, pv,round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)",round( CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)",round( CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)",round( CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)" FROM (SELECT path, count( 1 ) AS pv, sum( CASE WHEN "response_code" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)", avg( duration ) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(duration, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(duration, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "upstream_host" != " GROUP BY path ) ORDER BY "Average Latency (ms)" desc
```

----End

Viewing CSE Layer Monitoring in Seconds

- Step 1** Log in to the LTS console. In the navigation pane, choose **Dashboards**.
- Step 2** Choose **CSE dashboard templates** under **Dashboard Templates** and click **CSE Layer Monitoring in Seconds** to view the chart details.

- **upstream_host** allows you to filter upstream IP addresses. The associated query and analysis statement is:

```
select distinct(upstream_host)
```
- **trace_id** allows you to filter traces. The associated query and analysis statement is:

```
select distinct(trace_id)
```
- **QPS**. The associated query and analysis statement is:

```
SELECT TIME_FORMAT(TIME_CEIL(TIME_PARSE(start_time),'PT5S'),'yyyy-MM-dd HH:mm:ss') AS  
__time_ , COUNT(*) as QPS from log group by __time_
```
- **Success Rate**. The associated query and analysis statement is:

```
select __time,round(CASE WHEN "Success rate" > 0 THEN "Success rate" else 0 end,2) as "Success  
rate" from (select TIME_FORMAT(TIME_CEIL(TIME_PARSE(start_time),'PT5S'),'yyyy-MM-dd  
HH:mm:ss') as __time, sum(case when response_code < 400 then 1 else 0 end) * 100.0 / count(1) as  
'Success rate' from log group by __time)
```
- **Latency**. The associated query and analysis statement is:

```
select __time,round(CASE WHEN "Access latency" > 0 THEN "Access latency" else 0 end,2) as "Access  
latency",round(CASE WHEN "Upstream latency" > 0 THEN "Upstream latency" else 0 end,2) as  
"Upstream latency" from (select TIME_FORMAT(TIME_CEIL(TIME_PARSE(start_time),'PT5S'),'yyyy-MM-  
dd HH:mm:ss') as __time, avg(duration)* 1000 as 'Access latency',avg(upstream_service_time)* 1000  
as 'Upstream latency' from log group by __time)
```
- **Traffic**. The associated query and analysis statement is:

```
select __time,round( CASE WHEN "Incoming" > 0 THEN "Incoming" ELSE 0 END, 3 ) AS  
"Incoming",round( CASE WHEN "Outgoing body" > 0 THEN "Outgoing body" ELSE 0 END, 3 ) AS  
"Outgoing body" from (select TIME_FORMAT(TIME_CEIL(TIME_PARSE(start_time),'PT5S'),'yyyy-MM-  
dd HH:mm:ss') as __time , sum("bytes_received") / 1024.0 as "Incoming", sum("bytes_sent") / 1024.0  
as "Outgoing body" group by __time)
```
- **Status Codes**. The associated query and analysis statement is:

```
SELECT TIME_CEIL ( TIME_PARSE ( start_time ), 'PT5S' ) AS "time", SUM( CASE WHEN  
"response_code" >= 200 AND "response_code" < 300 THEN 1 ELSE 0 END ) AS "2XX", SUM( CASE  
WHEN "response_code" >= 300 AND "response_code" < 400 THEN 1 ELSE 0 END ) AS "3XX",  
SUM( CASE WHEN "response_code" >= 400 AND "response_code" < 500 THEN 1 ELSE 0 END ) AS  
"4XX", SUM( CASE WHEN "response_code" >= 500 AND "response_code" < 600 THEN 1 ELSE 0 END )  
AS "5XX", SUM( CASE WHEN "response_code" < 200 OR "response_code" >= 600 THEN 1 ELSE 0  
END ) AS "Other" FROM log WHERE TIME_PARSE ( start_time ) IS NOT NULL GROUP BY "time"  
ORDER BY "time" ASC LIMIT 100000
```

----End

5.3.3.6 DCS Dashboard Template

DCS is an in-memory database service. It is compatible with Redis, an in-memory database engine, and meets your requirements for high concurrency and fast data access. LTS collects, stores, and queries cloud resource operation records. You can use these records to perform security analysis, track resource changes, audit compliance, and locate faults.

LTS provides a one-stop wizard to collect DCS logs. The DCS audit log center dashboard displays charts such as the number of access users, access clients, and audit logs.

Prerequisites

Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

DCS Audit Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **DCS dashboard templates** under **Dashboard Templates** and click **DCS audit log center** to view the chart details.

- **Users.** The associated query and analysis statement is:

```
select count(distinct(user)) as user_num
```
- **Clients.** The associated query and analysis statement is:

```
select count(distinct(client_addr)) as client_num
```
- **Audit Logs.** The associated query and analysis statement is:

```
select count(1) as log_num
```
- **Average Response Time.** The associated query and analysis statement is:

```
select avg(use_time) as avg_time
```
- **Average QPS.** The associated query and analysis statement is:

```
select count(*) / CAST((TIMESTAMPDIFF (minute, MIN(_time),MAX(_time))+1) as FLOAT)
```
- **Top 5 Users.** The associated query and analysis statement is:

```
select user, count(1) as 'user_count' group by user order by count(1) desc LIMIT 5
```
- **Top 5 Clients.** The associated query and analysis statement is:

```
select client_addr, count(1) as 'remote_count' group by client_addr order by count(1) desc limit 5
```
- **Top 5 Commands.** The associated query and analysis statement is:

```
select command_name, count(1) as 'command_count' group by command_name order by count(1) desc
```
- **Hot Keys.** The associated query and analysis statement is:

```
SELECT count(*) as 'count', command_keys GROUP by command_keys order by count(*) desc limit 5
```
- **Audit Log Details.** The associated query and analysis statement is:

```
select "client_addr" as 'Client IP Address',"client_type" as 'Client Type',"server_addr" as 'Server IP Address',"command_name" as 'Command',"command_keys" as 'Keys',"command_param" as 'Command Content',"command_type" as 'Command Type',"use_time" as 'Execution Duration',"time" as 'Executed',"db" as 'DB',"user" as 'Username',"instance_id" as 'Instance ID',"role" as 'Node Role',"extend" as 'Extended Information'
```

----End

5.3.3.7 DDS Dashboard Template

DDS is a MongoDB-compatible database service that is secure, highly available, reliable, scalable, and easy to use. You can use DB instance creation, scaling, redundancy, backup, restoration, monitoring, and alarm reporting functions with just a few clicks on the DDS console. LTS allows you to analyze, search, visualize, and download logs, and view real-time logs.

LTS provides a one-stop wizard to collect DDS logs. The DDS audit log center dashboard displays charts such as the number of audit logs, access users, and access clients.

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

DDS Audit Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **DDS dashboard templates** under **Dashboard Templates** and click **DDS audit log center** to view the chart details.

- **Filter by operation type.** The associated query and analysis statement is:

```
select distinct(optype)
```

- Filter by client IP address. The associated query and analysis statement is:
`select distinct(user_ip)`
- Filter by database name. The associated query and analysis statement is:
`select distinct(db)`
- Filter by username. The associated query and analysis statement is:
`select distinct(user)`
- Filter by collection name. The associated query and analysis statement is:
`select distinct(coll)`
- **Audit Logs.** The associated query and analysis statement is:
`select count(1) as log_num`
- **Users.** The associated query and analysis statement is:
`select count(distinct(user)) as user_num`
- **Clients.** The associated query and analysis statement is:
`select count(distinct(user_ip)) as client_num`
- **Top 5 Commands.** The associated query and analysis statement is:
`select optype, count(1) as 'command_count' group by optype order by count(1) desc LIMIT 5`
- **Top 5 Users.** The associated query and analysis statement is:
`select user, count(1) as 'user_count' group by user order by count(1) desc LIMIT 5`
- **Top 5 Clients.** The associated query and analysis statement is:
`select user_ip, count(1) as 'remote_count' group by user_ip order by count(1) desc LIMIT 5`
- **Audit Log Details.** The associated query and analysis statement is:
`select "time" as "Executed","user" as "Username","param" as "Query Statement","instanceid" as "Instance ID","db" as "DB","coll" as "Collection Name ","user_ip" as "Client IP Address"`

----End

5.3.3.8 DMS Dashboard Template

DMS for Kafka is a message queuing service using open-source Apache Kafka. It provides Kafka instances with isolated computing, storage, and bandwidth resources. Rebalancing logs record rebalancing details, including the time, reason, and triggering client of rebalancing.

LTS provides a one-stop wizard to collect DMS rebalancing logs. It also supports multi-dimensional analysis and structuring, and offers a dashboard for DMS rebalancing logs. The dashboard displays the number of rebalancing consumer groups, number of rebalances, number of rebalances for consumer groups, rebalancing reasons, and group details of DMS rebalancing logs.

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

DMS Rebalancing Log Center

Step 1 Log in to the LTS console.

Step 2 In the navigation pane, choose **Dashboards**.

Step 3 Choose **DMS dashboard templates** under **Dashboard Templates** and click **DMS Rebalancing Log Center** to view the chart details.

- **Group ID.** The associated query and analysis statement is:
`select distinct("message.groupId")`

- Filter by rebalancing reason.
- **Consumer Groups for Rebalancing.** The associated query and analysis statement is:

```
select count(distinct("message.groupId")) as "total" from log where ("message.type"!='RESPONSE' and "message.type" !='REQUEST')
```
- **Rebalances.** The associated query and analysis statement is:

```
select count(*) as 'total' where ("message.type" !='RESPONSE' and "message.type" !='REQUEST')
```
- **Consumer Group Rebalances.** The associated query and analysis statement is:

```
select "message.groupId" as 'GroupId', count(*) as 'Count' where ('message.type' !='RESPONSE' and 'message.type' !='REQUEST') group by "message.groupId"
```
- **Rebalancing Reasons and Group Details.** The associated query and analysis statement is:

```
select __time as 'Time', "message.type" as 'Type', "message.groupId" as 'GroupId', "message.reason" as 'Reason', "message.group" as 'Group' where ('message.type' !='RESPONSE' and 'message.type' !='REQUEST')
```

----End

5.3.3.9 DSL Dashboard Template

LTS provides DSL processing for you to achieve one-stop log processing. Using domain-defined script languages and more than 200 built-in functions, you can implement end-to-end log processing tasks on the LTS console, such as log normalization, enrichment, transfer, anonymization, and filtering.

LTS provides the DSL processing task monitoring center dashboard template to display information such as the processing task ID/name and number of input/output lines.

Prerequisites

- A DSL processing task has been created.
- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

DSL Processing Task Monitoring Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **DSL dashboard templates** under **Dashboard Templates** and click **DSL processing task monitoring center** to view the chart details.

- Filter by processing task ID. The associated query and analysis statement is:

```
select distinct(task_id)
```
- Filter by processing task name. The associated query and analysis statement is:

```
select distinct(task_name)
```
- **Input Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "input" < 1000 THEN concat( cast( "input" AS VARCHAR ), 'Lines' ) WHEN "input" < 1000 * 1000 THEN concat( cast( round( "input"/ 1000, 1 ) AS VARCHAR ), 'Thousands lines' ) WHEN "input" < 1000000000 THEN concat( cast( round( "input"/ 1000000.0, 1 ) AS VARCHAR ), 'Million lines' ) WHEN "input"/ 1000.0 < 1000000000 THEN concat( cast( round( "input"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE concat( cast( round( "input"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END AS "total" from (select sum("process.accept") as "input")
```

- **Output Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "delivered" < 1000 THEN concat( cast( "delivered" AS VARCHAR ), 'Lines' )  
WHEN "delivered" < 1000 * 1000 THEN concat( cast( round( "delivered"/ 1000, 1 ) AS VARCHAR ), 'Thousands lines' )  
WHEN "delivered" < 1000000000 THEN concat( cast( round( "delivered"/ 1000000.0, 1 ) AS VARCHAR ), 'Million lines' )  
WHEN "delivered"/ 1000.0 < 1000000000 THEN concat( cast( round( "delivered"/ 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' )  
ELSE concat( cast( round( "delivered"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE  
concat( cast( round( "delivered"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' )  
END AS "total" from (select sum("process.delivered") as "delivered")
```
- **Filtered Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "drop" < 1000 THEN concat( cast( "drop" AS VARCHAR ), 'Lines' )  
WHEN "drop" < 1000 * 1000 THEN concat( cast( round( "drop"/ 1000, 1 ) AS VARCHAR ), 'Thousands lines' )  
WHEN "drop" < 1000000000 THEN concat( cast( round( "drop"/ 1000000.0, 1 ) AS VARCHAR ),  
'Million lines' )  
WHEN "drop"/ 1000.0 < 1000000000 THEN concat( cast( round( "drop"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' )  
ELSE concat( cast( round( "drop"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' )  
END AS "total" from (select sum("process.drop") as "drop")
```
- **Failed Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "failed" < 1000 THEN concat( cast( "failed" AS VARCHAR ), 'Lines' )  
WHEN "failed" < 1000 * 1000 THEN concat( cast( round( "failed"/ 1000, 1 ) AS VARCHAR ), 'Thousands lines' )  
WHEN "failed" < 1000000000 THEN concat( cast( round( "failed"/ 1000000.0, 1 ) AS  
VARCHAR ), 'Million lines' )  
WHEN "failed"/ 1000.0 < 1000000000 THEN  
concat( cast( round( "failed"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE  
concat( cast( round( "failed"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' )  
END AS "total" from (select sum("process.failed") as "failed")
```
- **Execution Records.** The associated query and analysis statement is:

```
select TIME_FORMAT( MILLIS_TO_TIMESTAMP("start"), 'yyyy-MM-dd HH:mm:ss:SSS', '+08:00') as  
"Started", TIME_FORMAT( MILLIS_TO_TIMESTAMP("end"), 'yyyy-MM-dd HH:mm:ss:SSS', '+08:00') as  
"Ended", "process.accept" as "Input Lines", "process.delivered" as "Output Lines", "process.drop" as  
"Filtered Lines", "process.failed" as "Failed Lines" limit 1000
```

----End

5.3.3.10 ER Dashboard Template

An enterprise router connects VPCs and on-premises networks to build a central hub network that features high-specification, high-bandwidth, and high-performance. Enterprise routers use the Border Gateway Protocol (BGP) to learn routes, dynamically select routes, or switch between connections, thereby significantly improving network scalability and O&M efficiency and ensuring service continuity.

LTS provides a one-stop wizard to collect enterprise router logs. It also supports multi-dimensional analysis and structuring, and offers a dashboard for enterprise router logs. This dashboard displays data such as the top 20 packets, top 20 traffic, and number of flow logs in enterprise router logs.

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Enterprise Router Flow Log Center

- Step 1** Log in to the LTS console.
- Step 2** In the navigation pane, choose **Dashboards**.
- Step 3** Choose **ER dashboard templates** under **Dashboard Templates** and click **Enterprise Router Flow Log Center** to view the chart details.

- Filter by instance ID. The associated query and analysis statement is:

