Alibaba Cloud Log Service

User Guide

Issue: 20190621

MORE THAN JUST CLOUD | Alibaba Cloud

Legal disclaimer

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

1. **You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.**
2. **No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.**
3. **The content of this document may be changed due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.**
4. **This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults " and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity , applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use**

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus , page names, and other UI elements.	Click OK.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

Legal disclaimer..... I
Generic conventions..... I
1 Preparation..... 1
1.1 Preparation..... 1
1.2 Manage a project..... 5
1.3 Manage a Logstore..... 6
1.4 Manage a Shard..... 12
2 Data Collection..... 17
2.1 Collection methods..... 17
2.2 Collection acceleration..... 20
2.2.1 Overview..... 20
2.2.2 Enable Global Acceleration..... 24
2.2.3 Configure Logtail collection acceleration..... 30
2.2.4 Disable Global Acceleration..... 32
3 Logtail collection..... 34
3.1 Overview..... 34
3.1.1 Overview..... 34
3.1.2 Logtail collection process..... 40
3.1.3 Logtail configuration and recording files..... 43
3.2 Select a network type..... 54
3.3 Install..... 59
3.3.1 Install Logtail in Linux..... 59
3.3.2 Install Logtail in Windows..... 70
3.3.3 Configure startup parameters..... 77
3.4 Machine Group..... 86
3.4.1 Overview..... 86
3.4.2 Create a machine group with an IP address as its identifier..... 88
3.4.3 Create an ID to identify a machine group..... 91
3.4.4 Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs..... 96
3.4.5 Create a Logtail configuration..... 98
3.4.6 Manage a machine group..... 100
3.5 Text logs..... 105
3.5.1 Collect text logs..... 105
3.5.2 Configure and parse text logs..... 117
3.5.3 Text logs - Configure time format..... 119
3.5.4 Import history logs..... 122
3.5.5 Set a log topic..... 125
3.6 Container log collection..... 128
3.6.1 Collect standard Docker logs..... 129

3.6.2 Kubernetes log collection process..... 135
3.6.3 Container text logs.....146
3.6.4 Containers-standard output.....153
3.6.5 Configure Kubernetes log collection on CRD.....165
3.6.6 Kubernetes-Sidecar log collection mode..... 174
3.7 Limits..... 187
4 Cloud product collection..... 192
4.1 Cloud product logs.....192
4.2 API Gateway Access Log..... 193
4.3 Access logs of Layer-7 Server Load Balancer..... 196
4.4 DDoS log collection..... 202
4.4.1 Overview.....202
4.4.2 Collection procedure.....205
4.4.3 Log analysis..... 211
4.4.4 Log Report.....224
4.4.5 Billing method..... 236
4.5 Logs of BGP-line Anti-DDoS Pro..... 241
4.5.1 Overview.....241
4.5.2 Enable or disable log collection.....246
4.5.3 Manage log storage space.....249
4.5.4 Log fields..... 250
4.5.5 Analyze the logs of BGP-line Anti-DDoS Pro..... 253
4.5.6 Log reports of BGP-line Anti-DDoS Pro..... 266
4.5.7 Advanced settings..... 276
4.6 TDS logs.....279
4.7 WAF logs.....281
4.7.1 Real-time log analysis..... 281
4.7.2 Billing method..... 283
4.7.3 Activate WAF Log Service..... 285
4.7.4 Log collection..... 286
4.7.5 Log Analyses.....290
4.7.6 Log Reports..... 305
4.7.7 Fields in the log entry..... 318
4.7.8 Advanced settings..... 323
4.7.9 Grant log query and analysis permissions to a RAM user..... 324
4.7.10 Manage log storage.....327
4.8 Anti-Bot logs..... 328
启用Anti-Bot日志采集.....328
日志字段说明..... 330
4.9 ActionTrail access logs..... 335
4.9.1 Overview.....335
4.9.2 Procedure..... 340
5 Other collection methods..... 347
5.1 Web Tracking 347

5.2 Use the Kafka protocol to collect data to Log Service..... 352
5.3 Use the Syslog protocol to collect data to Log Service.....363
5.4 Logstash.....371
5.4.1 Install Logstash.....371
5.4.2 Set Logstash as a Windows service..... 373
5.4.3 Create Logstash collection configurations.....375
5.4.4 Advanced functions.....377
5.4.5 Logstash error processing..... 378
5.5 SDK collection.....378
5.5.1 Producer Library..... 378
5.5.2 Log4j Appender.....381
5.5.3 C Producer Library..... 381
5.6 Common log formats.....382
5.6.1 Apache log.....382
5.6.2 Nginx logs 387
5.6.3 Python logs.....389
5.6.4 Log4j logs..... 396
5.6.5 Node.js logs..... 397
5.6.6 WordPress logs..... 399
5.6.7 Delimiter logs..... 400
5.6.8 JSON logs.....406
5.6.9 ThinkPHP logs.....409
5.6.10 Use Logstash to collect IIS logs..... 410
5.6.11 Use Logstash to collect CSV logs..... 411
5.6.12 Use Logstash to collect other logs..... 414
5.6.13 Unity3D logs.....415
6 Index and query.....418
6.1 Overview.....418
6.2 Syntax description.....420
6.3 Enable and set indexes.....424
6.4 Query logs.....432
6.5 Download logs..... 438
6.6 Data type of index.....441
6.6.1 Overview.....441
6.6.2 Text type.....444
6.6.3 JSON type..... 446
6.6.4 Value type.....447
6.7 Query.....447
6.7.1 Query syntax.....447
6.7.2 LiveTail.....454
6.7.3 Use LogReduce to group log data.....460
6.7.4 Context query..... 468
6.7.5 Saved search..... 471
6.7.6 Quick analysis.....474
6.7.7 Other functions.....479

6.8 Analysis grammar..... 482
6.8.1 General aggregate functions..... 482
6.8.2 Security detection functions..... 484
6.8.3 Mapping function..... 488
6.8.4 Estimating functions..... 490
6.8.5 Mathematical statistics functions..... 492
6.8.6 Mathematical calculation functions..... 493
6.8.7 String functions..... 495
6.8.8 Date and time functions..... 497
6.8.9 URL functions..... 503
6.8.10 Regular expression functions..... 504
6.8.11 JSON functions..... 505
6.8.12 Type conversion functions..... 506
6.8.13 IP functions..... 507
6.8.14 GROUP BY syntax..... 510
6.8.15 Window functions..... 512
6.8.16 HAVING syntax..... 514
6.8.17 ORDER BY syntax..... 515
6.8.18 LIMIT syntax..... 515
6.8.19 Case when and if branch syntax..... 516
6.8.20 Nested subquery..... 518
6.8.21 Arrays..... 518
6.8.22 Binary string functions..... 521
6.8.23 Bit operation..... 523
6.8.24 Interval-valued comparison and periodicity-valued comparison functions..... 523
6.8.25 Comparison functions and operators..... 530
6.8.26 Lambda functions..... 532
6.8.27 Logical functions..... 536
6.8.28 Column alias..... 537
6.8.29 Geospatial functions..... 537
6.8.30 Geo functions..... 541
6.8.31 Join syntax..... 542
6.8.32 UNNEST function..... 543
6.8.33 Phone number functions..... 546
6.9 Machine learning syntax and functions..... 548
6.9.1 Introduction..... 548
6.9.2 Smooth function..... 551
6.9.3 Multi-period estimation function..... 556
6.9.4 Change point detection function..... 558
6.9.5 Maximum value detection function..... 562
6.9.6 Prediction and anomaly detection function..... 563
6.9.7 Sequence decomposition function..... 569
6.9.8 Time series clustering function..... 570
6.9.9 Frequent pattern statistical function..... 576

6.9.10 Differential pattern statistical function..... 577
6.10 Advanced analysis..... 579
6.10.1 Case study..... 579
6.10.2 Optimize query for analysis..... 581
6.11 Use JDBC to query and analyze logs..... 583
7 Query and visualization..... 587
7.1 Analysis graph..... 587
7.1.1 Graph description..... 587
7.1.2 Table..... 588
7.1.3 Line chart..... 592
7.1.4 Column chart..... 595
7.1.5 Bar chart..... 597
7.1.6 Pie chart..... 599
7.1.7 Area chart..... 603
7.1.8 Single value chart..... 605
7.1.9 Flow chart..... 612
7.1.10 Tree map..... 614
7.2 Dashboard..... 615
7.2.1 Dashboard overview..... 615
7.2.2 Create and delete a dashboard..... 617
7.2.3 Set the display parameters for a dashboard..... 623
7.2.4 Edit a dashboard..... 628
7.2.5 Subscribe to dashboard snapshots..... 636
7.2.6 Drill-down analysis..... 640
7.2.7 Set and use a filter in a dashboard..... 650
7.2.8 Markdown chart..... 660
7.3 Other visualization methods..... 667
7.3.1 Console sharing embedment..... 667
7.3.2 Use JDBC to count and visualize logs..... 670
7.3.3 OpenTracing implementation of Jaeger..... 678
7.3.4 Interconnect with DataV big screen..... 686
7.3.5 Interconnection with Grafana..... 697
8 Alarm..... 712
8.1 Alarm function overview..... 712
8.2 Configure an alarm..... 715
8.2.1 Configure an alarm..... 715
8.2.2 Set alarm notifications..... 725
8.2.3 Grant a RAM user the permissions to configure alarms..... 733
8.3 Modify and view an alarm..... 736
8.3.1 Modify an alarm configuration..... 736
8.3.2 View and use alarm logs..... 739
8.3.3 Manage an alarm..... 741
8.3.4 Upgrade an alarm configuration to the latest version..... 744
8.4 Relevant syntax and fields for reference..... 747

8.4.1 Set an alarm condition expression....747
8.4.2 Alarm log fields.... 753
9 Real-time subscription and consumption....758
9.1 Overview....758
9.2 Preview log data....759
9.3 Consumption by consumer groups.... 760
9.3.1 Use a consumer group to consume logs.... 760
9.3.2 View consumer group status.... 767
9.3.3 Consumer group - Monitoring alarm.... 770
9.4 Use Fuction Compute to cosume LogHub Logs.... 773
9.4.1 Development guide.... 773
9.4.2 Configure Function Compute log consumption....778
9.5 Use Flink to consume LogHub logs.... 788
9.6 Use Storm to consume LogHub logs....797
9.7 Use Spark Streaming to consume LogHub logs.... 801
9.8 Use CloudMonitor to consume LogHub logs.... 801
10 Data shipping....802
10.1 Overview....802
10.2 Ship logs to OSS.... 802
10.2.1 Ship logs to OSS.... 802
10.2.2 JSON storage.... 811
10.2.3 CSV storage....813
10.2.4 Parquet storage.... 816
10.2.5 Advanced RAM authorization.... 820
10.3 Ship data to MaxCompute....823
10.3.1 Ship data to MaxCompute by using DataWorks.... 823
10.4 Send logs to an SIEM system....837
10.4.1 Overview....837
10.4.2 Use HTTPS to send logs to an SIEM system....839
10.4.3 Use Syslog to send logs to an SIEM system....846
10.5 Manage LogShipper tasks.... 854
11 Log Service Monitor.... 856
11.1 Monitor Log Service.... 856
11.2 Service log.... 858
11.2.1 Service log overview....858
11.2.2 Use the service log function....861
11.2.3 Log types.... 864
11.2.4 Service log dashboards....877
11.3 Log monironing by CloudMonitor....877
11.3.1 Log Service monitoring metrics....878
11.3.2 Use CloudMonitor to set alarm rules....882
12 Access control RAM....891
12.1 Authorization - Overview....891
12.2 Grant a RAM user the permissions to access Log Service....894

12.3 Custom RAM authorization.....895
12.4 Service role.....901
12.5 User Role.....903

1 Preparation

1.1 Preparation

Log Service provides multiple log collection methods. You can use Log Service to collect Elastic Compute Service (ECS) logs, local server logs, IoT device logs, and other cloud product logs.

Before using Log Service, you must first make the following preparations.

Procedure

1. Activate Log Service

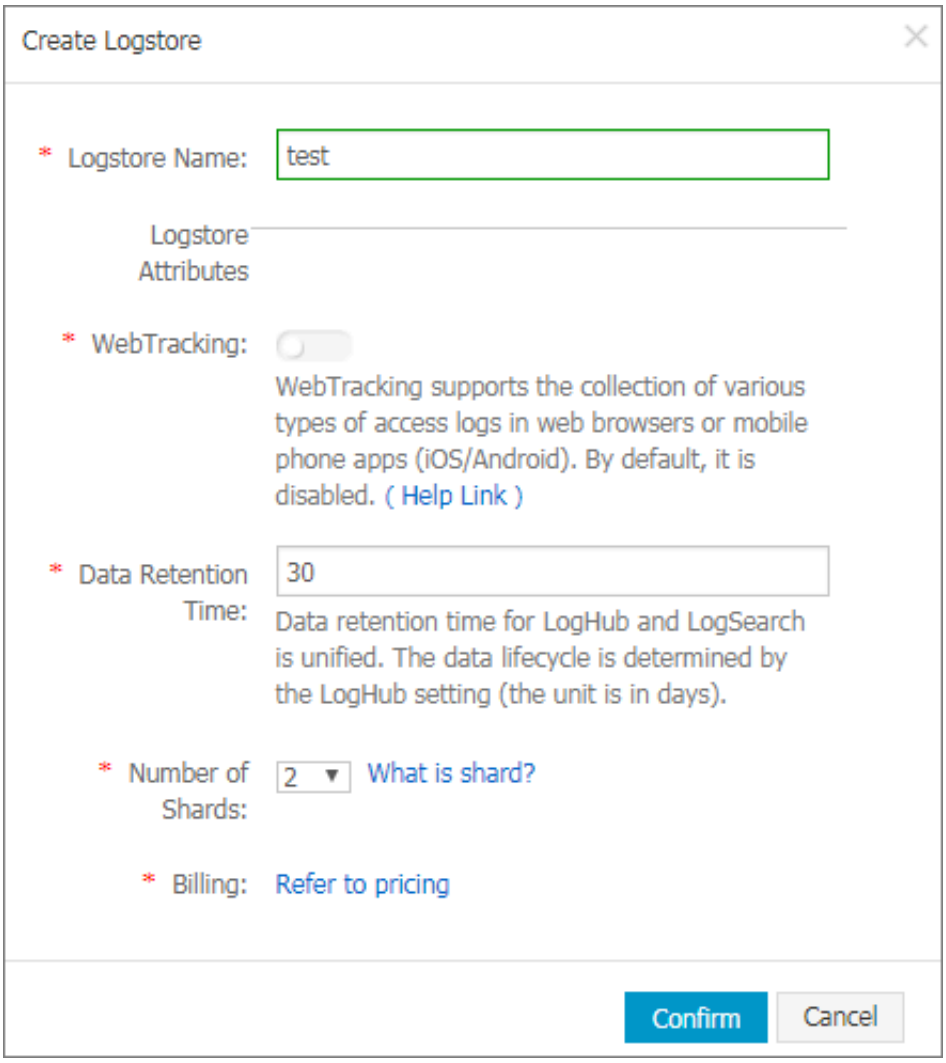
   Log on to the *Log Service product page* with your registered Alibaba Cloud account. Click Get it Free. The system automatically redirects to the purchase page. Select the I agree with Log Service Agreement of Service check box and then click Enable Now to activate Log Service.