```
SELECT DISTINCT(instance_id)
```
- Filter by attachment ID. The associated query and analysis statement is:

```
SELECT DISTINCT(resource_id)
```
- **Traffic Flow.** You can filter the traffic data by **ingress** and **egress**. The associated query and analysis statement is:
- Filter by source IP address. The associated query and analysis statement is:

```
SELECT DISTINCT(srcaddr)
```
- Filter by destination IP address. The associated query and analysis statement is:

```
SELECT DISTINCT(dstaddr)
```
- Filter by protocol. The associated query and analysis statement is:

```
SELECT DISTINCT(protocol)
```
- **Top 20 by Packets.** The associated query and analysis statement is:

```
SELECT "srcaddr" as "Source IP", "dstaddr" as "Destination IP", sum("packets") as "Packets",  
"resource_id" as "Attachment ID", "instance_id" as "Instance ID" group by "instance_id", "resource_id",  
"srcaddr", "dstaddr" order by "Packets" desc limit 20
```
- **Top 20 by Traffic.** The associated query and analysis statement is:

```
SELECT "srcaddr" as "Source IP", "dstaddr" as "Destination IP", sum("bytes") as "Bytes", "resource_id"  
as "Attachment ID", "instance_id" as "Instance ID" group by "instance_id", "resource_id", "srcaddr",  
"dstaddr" order by "Bytes" desc limit 20
```
- **Flow Logs.** The associated query and analysis statement is:

```
select time_series(_time, 'PT1H', 'yyyy-MM-dd HH:mm:ss', '0', '+08:00') as "Time", count(*) as "Flow  
Logs" group by "Time" order by "Time"
```
- **Flow Log Details.** The associated query and analysis statement is:

```
SELECT "instance_id" as "Instance ID", "resource_id" as "Attachment ID", "project_id" as "Project ID",  
"srcaddr" as "Source IP", "dstaddr" as "Destination IP", "srcport" as "Source Port", "dstport" as  
"Destination Port", "protocol" as "Protocol", "direct" as "Traffic Flow", "packets" as "Packets", "bytes"  
as "Bytes", TIME_FORMAT( MILLIS_TO_TIMESTAMP("start"*1000), 'yyyy-MM-dd HH:mm:ss', '+08:00')  
as "Started", TIME_FORMAT( MILLIS_TO_TIMESTAMP("end"*1000), 'yyyy-MM-dd HH:mm:ss',  
'+08:00') as "Ended"
```

----End

5.3.3.11 METRIC Dashboard Template

You can create metric rules on the LTS console to generate statistical reports as required. You can also set a single log filter or add associations and groups to set multiple log filters to retain logs that meet the filters. Statistics on the structured logs within a specified time range can be collected dynamically and displayed on the Prometheus instances of AOM, which is easy to operate and powerful.

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Monitoring Center for Metric Generation Tasks

- Step 1** Log in to the LTS console. In the navigation pane, choose **Dashboards**.
- Step 2** Choose **METRIC dashboard templates** under **Dashboard Templates** and click **DSL processing task monitoring center** to view the chart details.
 - Filter by rule ID. The associated query and analysis statement is:

```
select distinct(task_set)
```

- **Input Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "input" < 1000 THEN concat( cast( "input" AS VARCHAR ), 'Lines' ) WHEN
"input" < 1000 * 1000 THEN concat( cast( round( "input"/ 1000, 1 ) AS VARCHAR ), 'Thousands
lines' ) WHEN "input" < 1000000000 THEN concat( cast( round( "input"/ 1000000.0, 1 ) AS
VARCHAR ), 'Million lines' ) WHEN "input"/ 1000.0 < 1000000000 THEN
concat( cast( round( "input"/ 1000 / 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE
concat( cast( round( "input"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END
AS "total" from (select sum("input") as "input")
```
- **Output Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "output" < 1000 THEN concat( cast( "output" AS VARCHAR ), 'Lines' ) WHEN
"output" < 1000 * 1000 THEN concat( cast( round( "output"/ 1000, 1 ) AS VARCHAR ), 'Thousands
lines' ) WHEN "output" < 1000000000 THEN concat( cast( round( "output"/ 1000000.0, 1 ) AS
VARCHAR ), 'Million lines' ) WHEN "output"/ 1000.0 < 1000000000 THEN
concat( cast( round( "output"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE
concat( cast( round( "output"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END
AS "total" from (select sum("output") as "output")
```
- **Filter Criteria Met.** The associated query and analysis statement is:

```
SELECT CASE WHEN "filters" < 1000 THEN concat( cast( "filters" AS VARCHAR ), 'Lines' ) WHEN
"filters" < 1000 * 1000 THEN concat( cast( round( "filters"/ 1000, 1 ) AS VARCHAR ), 'Thousands
lines' ) WHEN "filters" < 1000000000 THEN concat( cast( round( "filters"/ 1000000.0, 1 ) AS
VARCHAR ), 'Million lines' ) WHEN "filters"/ 1000.0 < 1000000000 THEN
concat( cast( round( "filters"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE
concat( cast( round( "filters"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END
AS "total" from (select sum("filters") as "filters")
```
- **Filter Criteria Unmet.** The associated query and analysis statement is:

```
SELECT CASE WHEN "filter_drops" < 1000 THEN concat( cast( "filter_drops" AS VARCHAR ),
'Lines' ) WHEN "filter_drops" < 1000 * 1000 THEN concat( cast( round( "filter_drops"/ 1000, 1 ) AS
VARCHAR ), 'Thousands lines' ) WHEN "filter_drops" < 1000000000 THEN
concat( cast( round( "filter_drops"/ 1000000.0, 1 ) AS VARCHAR ), 'Million lines' ) WHEN
"filter_drops"/ 1000.0 < 1000000000 THEN concat( cast( round( "filter_drops"/ 1000 / 1000000.0, 1 )
AS VARCHAR ), 'Billion lines' ) ELSE concat( cast( round( "filter_drops"/ 1000.0 / 1000 / 1000 / 1000,
1 ) AS VARCHAR ), 'Trillion lines' ) END AS "total" from (select sum("filter_drops") as "filter_drops")
```
- **Sampled Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "samples" < 1000 THEN concat( cast( "samples" AS VARCHAR ), 'Lines' )
WHEN "samples" < 1000 * 1000 THEN concat( cast( round( "samples"/ 1000, 1 ) AS VARCHAR ),
'Thousands lines' ) WHEN "samples" < 1000000000 THEN concat( cast( round( "samples"/ 1000000.0,
1 ) AS VARCHAR ), 'Million lines' ) WHEN "samples"/ 1000.0 < 1000000000 THEN
concat( cast( round( "samples"/ 1000 / 1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE
concat( cast( round( "samples"/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END
AS "total" from (select sum("samples") as "samples")
```
- **Unsampled Lines.** The associated query and analysis statement is:

```
SELECT CASE WHEN "sample_drops" < 1000 THEN concat( cast( "sample_drops" AS VARCHAR ),
'Lines' ) WHEN "sample_drops" < 1000 * 1000 THEN concat( cast( round( "sample_drops"/ 1000, 1 )
AS VARCHAR ), 'Thousands lines' ) WHEN "sample_drops" < 1000000000 THEN
concat( cast( round( "sample_drops"/ 1000000.0, 1 ) AS VARCHAR ), 'Million lines' ) WHEN
"sample_drops"/ 1000.0 < 1000000000 THEN concat( cast( round( "sample_drops"/ 1000 / 1000000.0,
1 ) AS VARCHAR ), 'Billion lines' ) ELSE concat( cast( round( "sample_drops"/ 1000.0 / 1000 / 1000 /
1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END AS "total" from (select sum( "sample_drops" ) as
"sample_drops")
```
- **Lines Out of Log Time Range.** The associated query and analysis statement is:

```
SELECT CASE WHEN "out_of_bounds" < 1000 THEN concat( cast( "out_of_bounds" AS VARCHAR ),
'Lines' ) WHEN "out_of_bounds" < 1000 * 1000 THEN concat( cast( round( "out_of_bounds"/ 1000,
1 ) AS VARCHAR ), 'Thousands lines' ) WHEN "out_of_bounds" < 1000000000 THEN
concat( cast( round( "out_of_bounds"/ 1000000.0, 1 ) AS VARCHAR ), 'Million lines' ) WHEN
"out_of_bounds"/ 1000.0 < 1000000000 THEN concat( cast( round( "out_of_bounds"/ 1000 /
1000000.0, 1 ) AS VARCHAR ), 'Billion lines' ) ELSE concat( cast( round( "out_of_bounds"/ 1000.0 /
1000 / 1000 / 1000, 1 ) AS VARCHAR ), 'Trillion lines' ) END AS "total" from (select
sum("out_of_bounds") as "out_of_bounds")
```
- **Execute Records(Lines).** The associated query and analysis statement is:

```
select TIME_FORMAT( "__time", 'yyyy-MM-dd HH:mm:ss:SSS', '+08:00') as "Log Time", sum("input")
as "Input",sum("output") as "Output",sum("filters") as "Filter Criteria Met",sum("filter_drops") as
"Filter Criteria Unmet",sum("samples") as "Sampled",sum("sample_drops") as
```

```
"Unsampled",sum("out_of_bounds") as "Out of Log Time Range" group by __time order by __time  
desc limit 1000
```

----End

5.3.3.12 Nginx Dashboard Templates

LTS can collect Nginx logs and analyze them from multiple dimensions. LTS provides a one-stop wizard to collect Nginx logs. It also enables structuring and offers dashboards for Nginx logs. Nginx is a high-performance HTTP and reverse proxy web server that also provides IMAP, POP3, and SMTP services.

Nginx dashboard templates support [Viewing Nginx Monitoring by the Second](#), [Viewing Nginx Access Center](#), and [Viewing Nginx Monitoring Center](#).

Prerequisites

Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing Nginx Monitoring by the Second

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **NGINX dashboard templates** under **Dashboard Templates** and click **Nginx monitoring by the second** to view the chart details.

- **QPS.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT(TIME_CEIL(__time,'PT15S'),'yyyy-MM-dd HH:mm:ss','+08:00') AS __time_,  
COUNT(*) as QPS from log group by __time_
```
- **Success Rate.** The associated query and analysis statement is:

```
select __time,round(CASE WHEN "Success Rate" > 0 THEN "Success Rate" else 0 end,2) as "Success  
Rate" from (select TIME_FORMAT(TIME_CEIL(__time,'PT5S'),'yyyy-MM-dd HH:mm:ss','+08:00') as  
__time, sum(case when status < 400 then 1 else 0 end) * 100.0 / count(1) as 'Success Rate' from log  
group by __time)
```
- **Latency.** The associated query and analysis statement is:

```
select __time,round(CASE WHEN "Access latency" > 0 THEN "Access latency" else 0 end,2) as "Access  
latency",round(CASE WHEN "Upstream latency" > 0 THEN "Upstream latency" else 0 end,2) as  
"Upstream latency" from (select TIME_FORMAT(TIME_CEIL(__time,'PT5S'),'yyyy-MM-dd  
HH:mm:ss','+08:00') as __time, avg(request_time)* 1000 as 'Access  
latency',avg(upstream_response_time)* 1000 as 'Upstream latency' from log group by __time)
```
- **Traffic.** The associated query and analysis statement is:

```
select TIME_FORMAT(TIME_CEIL(__time,'PT5S'),'yyyy-MM-dd HH:mm:ss','+08:00') as __time ,  
sum("request_length") as "Incoming", sum("body_bytes_sent") as "Outgoing body" group by __time
```
- **Status Codes.** The associated query and analysis statement is:

```
select t.t as "time",  
CASE WHEN a."2XX" IS NOT NULL THEN CAST(a."2XX" AS BIGINT) ELSE 0 END as "2XX",  
CASE WHEN b."3XX" IS NOT NULL THEN CAST(b."3XX" AS BIGINT) ELSE 0 END as "3XX",  
CASE WHEN c."4XX" IS NOT NULL THEN CAST(c."4XX" AS BIGINT) ELSE 0 END as "4XX",  
CASE WHEN d."5XX" IS NOT NULL THEN CAST(d."5XX" AS BIGINT) ELSE 0 END as "5XX",  
CASE WHEN e."Other" IS NOT NULL THEN CAST(e."Other" AS BIGINT) ELSE 0 END as "Other"  
from (select TIME_CEIL(__time,'PT5S') as t from log group by t order by t asc ) t left join (select  
TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "2XX" from log WHERE "status" >=  
200 and "status" < 300 group by t order by t asc ) a on t.t =a.t left join (select  
TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "3XX" from log WHERE "status" >=  
300 and "status" < 400 group by t order by t asc) b on t.t =b.t left join (select  
TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "4XX" from log WHERE "status" >=  
400 and "status" < 500 group by t order by t asc) c on t.t =c.t left join (select  
TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "5XX" from log WHERE "status" >=  
500 and "status" < 600 group by t order by t asc) d on t.t =d.t left join (select  
TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "Other" from log WHERE "status" <  
200 or "status" >= 600 group by t order by t asc) e on t.t =e.t
```

- **Backend Response Codes.** The associated query and analysis statement is:

```
select t.t as "time",
CASE WHEN a."2XX" IS NOT NULL THEN CAST(a."2XX" AS BIGINT) ELSE 0 END as "2XX",
CASE WHEN b."3XX" IS NOT NULL THEN CAST(b."3XX" AS BIGINT) ELSE 0 END as "3XX",
CASE WHEN c."4XX" IS NOT NULL THEN CAST(c."4XX" AS BIGINT) ELSE 0 END as "4XX",
CASE WHEN d."5XX" IS NOT NULL THEN CAST(d."5XX" AS BIGINT) ELSE 0 END as "5XX",
CASE WHEN e."Other" IS NOT NULL THEN CAST(e."Other" AS BIGINT) ELSE 0 END as "Other"
from (
select TIME_CEIL(__time,'PT5S') as t from log group by t order by t asc
) t
left join(
select TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "2XX" from log WHERE
"upstream_status" >= 200 and "upstream_status" < 300 group by t order by t asc) a
on t.t = a.t
left join (
select TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "3XX" from log WHERE
"upstream_status" >= 300 and "upstream_status" < 400 group by t order by t asc) b
on t.t =b.t
left join (
select TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "4XX" from log WHERE
"upstream_status" >= 400 and "upstream_status" < 500 group by t order by t asc) c
on t.t =c.t
left join (
select TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "5XX" from log WHERE
"upstream_status" >= 500 and "upstream_status" < 600 group by t order by t asc) d
on t.t =d.t
left join (
select TIME_CEIL(__time,'PT5S') as t , CAST(COUNT(1) as VARCHAR) as "Other" from log WHERE
"upstream_status" < 200 or "upstream_status" >= 600 group by t order by t asc) e
on t.t =e.t
```

----End

Viewing Nginx Access Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **NGINX dashboard templates** under **Dashboard Templates** and click **Nginx access center** to view the chart details.

- **Day-over-day PV Change.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare("pv" , 86400) as diff from (select count(1) as "pv" from log))
- **Week-on-week PV Change.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare("pv" , 604800) as diff from (select count(1) as "pv" from log))
- **Day-over-day UV Change.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare("uv" , 86400) as diff from (select APPROX_COUNT_DISTINCT(my_remote_addr) as "uv" from log))
- **Week-on-week UV Change.** The associated query and analysis statement is:
select diff[1] as "total", round((diff[1] - diff[2]) / diff[2] * 100, 2) as inc from(select compare("uv" , 604800) as diff from (select APPROX_COUNT_DISTINCT(my_remote_addr) as "uv" from log))
- **PV Distribution (China).** The associated query and analysis statement is:
select ip_to_province(remote_addr) as province, count(1) as pv where IP_TO_COUNTRY (remote_addr) = 'China' group by province HAVING province not in ('','Reserved address','*')
- **PV Distribution (Global).** The associated query and analysis statement is:
SELECT ip_to_country(remote_addr) as country,COUNT(1) as PV GROUP BY country HAVING country not in ('','Reserved address','*')
- **UV Distribution (China).** The associated query and analysis statement is:
select ip_to_province(remote_addr) as province, APPROX_COUNT_DISTINCT(remote_addr) as UV where IP_TO_COUNTRY (remote_addr) = 'China' group by province HAVING province not in ('','Reserved address','*')

- **UV Distribution (Global).** The associated query and analysis statement is:
select ip_to_country(remote_addr) as country, APPROX_COUNT_DISTINCT(remote_addr) as uv group by country HAVING country not in ('Reserved address',*)
- **Average Latency Distribution (China).** The associated query and analysis statement is:
SELECT province,round(CASE WHEN "Average latency (ms)" > 0 THEN "Average latency (ms)" ELSE 0 END, 3) AS "Average latency (ms)"FROM (SELECT ip_to_province(remote_addr) as province,avg(request_time) * 1000 AS "Average latency (ms)"WHERE IP_TO_COUNTRY (remote_addr) = 'China'GROUP BY province HAVING province not in ('Reserved address',*))
- **Average Latency Distribution (Global).** The associated query and analysis statement is:
SELECT country,round(CASE WHEN "Average latency (ms)" > 0 THEN "Average latency (ms)" ELSE 0 END, 2) AS "Average latency (ms)"FROM (SELECT ip_to_country(remote_addr) as country,avg(request_time) * 1000 AS "Average latency (ms)" GROUP BY country HAVING country not in ('Reserved address',*))
- **PV/UV Today.** The associated query and analysis statement is:
SELECT TIME_FORMAT(_time_, 'yyyy-MM-dd HH:mm:ss', '+08:00') as _time_,PV,UV FROM (select TIME_CEIL(_time_,PT600S) AS _time_, count(1) as PV, APPROX_COUNT_DISTINCT(my_remote_addr) as UV from log WHERE __time <= CURRENT_TIMESTAMP and __time >= DATE_TRUNC('DAY', (CURRENT_TIMESTAMP + INTERVAL '8' HOUR)) - INTERVAL '8' HOUR group by _time_ order by _time_) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100000 OFFSET 1
- **PV/UV in 7 Days.** The associated query and analysis statement is:
SELECT TIME_FORMAT(_time_, 'yyyy-MM-dd HH:mm:ss', '+08:00') as _time_,PV,UV FROM (select TIME_CEIL(_time_,PT600S) AS _time_, count(1) as PV, APPROX_COUNT_DISTINCT(remote_addr) as UV from log WHERE __time <= CURRENT_TIMESTAMP and __time >= DATE_TRUNC('DAY', (CURRENT_TIMESTAMP + INTERVAL '8' HOUR)) - INTERVAL '8' HOUR - INTERVAL '7' DAY group by _time_ order by _time_) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100000 OFFSET 1
- **Top 10 Provinces by Visits.** The associated query and analysis statement is:
select ip_to_province(remote_addr) as "province", count(1) as "Visits" group by "province" HAVING "province" <> '-1' order by "Visits" asc limit 10
- **Top 10 Cities by Visits.** The associated query and analysis statement is:
select ip_to_city(remote_addr) as "city", count(1) as "Visits" group by "city" HAVING "city" <> '-1' order by "Visits" asc limit 10
- **Top 10 Hosts by Visits.** The associated query and analysis statement is:
select host as "Host", count(1) as "PV" group by "Host" order by "PV" asc limit 10
- **Top 10 UserAgents by Visits.** The associated query and analysis statement is:
select http_user_agent as "UserAgent", count(1) as "PV" group by "UserAgent" order by "PV" asc limit 10
- **Device Distribution by Type.** The associated query and analysis statement is:
select case when regexp_like(lower(http_user_agent), 'iphone|ipod|android|ios') then 'Mobile' else 'PC' end as type , count(1) as total group by type
- **Device Distribution by System.** The associated query and analysis statement is:
select case when regexp_like(lower(http_user_agent), 'iphone|ipod|ios') then 'IOS' when regexp_like(lower(http_user_agent), 'android') then 'Android' else 'other' end as type , count(1) as total group by type HAVING type != 'other'
- **TOP URL.** The associated query and analysis statement is:
select request_uri , count(1) as PV, APPROX_COUNT_DISTINCT(remote_addr) as UV, round(sum(case when status < 400 then 1 else 0 end) * 100.0 / count(1), 2) as "Access Success Rate" group by request_uri ORDER by PV desc
- **Top IP Addresses by Visits.** The associated query and analysis statement is:
select remote_addr as "Source IP Address",ip_to_country(remote_addr) as "Country/Region",ip_to_province(remote_addr) as "Province",ip_to_city(remote_addr) as "City",ip_to_provider(remote_addr) as "Carrier",count(1) as "PV",http_user_agent as "UserAgent Sampling",request_uri as "URL Sampling" group by remote_addr,http_user_agent,request_uri ORDER by "PV" desc

----End

Viewing Nginx Monitoring Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **NGINX dashboard templates** under **Dashboard Templates** and click **Nginx monitoring center** to view the chart details.

- **PV.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss', '+08:00' ) as _time_ , PV FROM ( SELECT TIME_CEIL( __time_ , 'PT300S' ) AS _time_ , count( 1 ) AS PV FROM log GROUP BY _time_ ) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100 OFFSET 1
```
- **Request Success Rate.** The associated query and analysis statement is:

```
select ROUND(sum(case when status < 400 then 1 else 0 end) * 100.0 / count(1),2) as cnt
```
- **Average Latency.** The associated query and analysis statement is:

```
select round(avg(request_time) * 1000, 3) as cnt
```
- **4xx Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "status" >= 400 and "status" < 500
```
- **404 Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "status" = 404
```
- **429 Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "status" = 429
```
- **504 Requests.** The associated query and analysis statement is:

```
SELECT COUNT(1) as cnt WHERE "status" = 504
```
- **5xx Requests.** The associated query and analysis statement is:

```
select TIME_FORMAT(TIME_CEIL(__time_,'PT300S'),'yyyy-MM-dd HH:mm:ss','+08:00') AS _time_ , count(1) as cnt where "status" >= 500 group by _time_
```
- **Status Code Distribution.** The associated query and analysis statement is:

```
SELECT status, COUNT(1) AS rm GROUP BY status
```
- **UV.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss', '+08:00' ) as _time_ , UV FROM (select TIME_CEIL( __time_ , 'PT600S' ) AS _time_ , APPROX_COUNT_DISTINCT( remote_addr ) as UV from log group by _time_ ) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100000 OFFSET 1
```
- **Traffic.** The associated query and analysis statement is:

```
select TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss', '+08:00' ) AS _time_ , round( CASE WHEN "Inbound" > 0 THEN "Inbound" ELSE 0 END, 2 ) AS "Inbound" , round( CASE WHEN "Outbound" > 0 THEN "Outbound" ELSE 0 END, 2 ) AS "Outbound" FROM (SELECT TIME_CEIL( __time_ , 'PT600S' ) AS _time_ , sum( request_length ) / 1024.0 AS "Inbound" , sum( bytes_sent ) / 1024.0 AS "Outbound" group by _time_ ) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100000 OFFSET 1
```
- **Access Failure Rate.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss', '+08:00' ) as _time_ , round( CASE WHEN "Failure rate" > 0 THEN "Failure rate" ELSE 0 END, 2 ) AS "Failure rate" , round( CASE WHEN "5XX" > 0 THEN "5XX" ELSE 0 END, 2 ) AS "5XX" from (select TIME_CEIL( __time_ , 'PT600S' ) AS _time_ , sum( case when status >= 400 then 1 else 0 end ) * 100.0 / count(1) as 'Failure rate' , sum( case when status >= 500 then 1 else 0 end ) * 100.0 / COUNT(1) as '5XX' group by _time_ ) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100000 OFFSET 1
```
- **Latency.** The associated query and analysis statement is:

```
select TIME_FORMAT( _time_ , 'yyyy-MM-dd HH:mm:ss', '+08:00' ) as _time_ , round( CASE WHEN "Avg." > 0 THEN "Avg." ELSE 0 END, 2 ) AS "Avg." , round( CASE WHEN "P50" > 0 THEN "P50" ELSE 0 END, 2 ) AS "P50" , round( CASE WHEN "P90" > 0 THEN "P90" ELSE 0 END, 2 ) AS "P90" , round( CASE WHEN "P99" > 0 THEN "P99" ELSE 0 END, 2 ) AS "P99" , round( CASE WHEN "P9999" > 0 THEN "P9999" ELSE 0 END, 2 ) AS "P9999" from (select TIME_CEIL( __time_ , 'PT600S' ) as _time_ , avg( request_time ) * 1000 as "Avg." , APPROX_QUANTILE_DS( "request_time" , 0.50 ) * 1000 as "P50" , APPROX_QUANTILE_DS( "request_time" , 0.90 ) * 1000 as "P90" , APPROX_QUANTILE_DS( "request_time" , 0.99 ) * 1000 as "P99" , APPROX_QUANTILE_DS( "request_time" , 0.9999 ) * 1000 as "P9999" group by _time_ ) WHERE _time_ <= CURRENT_TIMESTAMP LIMIT 100000 OFFSET 1
```
- **Top Host Requests.** The associated query and analysis statement is:

```
SELECT "host" , pv , uv , round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)" , round( CASE WHEN "Average Latency (ms)" > 0
```

```
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN
"Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN
"Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT
"host", count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( remote_addr ) AS uv, sum( CASE WHEN
"status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)",
avg( request_time ) * 1000 AS "Average Latency (ms)", sum( request_length ) / 1024.0 AS "Inbound
(KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "host" != " GROUP BY "host" )
ORDER BY pv DESC
```

- Top Host Latencies.** The associated query and analysis statement is:

```
SELECT "host", pv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate
(%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90
Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE
WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)"
FROM ( SELECT "host", count( 1 ) AS pv, sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END ) *
100.0 / count( 1 ) AS "Access Success Rate (%)", avg( request_time ) * 1000 AS "Average Latency
(ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",
APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != "
GROUP BY "host" ) ORDER BY "Average Latency (ms)" desc
```
- Top Host Failure Rates.** The associated query and analysis statement is:

```
SELECT "host", pv, round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)"
ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0 THEN
"Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90
Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE
WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)"
FROM ( SELECT "host", count( 1 ) AS pv, sum( CASE WHEN "status" >= 400 THEN 1 ELSE 0 END ) *
100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( request_time ) * 1000 AS "Average Latency
(ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",
APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != "
GROUP BY "host" ) ORDER BY "Access Failure Rate (%)" desc
```
- Top URL Requests.** The associated query and analysis statement is:

```
SELECT request_uri, pv, uv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success
Rate (%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)"
> 0 THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN
"Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN
"Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT
request_uri, count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( remote_addr ) AS uv, sum( CASE WHEN
"status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)",
avg( request_time ) * 1000 AS "Average Latency (ms)", sum( request_length ) / 1024.0 AS "Inbound
(KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "host" != " GROUP BY request_uri )
ORDER BY pv desc
```
- Top URL Latencies.** The associated query and analysis statement is:

```
SELECT request_uri, pv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate
(%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90
Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE
WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)"
FROM ( SELECT request_uri, count( 1 ) AS pv, sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END )
* 100.0 / count( 1 ) AS "Access Success Rate (%)", avg( request_time ) * 1000 AS "Average Latency
(ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",
APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != "
GROUP BY request_uri ) ORDER BY "Average Latency (ms)" desc
```
- Top URL Failure Rates.** The associated query and analysis statement is:

```
SELECT request_uri, pv, round( CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate
(%)" ELSE 0 END, 2 ) AS "Access Failure Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "P90
Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3 ) AS "P90 Latency (ms)", round( CASE
WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3 ) AS "P99 Latency (ms)"
FROM ( SELECT request_uri, count( 1 ) AS pv, sum( CASE WHEN "status" >= 400 THEN 1 ELSE 0 END )
* 100.0 / count( 1 ) AS "Access Failure Rate (%)", avg( request_time ) * 1000 AS "Average Latency
(ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",
APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " GROUP
BY request_uri ) ORDER BY "Access Failure Rate (%)" desc
```
- Top Backend Requests.** The associated query and analysis statement is:

```
SELECT addr, pv, uv, round( CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate
(%)" ELSE 0 END, 2 ) AS "Access Success Rate (%)", round( CASE WHEN "Average Latency (ms)" > 0
```