2. Ceate and enable AccessKey (for API/SDKs).

   An AccessKey is required to collect logs by using Logtail. Before using Log Service, you must create an AccessKey.

   In the *Log Service console*, hover your mouse over your avatar in the upper-right corner. Select accesskeys from the drop-down list . Click Continue to manage

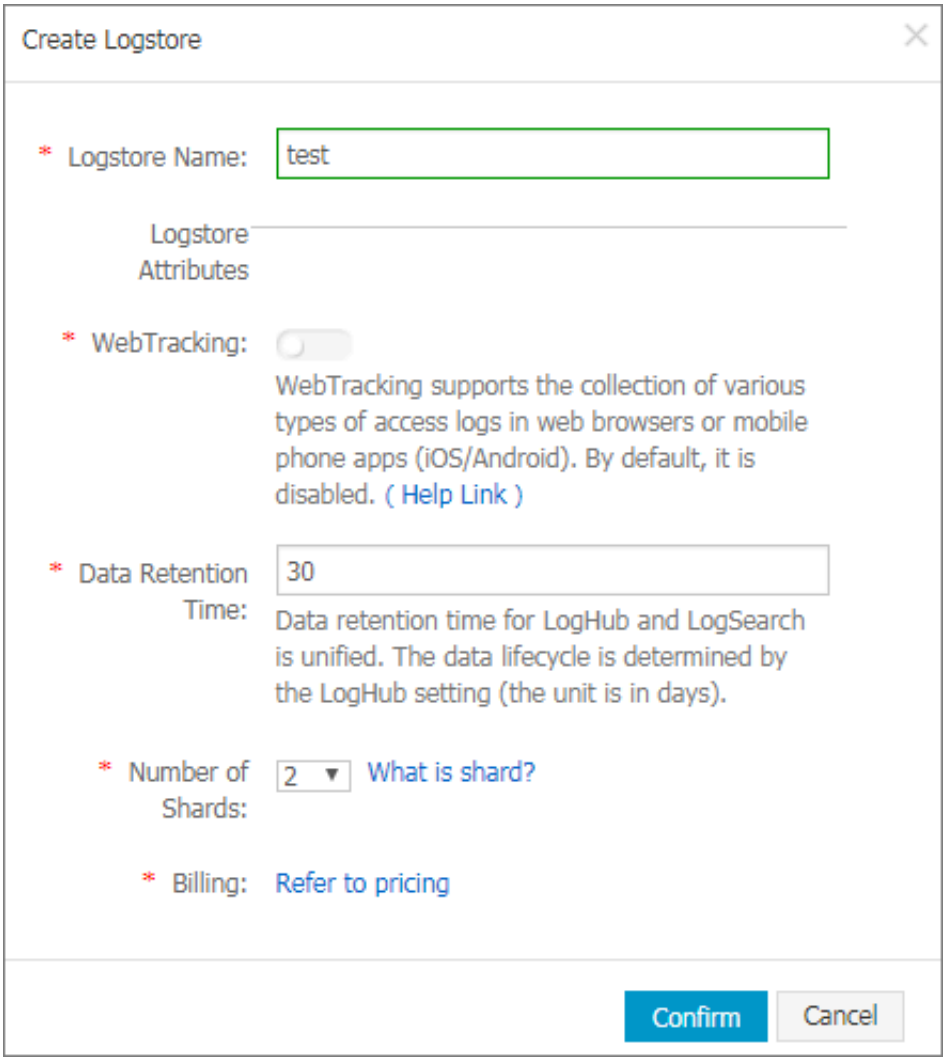
AccessKey in the appeared dialog box. The Access Key Management page appears. Create an AccessKey and check whether the created AccessKey is enabled.



3. Create a project

   When you log on to the Log Service console for the first time, the system prompts you to create a project. You can also click Create a project in the upper-right corner.

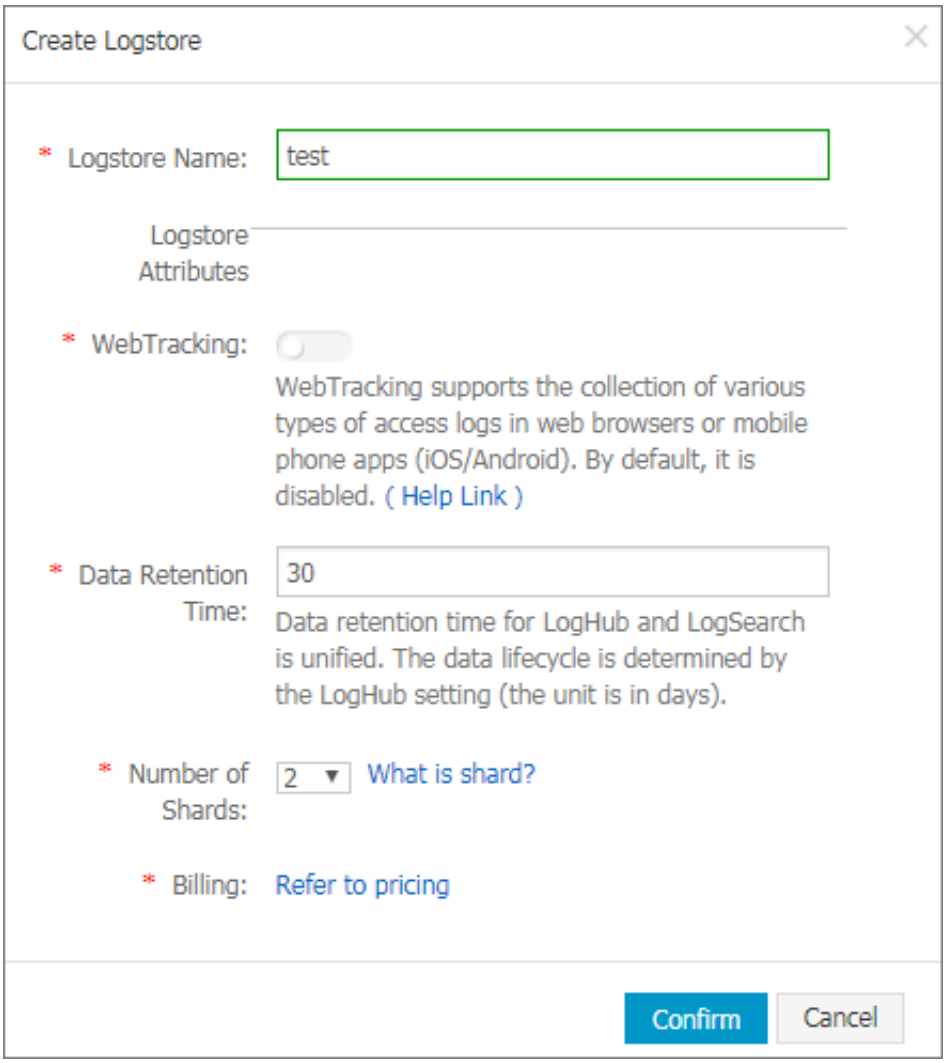
   You can also modify project description and delete a project. For more information, see *Manage a project*.



4. Create Logstore.

   The system prompts you to create a Logstore after you create a project. You can also click the project name and then click Create in the upper-right corner. When creating a Logstore, you must specify how you are going to use these logs.

   You can also modify or delete the Logstore. For more information, see *Manage a Logstore*.



5. Manage shards (optional)

   When creating a Logstore, you can select the number of shards based on the volume and generation speed of your logs. You can also change the number of shards by splitting or merging shards when modifying the Logstore.