```
THEN "Average Latency (ms)" ELSE 0 END, 3 ) AS "Average Latency (ms)", round( CASE WHEN "Inbound (KB)" > 0 THEN "Inbound (KB)" ELSE 0 END, 3 ) AS "Inbound (KB)", round( CASE WHEN "Outbound (KB)" > 0 THEN "Outbound (KB)" ELSE 0 END, 3 ) AS "Outbound (KB)" FROM ( SELECT upstream_addr as addr, count( 1 ) AS pv, APPROX_COUNT_DISTINCT ( remote_addr ) AS uv, sum( CASE WHEN "status" < 400 THEN 1 ELSE 0 END ) * 100.0 / count( 1 ) AS "Access Success Rate (%)", avg( request_time ) * 1000 AS "Average Latency (ms)", sum( request_length ) / 1024.0 AS "Inbound (KB)", sum( bytes_sent ) / 1024.0 AS "Outbound (KB)" WHERE "host" != " GROUP BY addr having length(upstream_addr) > 2) ORDER BY "pv" desc
```

- **Top Backend Latencies.** The associated query and analysis statement is:
SELECT addr,pv,round(CASE WHEN "Access Success Rate (%)" > 0 THEN "Access Success Rate (%)" ELSE 0 END, 2) AS "Access Success Rate (%)",round(CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3) AS "Average Latency (ms)",round(CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3) AS "P90 Latency (ms)",round(CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3) AS "P99 Latency (ms)" FROM (SELECT upstream_addr as addr,count(1) AS pv,sum(CASE WHEN "status" < 400 THEN 1 ELSE 0 END) * 100.0 / count(1) AS "Access Success Rate (%)",avg(request_time) * 1000 AS "Average Latency (ms)",APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)",APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " GROUP BY addr having length(upstream_addr) > 2) ORDER BY "Average Latency (ms)" desc
- **Top Backend Failure Rates.** The associated query and analysis statement is:
SELECT addr, pv, round(CASE WHEN "Access Failure Rate (%)" > 0 THEN "Access Failure Rate (%)" ELSE 0 END, 2) AS "Access Failure Rate (%)", round(CASE WHEN "Average Latency (ms)" > 0 THEN "Average Latency (ms)" ELSE 0 END, 3) AS "Average Latency (ms)", round(CASE WHEN "P90 Latency (ms)" > 0 THEN "P90 Latency (ms)" ELSE 0 END, 3) AS "P90 Latency (ms)", round(CASE WHEN "P99 Latency (ms)" > 0 THEN "P99 Latency (ms)" ELSE 0 END, 3) AS "P99 Latency (ms)" FROM (SELECT upstream_addr as addr, count(1) AS pv, sum(CASE WHEN "status" >= 400 THEN 1 ELSE 0 END) * 100.0 / count(1) AS "Access Failure Rate (%)", avg(request_time) * 1000 AS "Average Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.9) * 1000 AS "P90 Latency (ms)", APPROX_QUANTILE_DS(request_time, 0.99) * 1000 AS "P99 Latency (ms)" WHERE "host" != " GROUP BY addr having length(upstream_addr) > 2)ORDER BY "Access Failure Rate (%)" desc

----End

5.3.3.13 VPC Dashboard Template

A VPC is an isolated and private virtual network environment. You can configure the IP address ranges, subnets, and security groups, assign EIPs, and allocate bandwidth for a VPC. A VPC flow log records traffic information. It helps you monitor network traffic, analyze network attacks, and validate security group and Access Control List (ACL) rules.

LTS provides a one-stop wizard to collect VPC logs. It also enables structuring and offers a dashboard for VPC logs. The **VPC Flow Logs** dashboard displays the total number of actions, accepted bytes/packets, rejected bytes/packets, action sources, number of actions per minute, action distribution, flow log records distribution by status, carrier distribution of action sources, top 5 source/destination addresses by bytes, packets per minute, and ENIs of each protocol.

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing VPC Flow Logs

Step 1 Log in to the LTS console. In the navigation pane, choose **Log Management**.

Step 2 In the **Log Applications** area, click **VPC Flow Log Center** and choose **Go to the Dashboard**.

Step 3 Choose **VPC dashboard templates** under **Dashboard Templates** and click **VPC Flow Logs** to view the chart details.

- **Total Actions.** The associated query and analysis statement is:

```
select CASE WHEN total_actions < 1000 THEN concat(cast( total_actions AS VARCHAR), 'Actions')
WHEN total_actions < 1000 * 1000 THEN concat(cast(round(total_actions / 1000.0, 2) AS
VARCHAR),'Thousand actions') WHEN total_actions < 1000000000 THEN
concat(cast(round(total_actions / 1000000.0, 2) AS VARCHAR),'Million actions') WHEN total_actions /
1000.0 < 1000000000 THEN concat(cast(round(total_actions / 1000 / 1000000.0, 1) AS
VARCHAR),'Billion actions') ELSE concat(cast(round(total_actions / 1000.0 / 1000 / 1000 / 1000, 1) AS
VARCHAR),'Trillion actions') END AS "total_actions" from (select count(1) as total_actions where
log_status='OK' and version=1)
```
- **Total Accepted Bytes.** The associated query and analysis statement is:

```
select CASE WHEN accept_bytes < 1024 THEN concat(cast( accept_bytes AS VARCHAR), 'B') WHEN
accept_bytes < 1024 * 1024 THEN concat(cast(round(accept_bytes / 1024, 2) AS VARCHAR),'KB')
WHEN accept_bytes < 1000000000 THEN concat(cast(round(accept_bytes /1024.0 /1024, 2) AS
VARCHAR),'MB') WHEN accept_bytes / 1000.0 < 1000000000 THEN concat(cast(round(accept_bytes /
1024 / 1000000.0, 2) AS VARCHAR),'GB') ELSE concat(cast(round(accept_bytes / 1000.0 / 1000 /
1000 / 1000, 1) AS VARCHAR),'TB') END AS "accept_bytes" from (select sum(bytes) as accept_bytes
where log_status='OK' and version=1 and action='ACCEPT')
```
- **Total Accepted Packets.** The associated query and analysis statement is:

```
select CASE WHEN accept_packets < 1024 THEN concat(cast( accept_packets AS VARCHAR), 'B')
WHEN accept_packets < 1024 * 1024 THEN concat(cast(round(accept_packets / 1024, 2) AS
VARCHAR),'KB') WHEN accept_packets < 1000000000 THEN concat(cast(round(accept_packets /
1024.0 /1024, 2) AS VARCHAR),'MB') WHEN accept_packets / 1000.0 < 1000000000 THEN
concat(cast(round(accept_packets / 1024 / 1000000.0, 2) AS VARCHAR),'GB') ELSE
concat(cast(round(accept_packets / 1000.0 / 1000 / 1000 / 1000, 1) AS VARCHAR),'TB') END AS
"accept_packets" from (select sum(packets) as accept_packets where log_status='OK' and version=1
and action='ACCEPT')
```
- **Total Rejected Bytes.** The associated query and analysis statement is:

```
select CASE WHEN reject_bytes < 1024 THEN concat(cast( reject_bytes AS VARCHAR), 'B') WHEN
reject_bytes < 1024 * 1024 THEN concat(cast(round(reject_bytes / 1024, 2) AS VARCHAR),'KB') WHEN
reject_bytes < 1000000000 THEN concat(cast(round(reject_bytes /1024.0 /1024, 2) AS
VARCHAR),'MB') WHEN reject_bytes / 1000.0 < 1000000000 THEN concat(cast(round(reject_bytes /
1024 / 1000000.0, 2) AS VARCHAR),'GB') ELSE concat(cast(round(reject_bytes / 1000.0 / 1000 / 1000 /
1000, 1) AS VARCHAR),'TB') END AS "reject_bytes" from (select sum(bytes) as reject_bytes where
log_status='OK' and version=1 and action='REJECT')
```
- **Total Rejected Packets.** The associated query and analysis statement is:

```
select CASE WHEN reject_packets < 1024 THEN concat(cast( reject_packets AS VARCHAR), 'B') WHEN
reject_packets < 1024 * 1024 THEN concat(cast(round(reject_packets / 1024, 2) AS VARCHAR),'KB')
WHEN reject_packets < 1000000000 THEN concat(cast(round(reject_packets /1024.0 /1024, 2) AS
VARCHAR),'MB') WHEN reject_packets / 1000.0 < 1000000000 THEN
concat(cast(round(reject_packets / 1024 / 1000000.0, 2) AS VARCHAR),'GB') ELSE
concat(cast(round(reject_packets / 1000.0 / 1000 / 1000 / 1000, 1) AS VARCHAR),'TB') END AS
"reject_packets" from (select sum(packets) as reject_packets where log_status='OK' and version=1
and action='REJECT')
```
- **Action Sources.** The associated query and analysis statement is:

```
select IP_TO_PROVINCE(srcaddr) as province, count(1) as total_actions where IP_TO_COUNTRY
(srcaddr) = 'China' group by province HAVING province not in ('','Reserved address','*')
```
- **Actions/Min.** The associated query and analysis statement is:

```
select TIME_FORMAT(date_trunc('minute', MILLIS_TO_TIMESTAMP("start" * 1000)), 'MM-dd HH:mm')
as "t", "action", count(1) as "total_actions" where log_status='OK' and version=1 group by "t",
"action" order by t asc limit 1000
```
- **Action Results.** The associated query and analysis statement is:

```
select action, count(1) as total_actions where log_status='OK' and version=1 group by action
```
- **Flow Log Records By Status.** The associated query and analysis statement is:

```
select log_status, count(1) as total_actions where version=1 group by log_status
```
- **Carriers of Action Source.** The associated query and analysis statement is:

```
select ip_to_provider(srcaddr) as src_addr_provider, count(1) as total_actions where log_status='OK'
and version=1 group by src_addr_provider order by total_actions desc limit 5
```
- **Top 5 Sources by Bytes.** The associated query and analysis statement is:

```
select ip_to_provider(srcaddr) as src_addr_provider, count(1) as total_actions where log_status='OK'
and version=1 group by src_addr_provider order by total_actions desc limit 5
```

- **Top 5 Destinations by Bytes.** The associated query and analysis statement is:
select dstaddr, sum(bytes) as total_bytes where log_status='OK' and version=1 group by dstaddr order by total_bytes desc limit 5
- **Top 5 Destination Ports by Packets.** The associated query and analysis statement is:
select dstport, sum(packets) as total_packets where log_status='OK' and version=1 group by dstport order by total_packets desc limit 5
- **Packets/Min by Protocol.** The associated query and analysis statement is:
select TIME_FORMAT(date_trunc('minute', MILLIS_TO_TIMESTAMP("start" * 1000)), 'MM-dd HH:mm') as t, protocol, sum(packets) as total_packets where log_status='OK' and version=1 group by t, protocol order by t asc limit 1000
- **ENIs.** The associated query and analysis statement is:
select interface_id as "ID", sum(packets) as 'Data Packets', sum(bytes) as 'Data Packet Size' where log_status='OK' and version=1 group by "ID"

----End

5.3.3.14 WAF Dashboard Templates

WAF keeps web services stable and secure. It examines all HTTP and HTTPS requests to detect and block the following attacks: Structured Query Language (SQL) injection, cross-site scripting (XSS), web shells, command and code injections, file inclusion, sensitive file access, third-party vulnerability exploits, Challenge Collapsar (CC) attacks, malicious crawlers, and cross-site request forgery (CSRF).

For [Log Search and Analysis](#), the corresponding WAF dashboard templates are those for [Viewing WAF Security Log Center](#) and [Viewing WAF Access Log Center](#).

Prerequisites

- Logs have been structured. For details, see [Setting Cloud Structuring Parsing](#).

Viewing WAF Security Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **WAF dashboard templates** under **Dashboard Templates** and click **WAF security log center** to view the chart details.

- **Attacked Websites.** The associated query and analysis statement is:

```
SELECT diff [ 1 ] AS "VALUE", COALESCE ( diff [ 1 ]- diff [ 2 ], 0 ) AS "BEFORE" FROM
(
  SELECT
  compare ( "DATA", 86400 ) AS diff
  FROM
  ( SELECT count( DISTINCT "host" ) AS "DATA" FROM log
  WHERE action != "
  ))
```
- **Originating Countries.** The associated query and analysis statement is:

```
SELECT
diff [ 1 ] AS
"VALUE"
,
COALESCE ( diff [ 1 ]- diff [ 2 ], 0 ) AS "BEFORE"
FROM
(
  SELECT
```

```
compare ( "DATA", 86400 ) AS diff
FROM
( SELECT count( DISTINCT ip_to_country ( CASE WHEN sip = '-' THEN remote_ip ELSE sip END) )
AS "DATA" FROM log
WHERE action != "
)
)
```

- **Web Attacks Intercepted.** The associated query and analysis statement is:

```
SELECT
CASE
WHEN
diff [ 1 ] < 1000 THEN
concat( cast( diff [ 1 ] AS VARCHAR ), ' attacks' )
WHEN diff [ 1 ] < 1000 * 1000 THEN
concat( cast( round( diff [ 1 ]/ 1000, 1 ) AS VARCHAR ), ' thousand attacks' )
WHEN diff [ 1 ] < 1000000000 THEN
concat( cast( round( diff [ 1 ]/ 1000000.0, 1 ) AS VARCHAR ), ' million attacks' )
WHEN diff [ 1 ]/ 1000.0 < 1000000000 THEN
concat( cast( round( diff [ 1 ]/ 1000.0 / 1000000, 1 ) AS VARCHAR ), ' billion attacks' ) ELSE
concat( cast( round( diff [ 1 ]/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), ' trillion attacks' )
END AS
"value"
,
CASE WHEN diff [ 2 ]= 0 THEN 0 ELSE round( diff [ 3 ]- 1, 2 ) END AS ratio
FROM
( SELECT compare ( "data", 86400 ) AS diff FROM ( SELECT count( 1 ) AS "data" FROM log
WHERE action = " ) )
```

- **CC Attacks Intercepted.** The associated query and analysis statement is:

```
SELECT
CASE
WHEN
diff [ 1 ] < 1000 THEN
concat( cast( diff [ 1 ] AS VARCHAR ), ' attacks' )
WHEN diff [ 1 ] < 1000 * 1000 THEN
concat( cast( round( diff [ 1 ]/ 1000, 1 ) AS VARCHAR ), ' thousand attacks' )
WHEN diff [ 1 ] < 1000000000 THEN
concat( cast( round( diff [ 1 ]/ 1000000.0, 1 ) AS VARCHAR ), ' million attacks' )
WHEN diff [ 1 ]/ 1000.0 < 1000000000 THEN
concat( cast( round( diff [ 1 ]/ 1000.0 / 1000000, 1 ) AS VARCHAR ), ' billion attacks' ) ELSE
concat( cast( round( diff [ 1 ]/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), ' trillion attacks' )
END AS
"value"
,
CASE WHEN diff [ 2 ]= 0 THEN 0 ELSE round( diff [ 3 ]- 1, 2 ) END AS ratio
FROM
( SELECT compare ( "data", 86400 ) AS diff FROM ( SELECT count( 1 ) AS "data" FROM log
WHERE attack != 'default' ) )
```

- **Attacker UV.** The associated query and analysis statement is:

```
SELECT
CASE
WHEN
diff [ 1 ] < 1000 THEN
concat( cast( cast ( diff [ 1 ] AS INTEGER ) AS VARCHAR ), ' attackers' )
WHEN diff [ 1 ] < 1000 * 1000 THEN
concat( cast( round( diff [ 1 ]/ 1000, 1 ) AS VARCHAR ), ' thousand attackers' )
WHEN diff [ 1 ] < 1000000000 THEN
concat( cast( round( diff [ 1 ]/ 1000000.0, 1 ) AS VARCHAR ), ' million attackers' )
WHEN diff [ 1 ]/ 1000.0 < 1000000000 THEN
concat( cast( round( diff [ 1 ]/ 1000.0 / 1000000, 1 ) AS VARCHAR ), ' billion attackers' ) ELSE
concat( cast( round( diff [ 1 ]/ 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), ' trillion attackers' )
END AS "value",
CASE WHEN diff [ 2 ]= 0 THEN 0 ELSE round( diff [ 3 ]- 1, 2 ) END AS ratio
FROM
(
SELECT
compare ( "data", 86400 ) AS diff
```