   For more information about splitting and merging shards, see *Manage a Shard*.

6. Perform RAM authorization (optional)

   If you need to collect cloud product logs or post Log Service data to OSS or another product for storage and analysis, you must grant the relevant permissions for Log Service or other cloud products.

   To use a sub-account to perform operations in Log Service, you must grant permissions to the sub-account in the Resource Access Management (RAM) console.

   For more information about the authorization policies and procedure, see *Authorization - Overview*.

1.2 Manage a project

In the Log Service console, you can: Create a project. Delete a project.

Create a project



Note:

- Currently, Log Service can only create projects in the console.
- The project name must be globally unique among all Alibaba Cloud regions. The message "Project XXX already exists" is displayed if the project name you entered has already been used by another user. Enter another project name and try again.
- To create a project, you must specify the Alibaba Cloud region based on the source of the logs to be collected and other actual conditions. To collect logs from an Alibaba Cloud Elastic Compute Service (ECS) instance, we recommend that you create the project in the same region as the ECS instance to speed up log collection, and collect logs by using Alibaba Cloud intranet (without occupying the Internet bandwidth of the ECS instance).
- The region in which the project resides cannot be changed after the project is created. Log Service currently does not support migrating projects, so proceed with caution when selecting the region in which the project resides.
- You can create up to 50 projects in all Alibaba Cloud regions.

Procedure

1. Log on to the Log Service console.

2. Click Create Project in the upper-right corner.
3. Enter the Project Name and select the Region. Then, click Confirm.

Configuration items	Description
Project name	Enter the project name. The name can be 3–63 characters long, contain lowercase letters, numbers, and hyphens (-), and must begin and end with a lowercase letter or number.  Note: The project name cannot be modified after the project is created.
Description	Enter a simple description for the project. After the project is created, the description is displayed on the Project List page. It can be modified by clicking Modify at the right of the project on the Project List.
Region	You must specify an Alibaba Cloud region for each project. The region cannot be modified after the project is created, and the project cannot be migrated among regions.

Delete a project

You can delete a project in some situations, such as disabling Log Service and deleting all the logs in a project. Log Service allows you to delete a project in the console.

 Note:
After a project is deleted, all the log data and configuration information managed by this project are permanently released and are not recoverable. Therefore, proceed with caution when deleting a project to avoid data loss.

1. Log on to the Log Service console.
2. On the Project List page, click Delete

1.3 Manage a Logstore

A Logstore is a collection of resources created in a project. All data in a Logstore is from the same data source. The Logstore is a unit to query, analyze, and ship the collected log data. In the Log Service console, you can:

- *Create a Logstore.*
- *Modify Logstore configurations*
- *Delete a Logstore*

Create a Logstore.

> Note:
>
> - Each Logstore must be created in the certain Project.
> - Up to 200 Logstores can be created for each Log Service Project.
> - The Logstore name must be unique in the project.
> - The data retention time can be modified after a Logstore is created. Click Modify at the rightof the Logstore > on the Logstore List page, change the Data Retention Time and then click Modify.

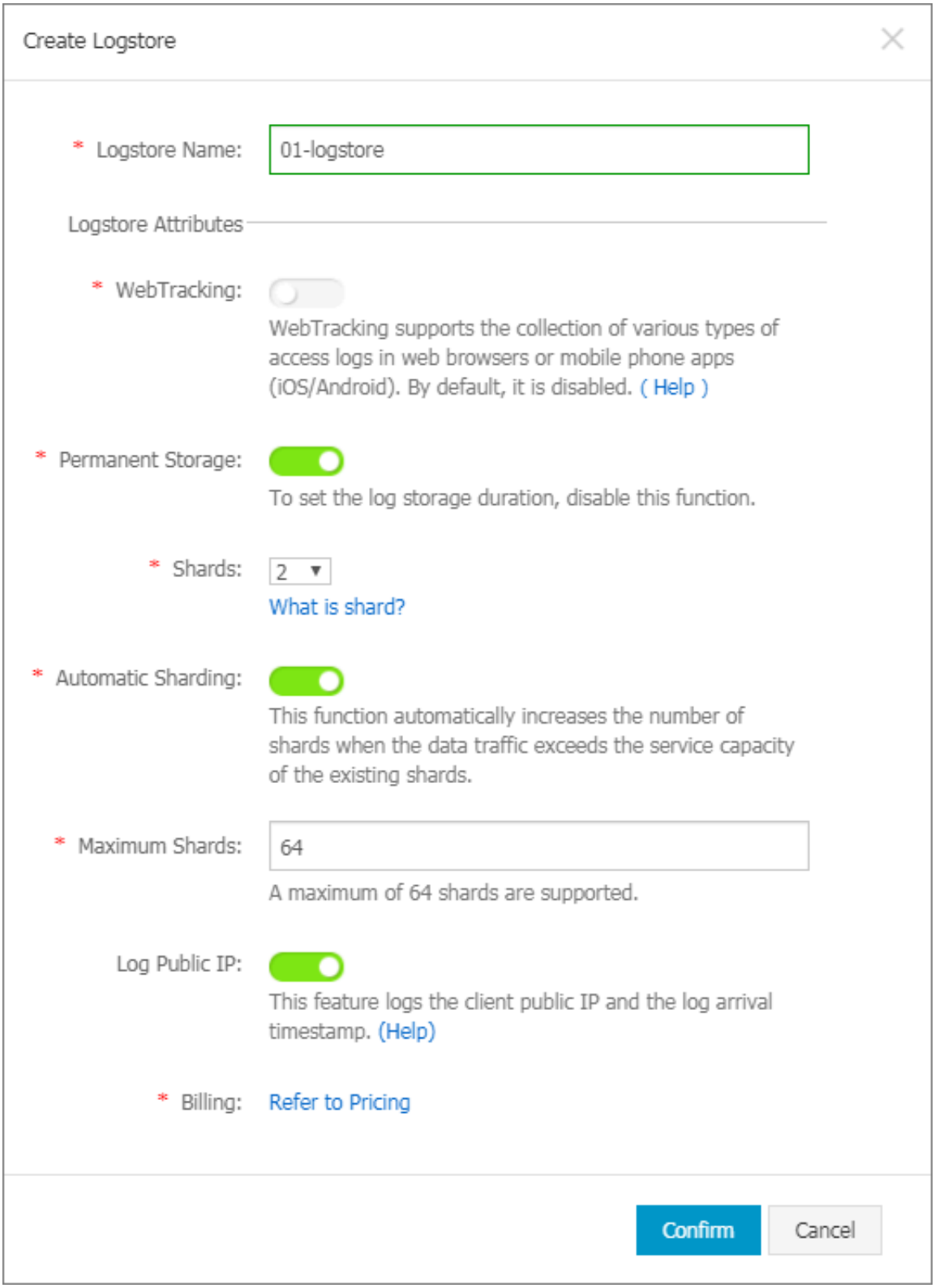
1. On the Project List page, click the project name. Click Create to create a Logstore.

   You can also click Create in the dialog box after creating a project.
2. Complete the configurations and click Confirm.

Configuration	Description
Logstore name	The name can be 3–63 characters long, contain lowercase letters, numbers, hyphens (-), and underscores (_), and must begin and end with a lowercase letter or number. The Logstore name, which must be unique in the project where it belongs. Note: The Logstore name cannot be modified after the Logstore is created.
WebTracking	Select whether or not to enable the WebTracking function . This function supports collecting log data from HTML, H5, iOS, or Android platform to Log Service. Disabled by default.

Configuration	Description
Permanent storage	Select whether or not to enable the Eternal save function. The Log Service allows to save the collected logs permanently. You can also disable the function and customize the Data Retention Time .
Data Retention Time	The time (in days) the collected logs are kept in the Logstore. It can be 1–3000 days. Logs are deleted if the Logs are deleted if the specified time is exceeded. If you disabled the Eternal save function, you need to customize the Data Retention Time .
Number of shards	The number of shards for the Logstore. Each Logstore can create 1–10 shards and each project can create at most 200 shards.
Auto-split	Select whether or not to enable the Auto-split function. It is enabled by default. When the data traffic exceeds the service capacity of the shard, enabling the Auto-split function will increase the number of shards automatically. For details, see *Manage a Shard*.
Maximum number of splits	The maximum number after the maximum shard auto split, the maximum automatic split to 64 partitions can be supported. If you turn onAuto split shard function, you need to setMaximum number of splits.

Configuration	Description
Log Public IP	This function allows the server to automatically add the following information to the tag filed of the received logs: • `__client_ip__` : refers to the Internet IP address of the device that generated the logs. • `__receive_time__` : refers to the time (in the format of the Unix timestamp) when logs arrive at the server.

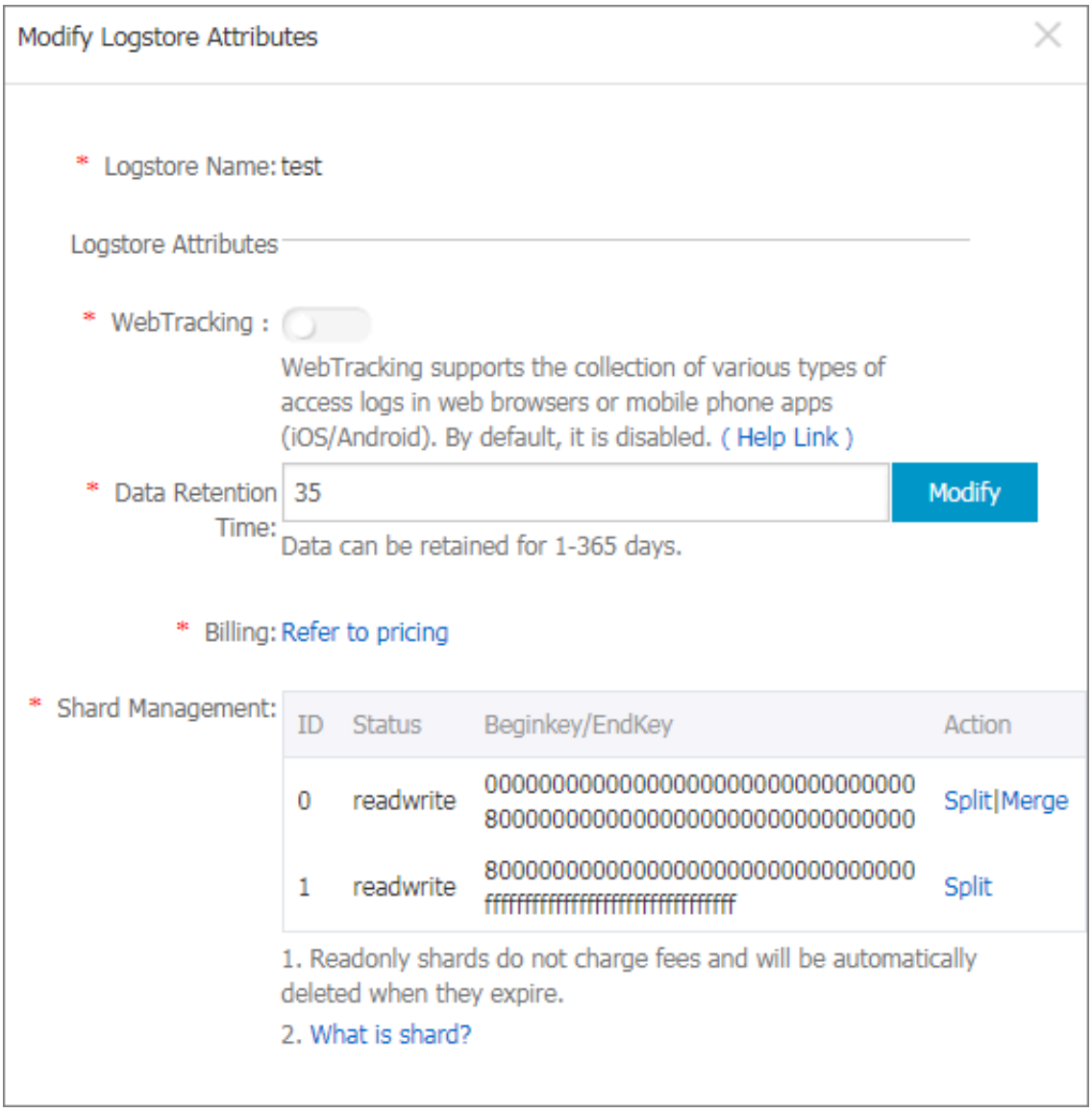


Modify Logstore configurations

After a Logstore is created, you can modify the Logstore configurations as needed.

1. Log on to the Log Service console.
2. On the Project List page, click the project name.

3. On the Logstore List page, click Modify at the right of the Logstore.
4. The Modify Logstore Attributes dialog box appears. Modify the Logstore configurations and then close the dialog box.



Delete a Logstore

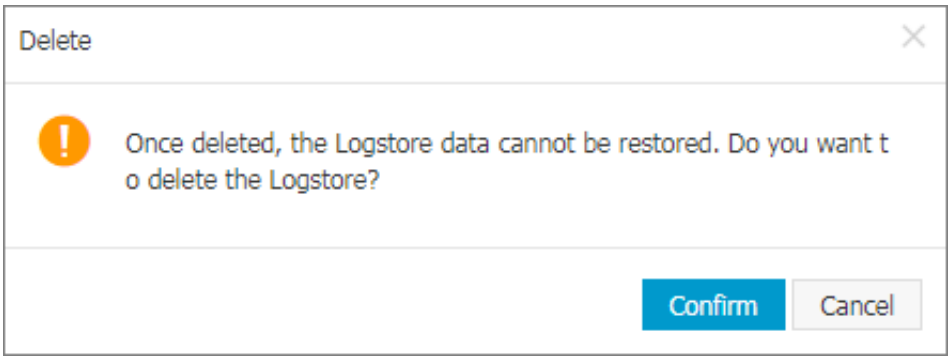
You can delete a Logstore in some situations, such as abandoning a Logstore. Log Service allows you to delete a Logstore in the console. Log Service allows you to delete a Logstore in the console.

Note:

- After a Logstore is deleted, the log data stored in the Logstore is permanently deleted and unrecoverable. So proceed with caution.
- Delete all the corresponding Logtail configurations before deleting a Logstore.

- If the system prompts that you are not granted the corresponding permission when you use a primary account to delete a Logstore, you need to open a ticket.

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Delete at the right of the Logstore you are about to delete.
4. Click OK in the displayed dialog box.



1.4 Manage a Shard

Logstore read/write logs must be stored in a certain shard. Each Logstore is divided into several shards. You must specify the number of shards when creating a Logstore. You can also split a shard or merge shards to increase or reduce the number of shards .