```
FROM
( SELECT count( DISTINCT CASE WHEN sip = '-' THEN remote_ip ELSE sip END ) AS "data"
FROM log
))
```

- **Total Attacks Intercepted.** The associated query and analysis statement is:

```
SELECT
CASE
WHEN
diff [ 1 ] < 1000 THEN
concat( cast( diff [ 1 ] AS VARCHAR ), ' attacks' )
WHEN diff [ 1 ] < 1000 * 1000 THEN
concat( cast( round( diff [ 1 ] / 1000, 1 ) AS VARCHAR ), ' thousand attacks' )
WHEN diff [ 1 ] < 1000000000 THEN
concat( cast( round( diff [ 1 ] / 1000000.0, 1 ) AS VARCHAR ), ' million attacks' )
WHEN diff [ 1 ] / 1000.0 < 1000000000 THEN
concat( cast( round( diff [ 1 ] / 1000.0 / 1000000, 1 ) AS VARCHAR ), ' billion attacks' ) ELSE
concat( cast( round( diff [ 1 ] / 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), ' trillion attacks' )
END AS
"value",
CASE WHEN diff [ 2 ] = 0 THEN 0 ELSE round( diff [ 3 ] - 1, 2 ) END AS "ratio"
FROM
(
SELECT
compare ( "data", 86400 ) AS diff
FROM
( SELECT count( 1 ) AS "data" FROM log WHERE action != " )
)
```

- **CC Attacks.** The associated query and analysis statement is:

```
SELECT
ip_to_province (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS province,
count( 1 ) AS " attacks"
WHERE attack != 'default' and ip_to_country(CASE WHEN sip = '-' THEN remote_ip ELSE sip END)
= 'China'
GROUP BY
province
```

- **Attack Types.** The associated query and analysis statement is:

```
SELECT time_format( MILLIS_TO_TIMESTAMP( TIMESTAMP_TO_MILLIS(__time) -
MOD(TIMESTAMP_TO_MILLIS(__time), 3600)), 'HH:mm' ) AS dt, count( 1 ) AS cnt, CASE WHEN
action = 'block' THEN 'Intercepted' WHEN action = 'log' THEN 'Log only' WHEN action = 'captcha'
THEN 'Verification code' END AS attack FROM log WHERE action != " GROUP BY
TIMESTAMP_TO_MILLIS(__time) - MOD(TIMESTAMP_TO_MILLIS(__time), 3600), attack ORDER BY cnt
DESC
```

- **Web Attacks.** The associated query and analysis statement is:

```
SELECT
ip_to_province (
CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS province,
count( 1 ) AS " attacks"
WHERE action = 'block' and ip_to_country(CASE WHEN sip = '-' THEN remote_ip ELSE sip END) =
'China'
GROUP BY
province
```

- **CC Attacks(World).** The associated query and analysis statement is:

```
SELECT
ip_to_country (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS country,
count( 1 ) AS " attacks"
WHERE attack != 'default'
GROUP BY
country
```

- **Web Attacks(World).** The associated query and analysis statement is:

```
SELECT
ip_to_country (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS country,
count( 1 ) AS " attacks"
WHERE action = 'block'
```



```
GROUP BY
country
```

----End

Viewing WAF Access Log Center

Step 1 Log in to the LTS console. In the navigation pane, choose **Dashboards**.

Step 2 Choose **WAF dashboard templates** under **Dashboard Templates** and click **WAF access log center** to view the chart details.

- **PV.** The associated query and analysis statement is:

```
SELECT CASE WHEN diff [ 1 ] < 1000 THEN concat( cast( diff [ 1 ] AS
VARCHAR ), '' ) WHEN diff [ 1 ] < 1000 * 1000 THEN concat( cast( round( diff [ 1 ] / 1000,
1 ) AS VARCHAR ), ' thousand' ) WHEN diff [ 1 ] < 1000000000 THEN
concat( cast( round( diff [ 1 ] / 1000000.0, 1 ) AS VARCHAR ), ' million' ) WHEN diff [ 1 ] / 1000.0
< 1000000000 THEN concat( cast( round( diff [ 1 ] / 1000 / 1000000.0, 1 ) AS VARCHAR ), '
billion' ) ELSE concat( cast( round( diff [ 1 ] / 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), '
trillion' ) END AS "VALUE" , CASE WHEN diff [ 2 ] = 0 THEN 0 ELSE round( diff [ 3 ] -
1, 2 ) END AS ratio FROM ( SELECT compare ( DATA, 86400 ) AS diff FROM
( SELECT count( 1 ) AS DATA FROM log ) )
```

- **UV.** The associated query and analysis statement is:

```
SELECT CASE WHEN diff [ 1 ] < 1000 THEN concat( cast( diff [ 1 ] AS
VARCHAR ), '' ) WHEN diff [ 1 ] < 1000 * 1000 THEN concat( cast( round( diff [ 1 ] / 1000,
1 ) AS VARCHAR ), ' thousand' ) WHEN diff [ 1 ] < 1000000000 THEN
concat( cast( round( diff [ 1 ] / 1000000.0, 1 ) AS VARCHAR ), ' million' ) WHEN diff [ 1 ] / 1000.0
< 1000000000 THEN concat( cast( round( diff [ 1 ] / 1000 / 1000000.0, 1 ) AS VARCHAR ), '
billion' ) ELSE concat( cast( round( diff [ 1 ] / 1000.0 / 1000 / 1000 / 1000, 1 ) AS VARCHAR ), '
trillion' ) END AS "VALUE" , CASE WHEN diff [ 2 ] = 0 THEN 0 ELSE
round( diff [ 3 ] - 1, 2 ) END AS ratio FROM ( SELECT compare ( DATA, 86400 )
AS diff FROM ( SELECT count( DISTINCT CASE WHEN sip = '-' THEN remote_ip ELSE sip
END ) AS "DATA" FROM log ) )
```

- **Incoming Traffic.** The associated query and analysis statement is:

```
SELECT CASE WHEN diff [ 1 ] < 102 THEN concat( cast( diff [ 1 ] AS
VARCHAR ), ' B' ) WHEN diff [ 1 ] < 1024 * 1024 THEN concat( cast( round( diff [ 1 ] / 1024,
1 ) AS VARCHAR ), ' KB' ) WHEN diff [ 1 ] < 1024 * 1024 * 1024 THEN
concat( cast( round( diff [ 1 ] / 1024.0 / 1024, 1 ) AS VARCHAR ), ' MB' ) WHEN diff [ 1 ] / 1024.0
< 1024 * 1024 * 1024 THEN concat( cast( round( diff [ 1 ] / 1024.0 / 1024 / 1024, 1 ) AS
VARCHAR ), ' GB' ) ELSE concat( cast( round( diff [ 1 ] / 1024.0 / 1024 / 1024 / 1024, 1 ) AS
VARCHAR ), ' TB' ) END AS "VALUE" , CASE WHEN diff [ 2 ] = 0 THEN
0 ELSE round( diff [ 3 ] - 1, 2 ) END AS ratio FROM ( SELECT compare ( "DATA",
86400 ) AS diff FROM ( SELECT COALESCE ( sum( request_length ), 0 ) AS "DATA" FROM
log ) )
```

- **Peak In-Bandwidth.** The associated query and analysis statement is:

```
SELECT CASE WHEN diff [ 1 ] < 102 THEN concat( cast( round( diff [ 1 ], 2 ) AS
VARCHAR ), ' B/s' ) WHEN diff [ 1 ] < 1024 * 1024 THEN concat( cast( round( diff [ 1 ] /
1024, 1 ) AS VARCHAR ), ' KB/s' ) WHEN diff [ 1 ] < 1024 * 1024 * 1024 THEN
concat( cast( round( diff [ 1 ] / 1024.0 / 1024, 1 ) AS VARCHAR ), ' MB/s' ) WHEN diff [ 1 ] /
1024.0 < 1024 * 1024 * 1024 THEN concat( cast( round( diff [ 1 ] / 1024.0 / 1024 / 1024, 1 ) AS
VARCHAR ), ' GB/s' ) ELSE concat( cast( round( diff [ 1 ] / 1024.0 / 1024 / 1024 / 1024, 1 ) AS
VARCHAR ), ' TB/s' ) END AS "VALUE" , CASE WHEN diff [ 2 ] = 0
THEN 0 ELSE round( diff [ 3 ] - 1, 2 ) END AS ratio FROM ( SELECT compare
( "DATA", 86400 ) AS diff FROM ( SELECT COALESCE ( max( "DATA" ), 0 ) AS
"DATA" FROM ( SELECT TIME_FLOOR( __time, 'PT1M' ) AS dt, sum( request_length ) / 60.0 AS
"DATA" FROM log GROUP BY dt LIMIT 1000 ) ) )
```

- **Peak Out-Bandwidth.** The associated query and analysis statement is:

```
SELECT CASE WHEN diff [ 1 ] < 102 THEN concat( cast( round( diff [ 1 ], 2 ) AS
VARCHAR ), ' B/s' ) WHEN diff [ 1 ] < 1024 * 1024 THEN concat( cast( round( diff [ 1 ] / 1024, 1 ) AS
VARCHAR ), ' KB/s' ) WHEN diff [ 1 ] < 1024 * 1024 * 1024 THEN concat( cast( round( diff [ 1 ] / 1024.0 / 1024, 1 ) AS
VARCHAR ), ' MB/s' ) WHEN diff [ 1 ] / 1024.0 < 1024 * 1024 * 1024 THEN concat( cast( round( diff
[ 1 ] / 1024.0 / 1024 / 1024, 1 ) AS VARCHAR ), ' GB/s' ) ELSE concat( cast( round( diff [ 1 ] / 1024.0 /
1024 / 1024 / 1024, 1 ) AS VARCHAR ), ' TB/s' ) END AS "value", case when diff [ 2 ] = 0 then 0 else
round( diff [ 3 ] - 1, 2 ) END AS "ratio" FROM ( SELECT compare ( "DATA", 86400 ) AS diff FROM
( SELECT COALESCE ( max( bytes_out ), 0 ) AS "DATA" FROM ( SELECT time_ceil( __time, 'PT1M' ) AS
dt, sum( body_bytes_sent ) / 60.0 AS bytes_out FROM log GROUP BY dt LIMIT 1000 ) ) )
```

- **Traffic Bandwidth Trend.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT( MILLIS_TO_TIMESTAMP( TIMESTAMP_TO_MILLIS(__time) -
MOD(TIMESTAMP_TO_MILLIS(__time) , 600000)), 'HH:mm' ) AS dt,
round( sum( request_length )/ 1024.0 / 600, 2 ) AS "Incoming (KB/s)",
round( sum( body_bytes_sent )/ 1024.0 / 600, 2 ) AS "Outgoing (KB/s)" where request_length is
not null GROUP BY TIMESTAMP_TO_MILLIS(__time) - MOD(TIMESTAMP_TO_MILLIS(__time) ,
600000) ORDER BY dt LIMIT 1000
```
- **PV/UV Trend.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT(MILLIS_TO_TIMESTAMP( TIMESTAMP_TO_MILLIS(__time) -
MOD(TIMESTAMP_TO_MILLIS(__time) , 3600000)), 'HH:mm' ) AS dt, count( 1 ) AS PV,
APPROX_COUNT_DISTINCT (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS UV
FROM log GROUP BY TIMESTAMP_TO_MILLIS(__time) -
MOD(TIMESTAMP_TO_MILLIS(__time) , 3600000) ORDER BY dt LIMIT 1000
```
- **Visit Statuses.** The associated query and analysis statement is:

```
SELECT TIME_FORMAT(MILLIS_TO_TIMESTAMP( TIMESTAMP_TO_MILLIS(__time) -
MOD(TIMESTAMP_TO_MILLIS(__time) , 3600000)), 'HH:mm' ) AS dt, count( 1 ) AS cnt,
concat( cast( "response_code" / 100 AS VARCHAR ), 'XX' ) AS "status" GROUP BY
TIMESTAMP_TO_MILLIS(__time) - MOD(TIMESTAMP_TO_MILLIS(__time) , 3600000) ,
"response_code" / 100 ORDER BY dt DESC LIMIT 10000
```
- **Source.** The associated query and analysis statement is:

```
SELECT ip_to_province (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS country,
count( 1 ) AS "Visits" where ip_to_country(CASE WHEN sip = '-' THEN remote_ip ELSE sip END) =
'China' GROUP BY country
```
- **Incoming Traffic Source (China).** The associated query and analysis statement is:

```
SELECT ip_to_province (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS region,
round( sum( request_length )/ 1024.0 / 1024, 4 ) AS "Incoming (MB)" where ip_to_country(CASE
WHEN sip = '-' THEN remote_ip ELSE sip END) = 'China' GROUP BY region
```
- **Incoming Traffic Source (World).** The associated query and analysis statement is:

```
SELECT ip_to_country (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS region,
round( sum( request_length )/ 1024.0 / 1024, 4 ) AS "Incoming (MB)" where request_length is not
null GROUP BY region
```
- **Source Network Providers.** The associated query and analysis statement is:

```
SELECT ip_to_provider (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) AS provider,
round( sum( request_length )/ 1024.0 / 1024.0, 3 ) AS mb_in GROUP BY provider
HAVING ip_to_provider (CASE WHEN sip = '-' THEN remote_ip ELSE sip END) != '*' ORDER
BY mb_in DESC LIMIT 10
```
- **Domain Names.** The associated query and analysis statement is:

```
SELECT http_host, count( 1 ) AS "Visits" GROUP BY http_host ORDER BY
"Visits" DESC LIMIT 30
```
- **URLs with the Slowest Response.** The associated query and analysis statement is:

```
SELECT http_host AS "Website",url_extract_path (COALESCE ( url, '/' )) AS URL,sum( request_time )/
count( 1 ) AS "Response Duration (ms)",count( 1 ) AS "Visits" GROUP BY http_host, url ORDER BY
"Response Duration (ms)" DESC LIMIT 100
```
- **Clients with the Most Visits.** The associated query and analysis statement is:

```
SELECT ip AS "Client", client AS "Network", concat( cast( (CASE WHEN pv IS NULL THEN 0 ELSE pv
END) AS VARCHAR ), '(', cast( case when head_pv = 'null' then 0 else (case when head_pv > 0
then head_pv else 0 end) end AS VARCHAR ), '/', cast( case when get_pv = 'null' then 0 else (case
when get_pv > 0 then get_pv else 0 end) end AS VARCHAR ), '/', cast( case when put_pv = 'null'
then 0 else (case when put_pv > 0 then put_pv else 0 end) end AS VARCHAR ), '/', cast( case when
post_pv = 'null' then 0 else (case when post_pv > 0 then post_pv else 0 end) end AS VARCHAR ), '/',
cast( case when delete_pv = 'null' then 0 else (case when delete_pv > 0 then delete_pv else 0 end)
end AS VARCHAR ), '/', ')' ) AS "PV(Head, Get, Put, Post, Delete)", error_count AS "Wrong Visits"
FROM ( SELECT ip, client, sum( CASE WHEN "method" = 'PUT' AND "status" < 400 THEN pv ELSE
0 END ) AS put_pv, sum( CASE WHEN "method" = 'GET' AND "status" < 400 THEN pv ELSE 0 END )
AS get_pv, sum( CASE WHEN "method" = 'POST' AND "status" < 400 THEN pv ELSE 0 END ) AS
post_pv, sum( CASE WHEN "method" = 'DELETE' AND "status" < 400 THEN pv ELSE 0 END ) AS
delete_pv, sum( CASE WHEN "method" = 'HEAD' AND "status" < 400 THEN pv ELSE 0 END ) AS
head_pv, sum( throughput ) AS throughput, sum( pv ) AS pv, sum( CASE WHEN "status" < 400
THEN 1 ELSE 0 END ) AS error_count FROM ( SELECT CASE WHEN sip = '-' THEN remote_ip
```