For existing shards, you can:

- *Split a shard*
- *Merge shards*
- *Delete a shard*

Split a shard

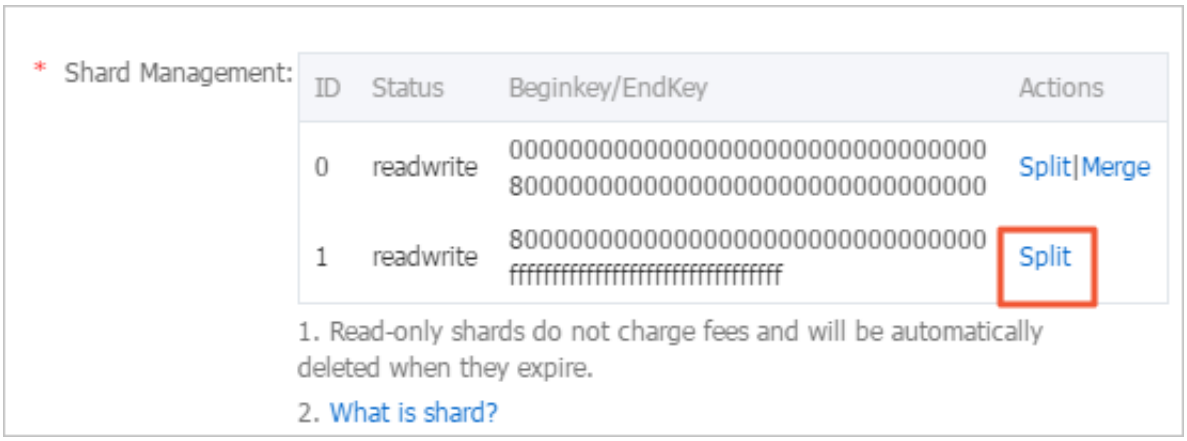
Each shard can write data at 5 MB/s and read data at 10 MB/s. When the data traffic exceeds the service capacity of the shard, we recommend that you increase the number of shards in time by splitting a shard. The expansion partition is completed by split operation.

Instructions

When splitting a shard, you must specify a ShardId in readwrite status and an MD5. The MD5 must be greater than the shard BeginKey and less than the shard EndKey.

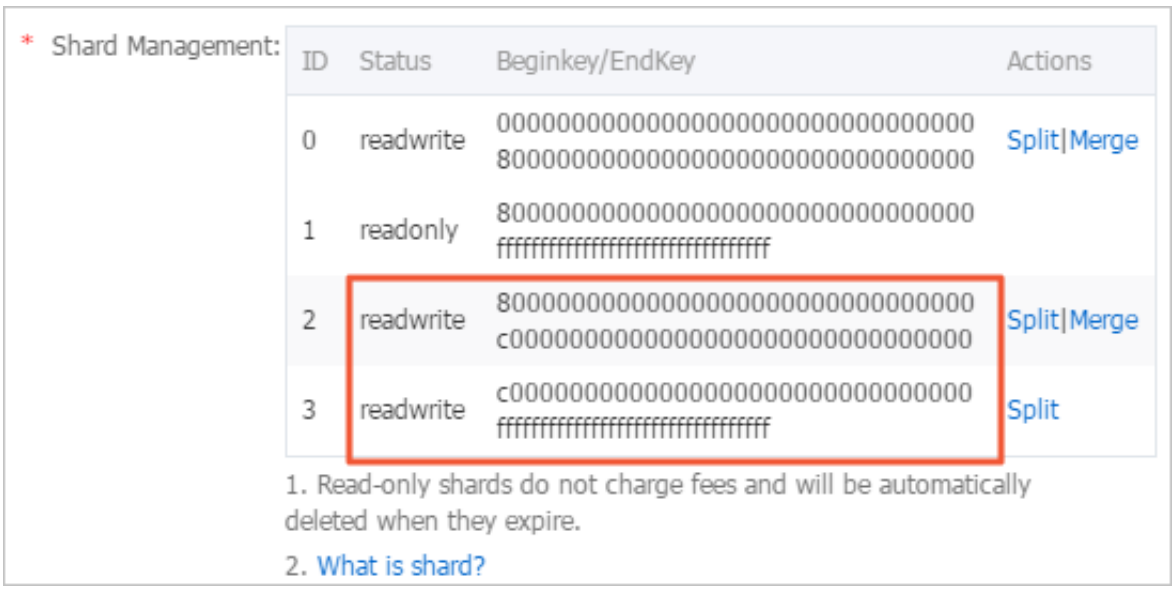
Split operations can split two other shards from one, that is, the number of shards is increased by 2 after the split. After the split, the status of the original shard specified to be split is changed from readwrite to readonly. Data can still be consumed, while new data cannot be written. The two newly generated shards are in readwrite status and arranged behind the original shard. The MD5 range of these two shards covers the range of the original shard.

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Modify at the right of the Logstore.
4. Click Split at the right of the shard to be split.



5. Click Confirm and close the dialog box.

   After the split, the status of the original shard is changed to readonly, and the MD5 range of the two newly generated shards covers the range of the original shard.



Merge shards

You can reduce the number of shards by merging shards. The ranges of the specified shard and the adjacent shard on the right are merged. A new shard in readwrite status is generated and its MD5 range covers the total range of the original two shards. The original two shards are now in the readonly status.

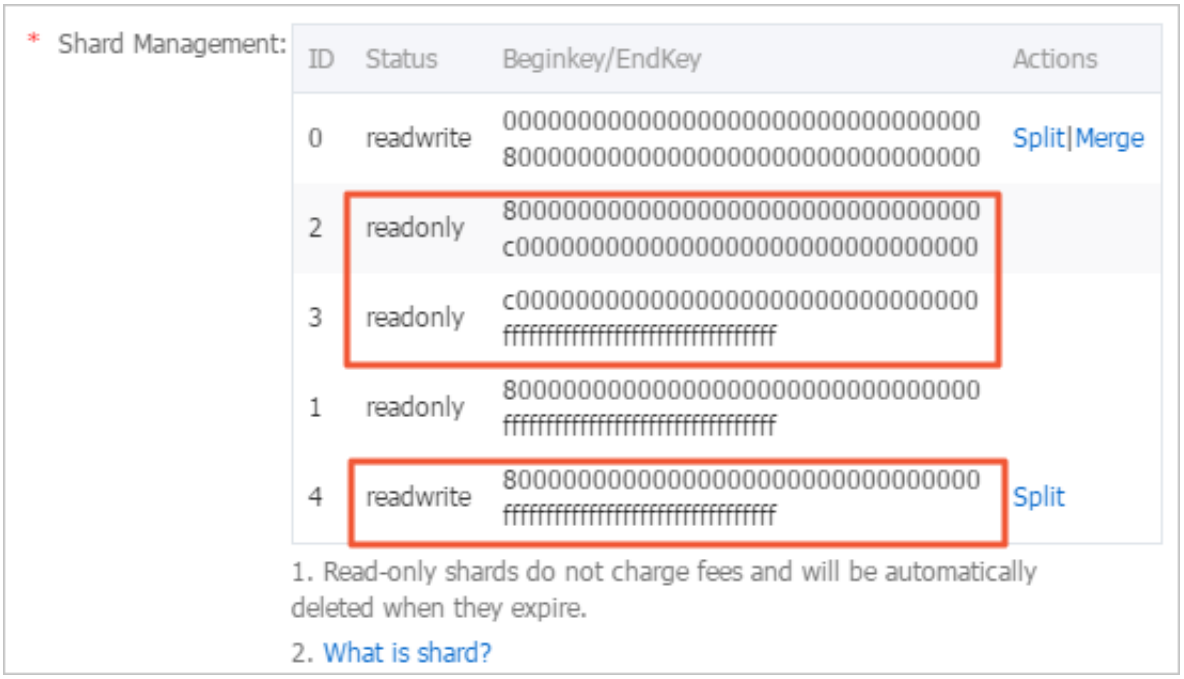
Instructions

When merging shards, you must specify a shard in readwrite status. Make sure the specified shard is not the last shard in readwrite status. The server automatically finds the adjacent shard at the right of the specified shard and merges these two shards. After the merge, the specified shard and the adjacent shard on the right are in readonly status. Data can still be consumed, while new data cannot be written. A new shard in readwrite status is generated and its MD5 range covers the total range of the original two shards.

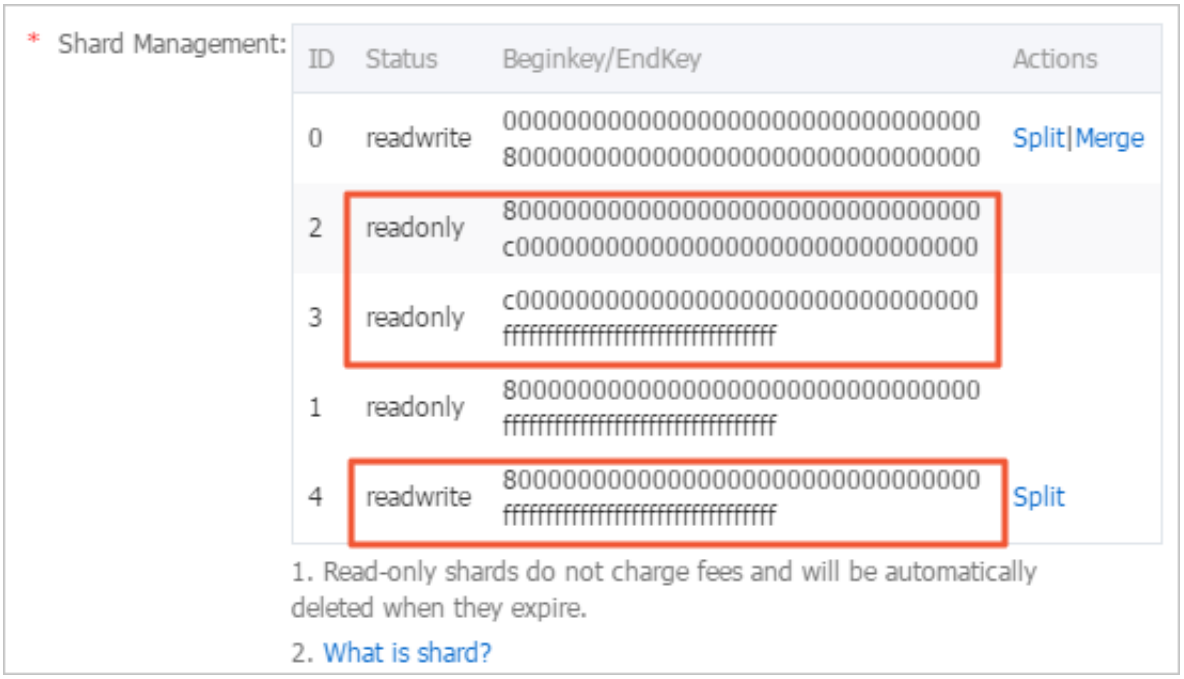
Procedure

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Modify at the right of the Logstore.

4. Click Merge at the right of the shard to be merged.



After the merge, the specified shard and the adjacent shard on the right are changed to the readonly status, and the MD5 range of the newly generated shard in readwrite status covers the total range of the original two shards.



Delete a shard

The Logstore lifecycle, namely, the data retention time can be configured as permanently and 1–3000 days. Shards and log data in the shards are automatically

deleted after the specified data retention time. Shards in readonly status are free of charge.

You can also delete all the shards in a Logstore by deleting a Logstore.

2 Data Collection

2.1 Collection methods

LogHub supports a variety of RESTful APIs that provide different log collection methods, for example, log collection through one or more clients, websites, protocols , SDKs, and APIs.

Data sources

Log Service can collect logs from the following sources:

Type	Source	Access method	Details
Application	Program output	*Logtail*	-
	Access logs	*Logtail*	*Collect and analyze Nginx access logs*
	Link track	Jaeger Collector and *Logtail*	-
Programming language	Java	*SDK* and *Java Producer Library*	-
	Log4J Appender	*1.x* and *2.x*	-
	LogBack Appender	*LogBack*	-
	C	*Native*	-
	Python	*Python*	-
	Python Logging	Python Logging Handler	-
	PHP	*PHP*	-
	C#	*C#*	-
	C++	*C++ SDK*	-
	Go	*Go*	-
	NodeJS	*NodeJs*	-
	JS	*JS/Web Tracking*	-
OS	Linux	*Logtail*	-

Type	Source	Access method	Details
	Windows	*Logtail*	-
	Mac/Unix	*Native C*	-
	Docker files	*Logtail file collection*	-
	Docker output	*Logtail container stdout*	-
Mobile client	iOS/Android	*iOS SDK* and *Android SDK*	-
	Websites	*JS/Web Tracking*	-
	Intelligent IoT	*C Producer Library*	-
Cloud product	Various products, such as ECS and OSS For more information, see *Cloud product logs*	Cloud product console	*Cloud product logs*
	Import MaxCompute data	Use Dataworks to export MaxCompute data	*#unique_25*
Third-party software	Logstash	*Logstash*	-

The following table lists the cloud products from which Log Service can collect logs:

Type	Cloud product name	Activation method	Details
Elastic computing	Elastic Compute Service (ECS)	Through Logtail installation	*Logtail introduction*
	Container Service/ Container Service for Kubernetes	Through the Container Service console	*Text logs* and *stdout*
Storage	Object Storage Service (OSS)	Through the OSS console	*OSS access logs*

Type	Cloud product name	Activation method	Details
Network	Server Load Balancer (SLB)	Through the SLB console	*Access logs of Layer-7 SLB*
	Virtual Private Cloud (VPC)	Through the VPC console	*Flow logs*
	API Gateway	Through the API Gateway console	*API Gateway access logs*
Security	ActionTrail	Through the ActionTrail console	*ActionTrail overview*
	DDoS Protection	Through the DDoS Protection console	*DDOS Protection overview*
	Threat Detection Service	Purchase Threat Detection Service Enterprise Edition and activate the service in the Threat Detection Service console.	*Log retrieval*
	Anti-Bot Service	Through the Anti-Bot Service console	*Anti-Bot Service logs*
Application	Log Service (LOG)	Through the Log Service console	*Log Service overview*

Network and access point selection

Log Service provides *service endpoints* in each region, and the following types of network access methods are supported:

- (Recommended) Intranet (classic networks) and private networks (VPCs): are applicable to regions with smooth service access and high-quality bandwidth links.
- Internet (classic networks): can be used without any limits. The access speed depends on the link quality. HTTPS is recommended to maintain access security.

FAQ

- Q: Which type of network applies to physical connections?

  A: Intranet/private networks

- Q: Can Internet IP addresses be collected during Internet data collection?

  A: Yes. You can follow the instructions provided in *Manage a Logstore* and enable the Internet IP address recording function.
- Q: Which type of network can I use if I want to collect logs from an ECS server located in region A and send them to a project on a Log Service server located in region B?

  A: Use the Internet to transfer logs after install the Internet-version Logtail on the ECS server. As for other scenarios, follow the instructions provided in *Select a network type*.
- Q: How can I determine whether access is established successfully?

  A: Access is established successfully if information is returned after you run the following command:

  ```
  curl $myproject.cn-hangzhou.log.aliyuncs.com
  ```

  In this command, `$myproject` indicates the project name, and `cn-hangzhou.log.aliuncs.com` indicates the access point.

2.2 Collection acceleration

2.2.1 Overview

Log Service adds a network type of Global Acceleration Public Network on the basis of Virtual Private Cloud (VPC) and public network. Compared with the ordinary public network access, Global Acceleration Public Network has significant advantages in terms of delay and stability, and is suitable for scenarios with high demands for data collection, low consumption delay, and reliability. Global Acceleration for Log Service depends on the acceleration environment provided by Alibaba Cloud *Dynamic Route for CDN* products. This function improves overall site performance and user experience by solving problems of slow response, packet loss, and unstable services. These problems are caused by factors such as cross-carriers access, network instability, traffic spikes, and network congestion.