```
ELSE sip END AS ip, "method", CASE WHEN ip_to_country ( CASE WHEN sip = '-' THEN
remote_ip ELSE sip END )= 'Shanghai' THEN 'Shanghai, China' WHEN ip_to_province ( CASE WHEN
sip = '-' THEN remote_ip ELSE sip END )= '*' THEN 'Unknown IP' WHEN ip_to_provider ( CASE
WHEN sip = '-' THEN remote_ip ELSE sip END )= 'Private IP' THEN 'Private IP' ELSE
concat( ip_to_country ( CASE WHEN sip = '-' THEN remote_ip ELSE sip END ), '/', ip_to_province
( CASE WHEN sip = '-' THEN remote_ip ELSE sip END ), '/', CASE WHEN ip_to_city ( CASE WHEN
sip = '-' THEN remote_ip ELSE sip END )= '*' THEN '' ELSE ip_to_city ( CASE WHEN sip = '-' THEN
remote_ip ELSE sip END ) END, ' ', ip_to_provider ( CASE WHEN sip = '-' THEN remote_ip ELSE sip
END )) END AS client, sum( CASE WHEN "response_code" < 400 THEN 1 ELSE 0 END ) AS pv,
round( sum( request_length )/ 1024.0 / 1024, 1 ) AS throughput, "response_code" AS "status"
FROM log GROUP BY ip, client, "method", "response_code" ORDER BY pv DESC, client,
"method" LIMIT 1000 ) GROUP BY ip, client ORDER BY pv DESC ) LIMIT 100
```

----End

6 Log Alarms

6.1 Configuring Log Alarm Rules

You can set alarm rules based on [key words](#) and [SQL statistics](#) for logs in log streams to monitor service status in real time. Currently, up to 200 keyword alarms can be created for each account.

Prerequisites

A log group and stream have been created. For details, see [Managing Log Groups](#) and [Managing Log Streams](#).


Creating a Keyword Alarm Rule

LTS allows you to collect statistics on log keywords in log streams and set alarm rules to monitor them. By checking the number of keyword occurrences in a specified period, you can have a real-time view of the service running.

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Alarms** in the navigation pane.
- Step 3** Click the **Alarm Rules** tab.
- Step 4** Click **Create**. The **Create Alarm Rule** right panel is displayed.
- Step 5** Configure alarm rule parameters.

Table 6-1 Keyword alarm rule parameters

Category	Parameter	Description
Basic Info	Rule Name	Name of the alarm rule. Enter 1 to 64 characters and do not start or end with a hyphen (-) or underscore (_). Only letters, digits, hyphens, and underscores are allowed. NOTE After an alarm rule is created, the rule name can be modified. After the modification, move the cursor over the rule name to view the new and original rule names. The original rule name cannot be changed.
	Description	Brief description of the rule. Enter up to 64 characters.
Statistical Analysis	Statistics	By keyword: applicable when keywords are used to search for and configure log alarms.
	Query Condition	Log Group Name: Select a log group.
		Log Stream Name: Select a log stream. NOTE If a log group contains more than one log stream, you can select multiple log streams when creating a keyword alarm rule.
		Query Time Range: Specify the query period of the statement. It is one period earlier than the current time. For example, if Query Time Range is set to one hour and the current time is 9:00, the period of the query statement is 8:00–9:00. <ul style="list-style-type: none"> The value ranges from 1 to 60 in the unit of minutes. The value ranges from 1 to 24 in the unit of hours.
	Keywords: Enter keywords that you want LTS to monitor in logs. Exact and fuzzy matches are supported. A keyword is case-sensitive and contains up to 1,024 characters.	

Category	Parameter	Description
	Check Rule	<p>Configure a condition that will trigger the alarm.</p> <p>Matching Log Events: When the number of log events that contain the configured keywords reaches the specified value, an alarm is triggered. Four comparison operators are supported: greater than (>), greater than or equal to (>=), less than (<), and less than or equal to (<=).</p> <ul style="list-style-type: none"> • Click + to add a conditional expression with an OR relationship. A maximum of 20 conditional expressions can be added. • Click  to delete a conditional expression. <p>The number of queries refers to the Query Frequency set in Advanced Settings and the number of times the condition must be met to trigger the alarm. The number of queries must be greater than or equal to the number of times the condition must be met.</p> <p>NOTE</p> <ul style="list-style-type: none"> • The alarm severity can be critical (default), major, minor, or info. • Number of queries: 1–10

Category	Parameter	Description
Advanced Settings	Query Frequency	<p>The options for this parameter are:</p> <ul style="list-style-type: none"> • Hourly: The query is performed at the top of each hour. • Daily: The query is run at a specific time every day. • Weekly: The query is run at a specific time on a specific day every week. • Custom interval: You can specify the interval from 1 minute to 60 minutes or from 1 hour to 24 hours. For example, if the current time is 9:00 and the Custom interval is set to 5 minutes, the first query is at 9:00, the second query is at 9:05, the third query is at 9:10, and so on. <p>NOTE When the query time range is set to a value larger than 1 hour, the query frequency must be set to every 5 minutes or a lower frequency.</p> <ul style="list-style-type: none"> • CRON: CRON expressions support schedules down to the minute and use 24-hour format. Examples: <ul style="list-style-type: none"> - 0/10 * * * *: The query starts from 00:00 and is performed every 10 minutes. That is, queries start at 00:00, 00:10, 00:20, 00:30, 00:40, 00:50, 01:00, and so on. For example, if the current time is 16:37, the next query is at 16:50. - 0 0/5 * * * *: The query starts from 00:00 and is performed every 5 hours at 00:00, 05:00, 10:00, 15:00, 20:00, and so on. For example, if the current time is 16:37, the next query is at 20:00. - 0 14 * * * *: The query is performed at 14:00 every day. - 0 0 10 * * * *: The query is performed at 00:00 on the 10th day of every month.
Advanced Settings	Restores	<p>Configure a policy for sending an alarm clearance notification.</p> <p>If alarm clearance notification is enabled and the trigger condition has not been met for the specified number of statistical periods, an alarm clearance notification is sent.</p> <p>Number of last queries: 1–10</p>

Category	Parameter	Description
Advanced Settings	Notify When	<ul style="list-style-type: none"> • Alarm triggered: Specify whether to send a notification when an alarm is triggered. If this option is enabled, a notification will be sent when the trigger condition is met. • Alarm cleared: Specify whether to send a notification when an alarm is cleared. If this option is enabled, a notification will be sent when the recovery policy is met.
Advanced Settings	Frequency	<p>You can select Once, Every 5 minutes, Every 10 minutes, Every 15 minutes, Every 30 minutes, Every hour, Every 3 hours, or Every 6 hours to send alarms.</p> <p>Once indicates that a notification is sent once an alarm is generated. Every 10 minutes indicates that the minimum interval between two notifications is 10 minutes, preventing alarm storms.</p>
Advanced Settings	Alarm Action Rules	<p>Select a desired rule from the drop-down list.</p> <p>If no rule is available, click Create Alarm Action Rule on the right.</p>
Advanced Settings	Language	<p>Specify the language (Chinese (simplified) or English) in which alarms are sent.</p>

Step 6 Click **OK**.

 **NOTE**

After an alarm rule is created, its status is **Enabled** by default. After the alarm rule is disabled, the alarm status is **Disabled**. After the alarm rule is disabled temporarily, the alarm status is **Temporarily closed to May 30, 2023 16:21:24.000 GMT+08:00**. (The time is for reference only.)

When the alarm rule is enabled, an alarm will be triggered if the alarm rule is met. When it is disabled, an alarm will not be triggered even if the alarm rule is met.

----End

Creating a SQL Alarm Rule



LTS can regularly run the SQL queries that you specify on structured logs and trigger an alarm when the alarm rule is met. You can view SQL alarms on the LTS console. Each SQL alarm rule can be associated with one to three charts. Each chart contains a SQL statement for querying a log stream.

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

- Step 1** Log in to the LTS console and choose **Log Alarms** in the navigation pane.
- Step 2** Click the **Alarm Rules** tab.
- Step 3** Click **Create**. The **Create Alarm Rule** right panel is displayed.
- Step 4** Configure alarm rule parameters by referring to [Table 6-2](#).

Table 6-2 SQL alarm rule parameters

Category	Parameter	Description
Basic Info	Rule Name	Name of the alarm rule. Enter 1 to 64 characters and do not start or end with a hyphen (-) or underscore (_). Only letters, digits, hyphens, and underscores are allowed. NOTE After an alarm rule is created, the rule name can be modified. After the modification, move the cursor over the rule name to view the new and original rule names. The original rule name cannot be changed.
	Description	Rule description. Enter up to 64 characters.
Statistical Analysis	Statistics	By SQL: Use SQL analysis to configure an alarm rule.

Category	Parameter	Description
	Charts	<p>You can add a chart in two ways.</p> <ul style="list-style-type: none"> Configure from Scratch: Click Configure from Scratch and then select a log group and stream. Set parameters as follows: <ul style="list-style-type: none"> Log Group Name: (Required) Select a log group. Log Stream Name: (Required) Select a log stream. <p>NOTE If the logs in the log stream have not been structured, configure log structuring first.</p> <p>Query Time Range: (Optional) the period specified for querying logs. It can be 1 to 60 minutes or 1 to 24 hours.</p> <p>Query Statement: required.</p> Import Configuration: Click + Import Configuration . On the displayed Custom page, select a log group and stream, select a chart, and click OK. If there are no charts available or the charts do not fit your needs, click Create Chart. Configure the chart parameters, click OK, and click Save and Back in the upper right corner to return to the Create Alarm Rule right panel. You can see that the chart you just created has been selected, and the query statement has been filled in. Specify the query time range (1 to 60 minutes or 1 to 24 hours). When the query frequency is set to every 1 to 4 minutes, the query time range can only be set to a value no larger than 1 hour. You can continue to add more charts by clicking + Import Configuration . <p>NOTE</p> <ul style="list-style-type: none"> Click  to go to the visualization page of the log stream. Click  to delete an added chart. Click Preview to view the data after visualized analysis. You must click Preview; otherwise, the alarm rule cannot be saved. Up to three charts can be added. The chart and the query statement are required.

Category	Parameter	Description
	Check Rule	<p>Enter a specific conditional expression. When the expression execution result is true, an alarm is generated.</p> <p>NOTE</p> <ul style="list-style-type: none"> • Conditional expressions support Chinese characters. • Conditional expressions cannot contain only digits or start with a digit. • Specify the number of queries and the number of times the condition must be met to trigger the alarm. The number of queries must be greater than or equal to the number of times the condition must be met. • The alarm severity can be critical (default), major, minor, or info. • Number of queries: 1–10 <p>Basic syntax and syntax across multiple charts are supported.</p> <ul style="list-style-type: none"> • Basic syntax <ul style="list-style-type: none"> – Basic arithmetic operators: addition (+), subtraction (-), multiplication (*), division (/), and modulo (%). Example: x * 10 + y > 100 – Comparison operators: greater than (>), greater than or equal to (>=), less than (<), less than or equal to (<=), equal to (==), and not equal to (!=). Example: x >= 100. – Logical operators: && (and) and (or). Example: x > 0 && y < 200 – Logical negation (!). Example: !(x < 1 && x > 100) – Numeric constants: They are processed as 64-bit floating point numbers. Example: x > 10 – String constants. Example: str == "string" – Boolean constants: true and false. Example: (x < 100)!=true – Parentheses: used to change the order of operations. Example: x *(y + 10) < 200 – contains function: used to check whether a string contains a substring. For example, if you run contains(str, "hello") and true is returned, the string contains the hello substring. • Syntax across multiple charts <ul style="list-style-type: none"> – Basic arithmetic operators: addition (+), subtraction (-), multiplication (*), division (/), and modulo (%). – Comparison operators: greater than (>), greater than or equal to (>=), less than (<), less than or

Category	Parameter	Description
		<p>equal to (<=), equal to (==), and not equal to (!=).</p> <ul style="list-style-type: none"> - Logical operators: && (and) and (or). - Logical negation (!) - contains function - Parentheses
Advanced Settings	Query Frequency	<p>The options for this parameter are:</p> <ul style="list-style-type: none"> ● Hourly: The query is performed at the top of each hour. ● Daily: The query is run at a specific time every day. ● Weekly: The query is run at a specific time on a specific day every week. ● Custom interval: You can specify the interval from 1 minute to 60 minutes or from 1 hour to 24 hours. For example, if the current time is 9:00 and the Custom interval is set to 5 minutes, the first query is at 9:00, the second query is at 9:05, the third query is at 9:10, and so on. <p>NOTE When the query time range is set to a value larger than 1 hour, the query frequency must be set to every 5 minutes or a lower frequency.</p> <ul style="list-style-type: none"> ● CRON: CRON expressions support schedules down to the minute and use 24-hour format. Examples: <ul style="list-style-type: none"> - 0/10 * * * *: The query starts from 00:00 and is performed every 10 minutes. That is, queries start at 00:00, 00:10, 00:20, 00:30, 00:40, 00:50, 01:00, and so on. For example, if the current time is 16:37, the next query is at 16:50. - 0 0/5 * * * *: The query starts from 00:00 and is performed every 5 hours at 00:00, 05:00, 10:00, 15:00, 20:00, and so on. For example, if the current time is 16:37, the next query is at 20:00. - 0 14 * * * *: The query is performed at 14:00 every day. - 0 0 10 * * * *: The query is performed at 00:00 on the 10th day of every month.
Advanced Settings	Restores	<p>Configure a policy for sending an alarm clearance notification.</p> <p>If alarm clearance notification is enabled and the trigger condition has not been met for the specified number of statistical periods, an alarm clearance notification is sent.</p> <p>Number of last queries: 1–10</p>

Category	Parameter	Description
Advanced Settings	Notify When	<ul style="list-style-type: none"> • Alarm triggered: Specify whether to send a notification when an alarm is triggered. If this option is enabled, a notification will be sent when the trigger condition is met. • Alarm cleared: Specify whether to send a notification when an alarm is cleared. If this option is enabled, a notification will be sent when the recovery policy is met.
Advanced Settings	Frequency	<p>You can select Once, Every 5 minutes, Every 10 minutes, Every 15 minutes, Every 30 minutes, Every hour, Every 3 hours, or Every 6 hours to send alarms.</p> <p>Once indicates that a notification is sent once an alarm is generated. Every 10 minutes indicates that the minimum interval between two notifications is 10 minutes, preventing alarm storms.</p>
Advanced Settings	Alarm Action Rules	<p>Select a desired rule from the drop-down list.</p> <p>If no rule is available, click Create Alarm Action Rule on the right. For details, see Creating an Alarm Action Rule.</p>
Advanced Settings	Language	<p>Specify the language (Chinese (simplified) or English) in which alarms are sent.</p>

Step 5 Click **OK**.

----End

Follow-up Operations on Alarm Rules

After creating an alarm rule, you can modify, enable, disable, copy, or delete it. Exercise caution when performing these operations.

- You can perform the following operations on a single alarm rule.
 - Modifying an alarm rule: Click **Modify** in the **Operation** column of the target alarm rule. On the displayed page, modify the rule name, query condition, and check rule, and click **OK**.
 - Enabling an alarm rule: Click **More > Enable** in the **Operation** column of the target alarm rule.
 - Disabling an alarm rule: Click **More > Disable** in the **Operation** column of the target alarm rule.
 - Temporarily disabling an alarm rule: Click **More > Disable Temporarily** in the **Operation** column of the target alarm rule.
 - Copying an alarm rule: Click **More > Copy** in the **Operation** column of the target alarm rule.

Deleting an alarm rule: Click **Delete** in the **Operation** column of the target alarm rule. In the displayed dialog box, click **OK**.

 **NOTE**

Deleted alarm rules cannot be recovered. Exercise caution when performing this operation.

- After selecting multiple alarm rules, you can perform the following operations on them: **Enable**, **Disable**, **Disable Temporarily**, **Re-Enable**, **Enable Clearance**, **Disable Clearance**, **Delete**, and **Export**.
- You can move the cursor to the rule name to view both the new and original names after modification. The original rule name cannot be changed.

6.2 Configuring Log Alarm Notifications

6.2.1 Creating a Message Template on the LTS Console

A message template defines the format of alarm notification messages sent to subscribers. LTS provides built-in message templates. Subscribers can select templates based on protocols. If the template of a specified protocol does not exist, the built-in template of that protocol is used to send messages to subscribers. When using a message template to send alarm notification messages, the system automatically replaces the template variables with the content in the alarm rule.

Creating a Message Template

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**.

Step 2 Choose **Log Alarms** in the navigation pane and click the **Alarm Action Rules** tab.

 **NOTE**

By default, LTS provides the following built-in message templates. If no message content is configured in your selected template, LTS uses a built-in template instead.

- **keywords_template** (English): keyword alarm template

Step 3 Click **Create** on the **Message Templates** tab page. On the displayed page, set the required parameters.

 **NOTE**

- The email content supports HTML tags and message preview.
- You can create up to 100 message templates for AOM and LTS. If there are already 100 templates, delete unnecessary templates before creating a new one.

Table 6-3 Message template parameters

Parameter	Description	Verification Rule	Example
Template Name	Message template name.	Include digits, letters, underscores (_), and hyphens (-). Do not start or end with an underscore or hyphen. (Max. 100 characters)	LTS-test
Description	Description of the template.	Include digits, letters, and underscores (_). Do not start or end with an underscore. (Max. 1,024 characters)	-
Message Header	Default message header to be added in messages.	<ul style="list-style-type: none">• English	<ul style="list-style-type: none">• "Dear user,"
Notification method	Notification method.	<ul style="list-style-type: none">• Email• SMS• HTTP/HTTPS	-
Topic	Message topic.	Customize the topic name or use variables. (Max. 512 characters) Only email templates need a topic name.	test

Parameter	Description	Verification Rule	Example
Body	Message content.	<p>Add Variable</p> <ul style="list-style-type: none"> Original rule name: <i>`\${event_name}`</i> Alarm severity: <i>`\${event_severity}`</i> Occurrence time: <i>`\${starts_at}`</i> Occurrence region: <i>`\${region_name}`</i> Account: <i>`\${domain_name}`</i> Alarm source: <i>`\${event.metadata.resource_provider}`</i> Resource type: <i>`\${event.metadata.resource_type}`</i> Resource ID: <i>`\${resources}`</i> Alarm status: <i>`\${event.annotations.alarm_status}`</i> Expression: <i>`\${event.annotations.condition_expression}`</i> Current value: <i>`\${event.annotations.current_value}`</i> Statistical period: <i>`\${event.annotations.frequency}`</i> Rule name: <i>`\${event.annotations.alarm_rule_alias}`</i> Frequency: <i>`\${event.annotations.notification_frequency}`</i> Original log group name: <i>`\${event.annotations.results[0].log_group_name}`</i> Original log stream name: <i>`\${event.annotations.results[0].log_stream_name}`</i> Variables supported by keyword alarms: 	<p><i>`\${event_name}`</i> <i>`\${event_severity}`</i> <i>`\${starts_at}`</i> <i>`\${region_name}`</i></p>

Parameter	Description	Verification Rule	Example
		<ol style="list-style-type: none"> 1. Query time: <i>\$event.annotations.results[0].time</i> 2. Query logs: (The maximum log size is 2 KB. If a log exceeds 2 KB, the extra part will be truncated and discarded.) <i>\$event.annotations.results[0].raw_results</i> 3. Query URL: <i>\$event.annotations.results[0].url</i> 4. Log group/stream name: <i>\$event.annotations.results[0].resource_id</i> <p>NOTE Only the original name of the log group or log stream created for the first time can be added. The modified log group or log stream name cannot be added.</p> <ol style="list-style-type: none"> 5. Enterprise project ID of the log stream: <i>\$event.annotations.results[0].eps_id</i> 6. Query custom field: <i>\$event.annotations.results[0].fields.xxx</i> <p>NOTE <i>xxx</i> indicates the structured fields and system fields (such as hostIP and hostName) of the raw logs. The maximum size of a log field is 1 KB. If a field exceeds 1 KB, the extra part will be truncated and discarded.</p> <p>Copy from Existing</p> <ul style="list-style-type: none"> ● keywords_template ● Custom templates (created with variables) 	

Step 4 When the configuration is complete, click **OK**.

----End

Modifying a Message Template

Step 1 In the message template list, click **Modify** in the row that contains the target template, and modify the template according to [Table 6-3](#). The template name cannot be modified.

 **NOTE**

Built-in message templates cannot be modified.

Step 2 Click **OK**.

----End

Copying a Message Template

Step 1 In the message template list, click **Copy** in the row that contains the target template. Set a new template name.

Step 2 Click **OK**.

----End

Deleting a Message Template

Step 1 In the message template list, click **Delete** in the **Operation** column of the target template.

 **NOTE**

Built-in message templates cannot be deleted.

Step 2 Click **OK**.

----End

Deleting Message Templates in a Batch

Step 1 In the message template list, select the templates to be deleted and click **Delete**.

Step 2 Click **OK**.

----End

Exporting Message Templates

Step 1 In the message template list, select the templates to be exported and click **Export**.

Step 2 Click **Export all data to an XLSX file** or **Export selected data to an XLSX file**. After the data is exported, you can view it on the local PC.

----End

6.2.2 Creating an Alarm Action Rule

LTS allows you to customize alarm action rules. You can create an alarm action rule to associate an SMN topic with a message template. You can also customize notification content when creating a message template.

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Prerequisites

- A topic has been created.
- A topic policy has been configured.
- A subscriber has been added to the topic. A subscriber is the recipient of the notification, for example, an email or SMS message receiver.

Precaution

You can create a maximum of 1,000 alarm action rules. If this number has been reached, delete unnecessary rules.

Creating an Alarm Action Rule

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Alarms** in the navigation pane.
- Step 3** Click the **Alarm Action Rules** tab.
- Step 4** Click **Create**. Set parameters such as the action rule name and action rule configuration.

Table 6-4 Parameters for configuring an alarm action rule

Parameter	Description
Action Rule	Enter 1 to 64 characters. Only digits, letters, underscores (_), and hyphens (-) are allowed. Do not start or end with an underscore or hyphen.
Enterprise Project	Select an enterprise project. This parameter is displayed only when the enterprise project function is enabled for the current account.
Description	Enter a description for the rule. Up to 1,024 characters are allowed.
Topic	SMN topic. Select your desired topic from the drop-down list. If there is no topic you want to select, create one on the SMN console.

Parameter	Description
Message Template	Notification message template. Select your desired template from the drop-down list. If no proper message template is available, click Create Template to create one.


Step 5 Click **OK**.

----End

More Operations

After an alarm action rule is created, you can perform operations described in [Table 6-5](#).

Table 6-5 Related operations

Operation	Description
Modifying an alarm action rule	Click Modify in the Operation column.
Exporting an alarm action rule	Select one or more alarm action rules and click Export . If no alarm action rule is selected, clicking Export will export all alarm action rules.
Deleting an alarm action rule	<ul style="list-style-type: none"> To delete a single rule, click Delete in the Operation column in the row that contains the rule, and then click Yes on the displayed page. To delete one or more rules, select them, click Delete above the rule list, and then click Yes on the displayed page. <p>NOTE Before deleting an alarm action rule, you need to delete the alarm rule bound to the action rule.</p>
Searching for an alarm action rule	Enter a rule name in the search box in the upper right corner and click  .

6.3 Viewing Alarms in LTS

LTS allows you to configure keyword alarm rules to periodically query log data. When an alarm rule is met, an alarm will be reported. You can view the alarms on the LTS console.


Prerequisites

You have created an alarm rule.


Procedure

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Alarms** in the navigation pane.
- Step 3** Click the **Alarms** tab. The alarms generated in 30 minutes from now and their trend charts are displayed by default.
- Step 4** Set criteria to search for your target alarms.
- In the search box above the alarm list, select a log group, log stream, alarm severity, and rule name.
 - Set a time range. By default, 30 minutes is specified (relative time from now). There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.

NOTE

- From now: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31.
 - From last: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and **1 hour** is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
 - **Specified**: queries log data that is generated in a specified time range.
- Step 5** Click  after you set the search criteria. The details and trend of the alarms that match the criteria will be displayed.
- Step 6** You can point to the **Details** column of an alarm on the **Active Alarms** tab to view the complete alarm details. Alternatively, click the name in the **Alarm Name** column of an alarm. Details about the alarm are displayed in the right panel that pops up.

After the reported fault is rectified, you can click the deletion button in the row that contains the corresponding alarm on the **Active Alarms** tab to clear the alarm. The cleared alarm will then be displayed on the **Historical Alarms** tab.

If you have configured search criteria to filter alarms, you need to manually refresh the alarm list. To enable automatic refresh, click  in the upper right corner and select **Refresh Every 30s**, **Refresh Every 1m**, or **Refresh Every 5m** from the drop-down list box. You can still manually refresh the alarm list when automatic refresh is enabled by selecting **Refresh Now** from the drop-down list box.

----End

7 Log Transfer

7.1 Overview

After being reported from hosts and cloud services to LTS, logs will be retained in LTS for the retention period you specify. Once this period ends, LTS will delete them. You can specify the retention period when creating a log group or stream. For details, see [Managing Log Groups](#) and [Managing Log Streams](#). Retained logs are deleted once the period is over. For long-term storage or persistent logging, you can transfer logs to other cloud services.

 **NOTE**

Log transfer refers to when logs are replicated to other cloud services. LTS deletes retained logs once the retention period is over, but the logs that have been transferred to other services are not affected.

7.2 Transferring Logs to OBS

You can transfer logs to OBS and download log files from the OBS console. You can choose scheduled or one-time transfers.

- [Creating a Log Transfer Task](#)
- [Creating a One-off Log Transfer Task](#)

 **NOTE**

- To use this function, you must have the **OBS Administrator** permissions apart from the LTS permissions.
- LTS transfers only logs generated after the transfer task is configured, not historical logs.

Prerequisites

- Logs have been ingested to LTS. For details, see [Log Ingestion](#).
- You have created an OBS bucket.

Creating a Log Transfer Task

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Log Transfer** in the navigation pane.
- Step 3** Click **Configure Log Transfer** in the upper right corner.
- Step 4** On the displayed page, configure the log transfer parameters.

Table 7-1 Transfer parameters

Parameter	Description	Example
Transfer Mode	If you select the current account as the log source account, select Scheduled . Scheduled: Logs are periodically transferred to OBS for long-term storage.	Scheduled
Enable Transfer	Enabled by default.	Enabled
Transfer Destination	Select a cloud service for log transfer.	OBS
Log Group Name	Select a log group.	N/A
Enterprise Project Name	Select an enterprise project. <ul style="list-style-type: none"> • This parameter is displayed only when the enterprise project function is enabled for the current account. • If the enterprise project function is enabled for the current account: <ul style="list-style-type: none"> – All enterprise projects under the current account are displayed in the drop-down list when Log Source Account is set to Current. – default is displayed when Log Source Account is set to Other and the enterprise project function is not enabled for the delegator account. – All enterprise projects under the delegator account are displayed when Log Source Account is set to Other and the enterprise project function is enabled for the delegator account. 	-
Log Stream Name	Select a log stream. NOTE Log streams that have been configured with OBS transfer settings cannot be configured again.	-

Parameter	Description	Example
OBS Bucket	<ul style="list-style-type: none"> • Select an OBS bucket. <ul style="list-style-type: none"> – If no OBS buckets are available, click View OBS Bucket to access the OBS console and create an OBS bucket. • Currently, LTS supports only standard OBS buckets with the single-AZ storage policy. 	-
Custom Log Transfer Path	<ul style="list-style-type: none"> • Enabled: Logs will be transferred to a custom path to separate transferred log files of different log streams. The format is /LogTanks/Region name/Custom path. The default custom path is lts/%Y/%m/%d, where %Y indicates the year, %m indicates the month, and %d indicates the day. A custom path must meet the following requirements: <ul style="list-style-type: none"> – Must start with /LogTanks/Region name. – Can contain only letters, digits, and the following special characters: &\$@;,:=+?-._/ %. The character % can only be followed only by Y (year), m (month), d (day), H (hour), and M (minute). Any number of characters can be added before and after %Y, %m, %d, %H, and %M, and the sequence of these variables can be changed. – Can contain 1–128 characters. Example: <ol style="list-style-type: none"> 1. If you enter LTS-test/%Y/%m/%d/%H/%M, the path is LogTanks/Region name/LTS-test/Y/m/d/H/M/Log file name. 2. If you enter LTS-test/%d/%H/%m/%Y, the path is LogTanks/Region name/LTS-test/d/H/m/Y/Log file name. • Disabled: Logs will be transferred to the default path. The default path is LogTanks/Region name/2019/01/01/Log group/Log stream/Log file name. 	LTS-test/%Y/%m/%d/%H/%M

Parameter	Description	Example
Log Prefix	<p>The file name prefix of the log files transferred to an OBS bucket</p> <p>The prefix must meet the following requirements:</p> <ul style="list-style-type: none"> • Can contain 0 to 64 characters. • Can contain only letters, digits, hyphens (-), underscores (_), and periods (.). <p>Example: If you enter LTS-log, the log file name will be LTS-log_Log file name.</p>	LTS-log
Format	<p>The storage format of logs. The value can be Raw Log Format or JSON.</p> <ul style="list-style-type: none"> • Example of the raw log format: (Logs displayed on the LTS console are in the raw format.) <pre>Sep 30 07:30:01 ecs-bd70 CRON[3459]: (root) CMD (/opt/oss/servicemgr/ICAgent/bin/manual/mstart.sh > /dev/null 2>&1)</pre> • The following is an example of the JSON format: <pre>{"host_name":"ecs-bd70","ip":"192.168.0.54","line_no":249,"message":"Sep 30 14:40:01 ecs-bd70 CRON[4363]: (root) CMD (/opt/oss/servicemgr/ICAgent/bin/manual/mstart.sh > /dev/null 2>&1)\n","path":"/var/log/syslog","time":1569825602303}</pre> 	Json
Log Transfer Interval	<p>The interval for automatically transferring logs to OBS buckets. The value can be 2, 5, or 30 minutes, or 1, 3, 6, or 12 hours.</p>	3 hours
Time Zone	<p>When logs are transferred to OBS buckets, the time in the transfer directory and file name will use the specified UTC time zone.</p>	(UTC) Coordinated Universal Time

Parameter	Description	Example
Filter by Tag Fields	<p>During transfer, logs will be filtered by tag fields collected by ICAgent.</p> <ul style="list-style-type: none"> Disabled: Logs will not be filtered by tag fields. Enabled: Default tag fields include those for hosts (hostIP, hostId, hostName, pathFile, and collectTime) and for Kubernetes (clusterName, clusterId, nameSpace, podName, containerName, and appName). Optional public tag fields are regionName, logStreamName, logGroupName, and projectId. <p>NOTE When Filter by Tag Fields is enabled, Format must be JSON.</p> <ul style="list-style-type: none"> Filter by Tag Fields: When this parameter is enabled, logs will be filtered by tags. 	Enabled

Step 5 Click **OK**. When the log transfer status changes to **Normal**, the transfer task has been created.

Step 6 Click the OBS bucket name in the **Transfer Destination** column to switch to the OBS console and view the transferred log files.

Transferred logs can be downloaded from OBS to your local computer for viewing.

 **NOTE**

Logs stored in OBS are in raw or JSON format.

----End

Creating a One-off Log Transfer Task

Step 1 Click **Configure Log Transfer** in the upper right corner.

Step 2 On the displayed page, configure the log transfer parameters.

Table 7-2 Transfer parameters

Parameter	Description	Example
Transfer Mode	One-time: Logs are transferred to OBS for long-term storage in a one-time way.	One-time
Transfer Destination	Select a cloud service for log transfer.	OBS
Log Group Name	Select a log group.	N/A

Parameter	Description	Example
Enterprise Project Name	<p>Select an enterprise project.</p> <ul style="list-style-type: none"> • This parameter is displayed only when the enterprise project function is enabled for the current account. • If the enterprise project function is enabled for the current account: <ul style="list-style-type: none"> - All enterprise projects under the current account are displayed in the drop-down list when Log Source Account is set to Current. - default is displayed when Log Source Account is set to Other and the enterprise project function is not enabled for the delegator account. - All enterprise projects under the delegator account are displayed when Log Source Account is set to Other and the enterprise project function is enabled for the delegator account. 	-
Log Stream Name	<p>Select a log stream.</p> <p>NOTE Log streams that have been configured with OBS transfer settings cannot be configured again.</p>	-
Filter By	<p>Keyword is selected by default. Enter the keyword to be filtered in the text box.</p>	-

Parameter	Description	Example
Log Time Range	<p>There are three types of time range: relative time from now, relative time from last, and specified time. Select a time range as required.</p> <ul style="list-style-type: none"> • From now: queries log data generated in a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from now, the charts on the dashboard display the log data that is generated from 18:20:31 to 19:20:31. • From last: queries log data generated in a time range that ends with the current time, such as the previous 1 or 15 minutes. For example, if the current time is 19:20:31 and 1 hour is selected as the relative time from last, the charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00. • Specified: queries log data that is generated in a specified time range. 	-
Total Log Events	Total number of log events.	-
Log Files	Max log events for each transfer: 20 million. Max transfer files: 200.	-
OBS Bucket	<ul style="list-style-type: none"> • Select an OBS bucket. If no OBS buckets are available, click View OBS Bucket to access the OBS console and create an OBS bucket. • Currently, LTS supports only Standard OBS buckets. • Data cannot be transferred to an OBS bucket whose storage class is Archive or for which cross-region replication has been configured. 	-
Bucket Directory	OBS bucket directory.	-
Transfer File Name	Custom transfer file name. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.	-

Parameter	Description	Example
Format	<p>Storage format of logs. The value can be Raw Log Format, JSON, or CSV.</p> <ul style="list-style-type: none"> • Example of the raw log format: (Logs displayed on the LTS console are in the raw format.) <pre>Sep 30 07:30:01 ecs-bd70 CRON[3459]: (root) CMD (/opt/oss/servicemgr/ICAgent/bin/manual/mstart.sh > /dev/null 2>&1)</pre> • Example of the JSON format: <pre>{"host_name":"ecs-bd70","ip":"192.168.0.54","line_no":249,"message":"Sep 30 14:40:01 ecs-bd70 CRON[4363]: (root) CMD (/opt/oss/servicemgr/ICAgent/bin/manual/mstart.sh > /dev/null 2>&1)\n","path":"/var/log/syslog","time":1569825602303}</pre> • CSV: Log content is displayed in a table. 	Json

Step 3 Click **OK**. When the log transfer status changes to **Normal**, the transfer task has been created.

Step 4 Click the OBS bucket name in the **Transfer Destination** column to switch to the OBS console and view the transferred log files.

Step 5 Download the transferred logs from OBS for viewing.

----End

Modifying a Log Transfer Task

1. Locate the row that contains the target transfer task and click **Modify** in the **Operation** column.
2. Click **OK**.

Viewing Transfer Details

1. Locate the target log transfer task and click **More > Details** in the row of the desired task to view its details.
2. On the displayed **Transfer Details** page, you can view the log transfer details.

Deleting a Log Transfer Task

If logs do not need to be transferred, you can delete the transfer task.

 **NOTE**

- After a transfer task is deleted, log transfer will be stopped. Exercise caution when performing the deletion.
 - After a transfer task is deleted, the logs that have been transferred remain in OBS.
 - When you create a transfer task, OBS will grant read and write permissions to LTS for the selected bucket. If one OBS bucket is used by multiple transfer tasks, perform the following operations to delete the transfer task:
 - If only one transfer task is created using this OBS bucket, delete the bucket access permission granted to specific users on the **Access Control > Bucket ACLs** tab page on the OBS console when you delete the transfer task.
 - If multiple transfer tasks are created using this OBS bucket, do not delete the bucket access permission. Otherwise, data transfer will fail.
1. Locate the row of the target transfer task and choose **Delete** in the **Operation** column.
 2. Click **OK**.

Viewing Transfer Status

The status of a transfer task can be **Normal**, **Abnormal**, or **Disabled**.

- **Normal:** The log transfer task works properly.
- **Abnormal:** An error occurred in the log transfer task. The possible cause is that the access control on the OBS bucket is configured incorrectly. Access the OBS console to correct the settings.
- **Closed:** The log transfer task is stopped.

8 Log Processing

8.1 Processing Logs with FunctionGraph Function Templates

You can create functions from templates or scratch to normalize, transfer, anonymize, and filter logs.

Step 1 Log in to the LTS console.

Step 2 Choose **Log Jobs** in the navigation pane and click the **Function Processing** tab.

Step 3 Create functions from desired templates.

 **NOTE**

For details, see the [Creating a Function Using a Template](#).

----End

9 Configuration Center

9.1 Setting LTS Log Collection Quota

Enabling or Disabling Log Collection Beyond Free Quota

When the monthly free quota (500 MB) is used up, you will be billed for any excess usage on a pay-per-use basis. To avoid extra expenses, you can configure log collection to stop when the quota runs out.

NOTE

- The function is enabled by default. If it is enabled, logs will continue to be collected after the free quota (500 MB) is used up. You will be billed for the excess usage on a pay-per-use basis.
- Log usage, including log read/write, log indexing, and log retention, are billed in LTS. If log collection is disabled when the free quota is used up, no fee is generated for log read/write and indexing because these operations will not be performed. However, log data that beyond the free quota is still retained in LTS and fees are generated for the log retention. When the logs age out after the specified retention period, no fees will be generated.
- If you enable or disable **Continue to Collect Logs When the Free Quota is Exceeded** in AOM, this function will be synchronously enabled or disabled in LTS.

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**.

Step 2 In the navigation pane, choose **Configuration Center**. The **Quota Configuration** tab page is displayed by default.

Step 3 Disable **Continue to Collect Logs When the Free Quota Is Exceeded**.

When the free quota (500 MB) is used up, log collection will be suspended. You can view the current resource usage in **Overview** area on the **Log Management** page. For details, see [Viewing Log Management](#).

----End

9.2 Configuring Log Content Delimiters

You can configure delimiters to split log content into words, so you can search for logs by these words. LTS has preconfigured the following delimiters:

```
,";=()[]{}@&<>/:\\?*\n\t\r
```

If the default delimiters cannot meet your needs, you can set custom delimiters.

NOTE

This function is available only in regions AF-Johannesburg, AP-Singapore, CN-Hong Kong, CN East-Shanghai1, LA-Mexico City1, LA-Mexico City2, LA-Santiago, and LA-Sao Paulo1.

Precautions

- Your custom delimiters are applicable only to the log events generated after the delimiters are configured.
- The delimiter configured on the **Delimiters** tab page takes effect for all log streams in the current region. For details about how to configure delimiters for a single log stream, see [Setting Indexes](#).

Procedure

Step 1 Log in to the management console and choose **Management & Deployment > Log Tank Service**.

Step 2 Choose **Configuration Center** in the navigation pane and click the **Delimiters** tab.

Step 3 Configure delimiters.

You can configure delimiters in either of the following ways. If you use both ways, the delimiters configured in the two ways will all take effect.

- **Common Delimiters:** Click **Edit** and enter custom delimiters in the text box.
- **ASCII Delimiters:** Click **Edit > Add ASCII Delimiter**, and enter ASCII values by referring to [ASCII Table](#).

Step 4 Preview the parsing result.

Enter log content in the text box and click **Preview**.

Step 5 Check whether the parsing result is correct. If it is correct, click **Save**.

NOTE

You can click **Reset** to restore the default delimiters.

----End

ASCII Table

Table 9-1 ASCII table

AS CII Val ue	Character	ASC II Val ue	Character	AS CII Val ue	Character	AS CII Val ue	Character
0	NUL (Null)	32	Space	64	@	96	`
1	SOH (Start of heading)	33	!	65	A	97	a
2	STX (Start of text)	34	"	66	B	98	b
3	ETX (End of text)	35	#	67	C	99	c
4	EOT (End of transmission)	36	\$	68	D	100	d
5	ENQ (Enquiry)	37	%	69	E	101	e
6	ACK (Acknowledge)	38	&	70	F	102	f
7	BEL (Bell)	39	'	71	G	103	g
8	BS (Backspace)	40	(72	H	104	h
9	HT (Horizontal tab)	41)	73	I	105	i
10	LF (Line feed)	42	*	74	J	106	j
11	VT (Vertical tab)	43	+	75	K	107	k
12	FF (Form feed)	44	,	76	L	108	l
13	CR (Carriage return)	45	-	77	M	109	m
14	SO (Shift out)	46	.	78	N	110	n
15	SI (Shift in)	47	/	79	O	111	o
16	DLE (Data link escape)	48	0	80	P	112	p

AS CII Value	Character	ASC II Value	Character	AS CII Value	Character	AS CII Value	Character
17	DC1 (Device control 1)	49	1	81	Q	113	q
18	DC2 (Device control 2)	50	2	82	R	114	r
19	DC3 (Device control 3)	51	3	83	S	115	s
20	DC4 (Device control 4)	52	4	84	T	116	t
21	NAK (Negative acknowledge)	53	5	85	U	117	u
22	SYN (Synchronous idle)	54	6	86	V	118	v
23	ETB (End of transmission block)	55	7	87	W	119	w
24	CAN (Cancel)	56	8	88	X	120	x
25	EM (End of medium)	57	9	89	Y	121	y
26	SUB (Substitute)	58	:	90	Z	122	z
27	ESC (Escape)	59	;	91	[123	{
28	FS (File separator)	60	<	92	\	124	
29	GS (Group separator)	61	=	93]	125	}
30	RS (Record separator)	62	>	94	^	126	~
31	US (Unit separator)	63	?	95	_	127	DEL (Delete)

9.3 Setting ICAgent Collection

On the **ICAgent Collection** tab page, you can disable or enable ICAgent collection (that is, specify whether ICAgent collects log data), syslog log collection to AOM, and container standard output to AOM.

Setting Collection

- Step 1** Log in to the management console and choose **Management & Deployment > Log Tank Service**.
- Step 2** Choose **Configuration Center** in the navigation pane and click the **ICAgent Collection** tab.
- Step 3** Toggle the **ICAgent Collection** switch to control whether ICAgent collects logs.
 - **ICAgent Collection** is toggled on by default. If you do not need to collect logs, toggle it off to reduce resource usage.
 - After it is toggled off, ICAgent will stop collecting logs, and the log collection function on the AOM console will also be disabled.
- Step 4** Toggle the **Collect Syslog Logs to AOM** switch to set whether ICAgent collects Syslogs to AOM 1.0. If it is toggled off, ICAgent does not collect Syslog logs to AOM 1.0. This function is only available with ICAgent 5.12.182 and later.
- Step 5** **Output to AOM**: Select a CCE cluster and toggle on or off the **Apply to Cluster** switch for it. If it is toggled off, ICAgent does not collect the cluster stdout logs to AOM. This function is only available with ICAgent 5.12.133 and later. You are advised to collect container standard output to LTS instead of AOM. For details, see [Ingesting CCE Application Logs to LTS](#).

----End