Global Acceleration for Log Service is based on Alibaba Cloud Content Delivery Network (CDN) hardware resources, and optimizes the stability of log collection and data transmission from various forms of data sources such as mobile phones,

Internet of Things (IoT) devices, smart devices, self-built Internet Data Centers (IDCs), and other cloud servers.

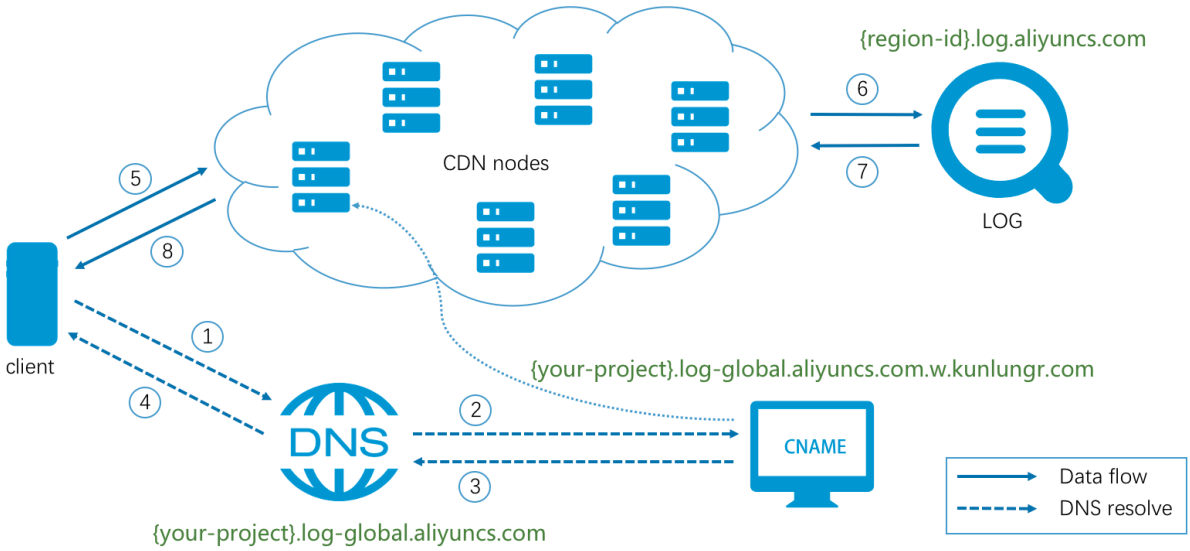
Technical principles

Global Acceleration for Log Service is based on Alibaba Cloud CDN hardware resources. Your global access terminals (such as mobile phones, IOT devices, smart devices, self-built IDCs, and other cloud servers), access the nearest edge node of Alibaba Cloud CDN all over the world and route to Log Service through CDN inner high-speed channels. Compared with common public network transmission, network delay and jitter can be reduced greatly in this method.



The processing flow of Global Acceleration requests for Log Service is shown in the preceding figure. The overall flow is detailed as follows:

1. The client needs to send a domain name resolution request to the public DNS before sending requests of log upload or log download to the Log Service acceleration domain name `your-project.log-global.aliyuncs.com`.
2. The domain name at the public DNS `your-project.log-global.aliyuncs.com` points to the CNAME address `your-project.log-global.aliyuncs.com.w.kunlungr.com`. The domain name resolution is forwarded to the CNAME nodes of Alibaba Cloud CDN.
3. Based on Alibaba Cloud CDN smart scheduling system, CNAME nodes return the IP address of the optimal CDN edge node to the public DNS.
4. The public DNS returns the IP address finally resolved to the client.
5. The client sends a request to the server based on the obtained IP address.
6. After receiving the request, the CDN edge node routes the request to the node closest to the Log Service server based on the dynamic route lookup and private transport protocol. Finally, the request is forwarded to Log Service.
7. After receiving the request from the CDN node, the server of Log Service returns the result of the request to the CDN node.
8. CDN transparently transmits the result or data returned by Log Service to the client .



Billing method

Global Acceleration costs for Log Service include:

- Costs for accessing Log Service

  Costs for accessing Log Service is the same as that in common public network. Log Service supports Pay-As-You-Go billing method, and provides FreeTier quota. For more information, see *Billing method*.

- Service costs for Dynamic Route for CDN

  For information about cloud product costs of Dynamic Route for CDN, see *Billing Method of Dynamic Route for CDN*.

Scenarios

- Advertisement

  Log data about advertising browsing and clicking are extremely important for advertising billing. Advertising carriers include mobile terminal embedding, H5 pages, PC ends, and more all over the world. In some remote areas, the public network data transmission is less stable and risks of log loss exist. A more stable and reliable log upload channel can be obtained through Global Acceleration.

- Online game

  The online game industry has high requirements on the performance and stability of data collection in the official website, logon service, sales service, game service, and other services. The timeliness and stability of data collection are hard to be guaranteed in the case of mobile game data collection and data back transmission

from globalized games. We recommend that you use Global Acceleration for Log Service to solve the preceding issues.

- Finance

  Financial-related applications require high availability and high security for network. Audit logs of each transaction and each user action must be collected securely and reliably to the server. At present, mobile transactions have become mainstream. For example, online banking, credit card malls, mobile securities, and other types of transactions can achieve secure, fast, and stable log collection by using HTTPS Global Acceleration for Log Service.

- Internet of Things

  IoT devices and smart devices (for example, smart speakers and smart watches) collect sensor data, operation logs, critical system logs, and other data to the server for data analysis. These devices are usually distributed all over the world and the surrounding network is not always reliably. To achieve stable and reliable log collection, we recommend using Global Acceleration for Log Service.

Acceleration effect

Region	Delay ms (common public network)	Delay ms (acceleration)	Time-out ratio % (common public network)	Time-out ratio % (acceleration)
Hangzhou	152.881	128.501	0.0	0.0
Europe	1750.738	614.227	0.5908	0.0
USA	736.614	458.340	0.0010	0.0
Singapore	567.287	277.897	0.0024	0.0
Middle East	2849.070	444.523	1.0168	0.0
Australia	1491.864	538.403	0.014	0.0

The test environment is as follows:

- Region of Log service: North China 5 (Hohhot)
- Average upload packet size: 10KB
- Test time range: one day (average)
- Request type: HTTPS
- Request server: Alibaba Cloud ECS (Specification 1C1GB)

 Note:
The acceleration effect is for reference only.

2.2.2 Enable Global Acceleration

To enable Global Acceleration for Log Service, see the following steps.

Prerequisite

- You have enabled Log Service and created the project and Logstore.
- You have enabled *Dynamic Route for CDN*.
- To *Enable HTTPS acceleration*, *Enable HTTP acceleration* first.

Configuration

After HTTP Global Acceleration is enabled for the project, you can also configure Global Acceleration of Logtail, SDK, and other methods according to your needs.

1. *Enable HTTP acceleration*.
2. Enable Global Acceleration of Logtail, SDK, and other methods.
   - HTTPS

     If you use HTTPS to access Log Service, make sure that HTTPS acceleration is enabled. To configure HTTPS acceleration, see *Enable HTTPS acceleration*.
   - Logtail log collection

     When you install Logtail, select the Global Acceleration network type at the page prompt. Then you can obtain global acceleration when you collect logs by using Logtail.
   - SDK, Producer, and Consumer

     Other ways to access Log Service such as SDK, Producer, and Consumer, can be accelerated by replacing the endpoint with `log-global.aliyuncs.com`.

Enable HTTP acceleration

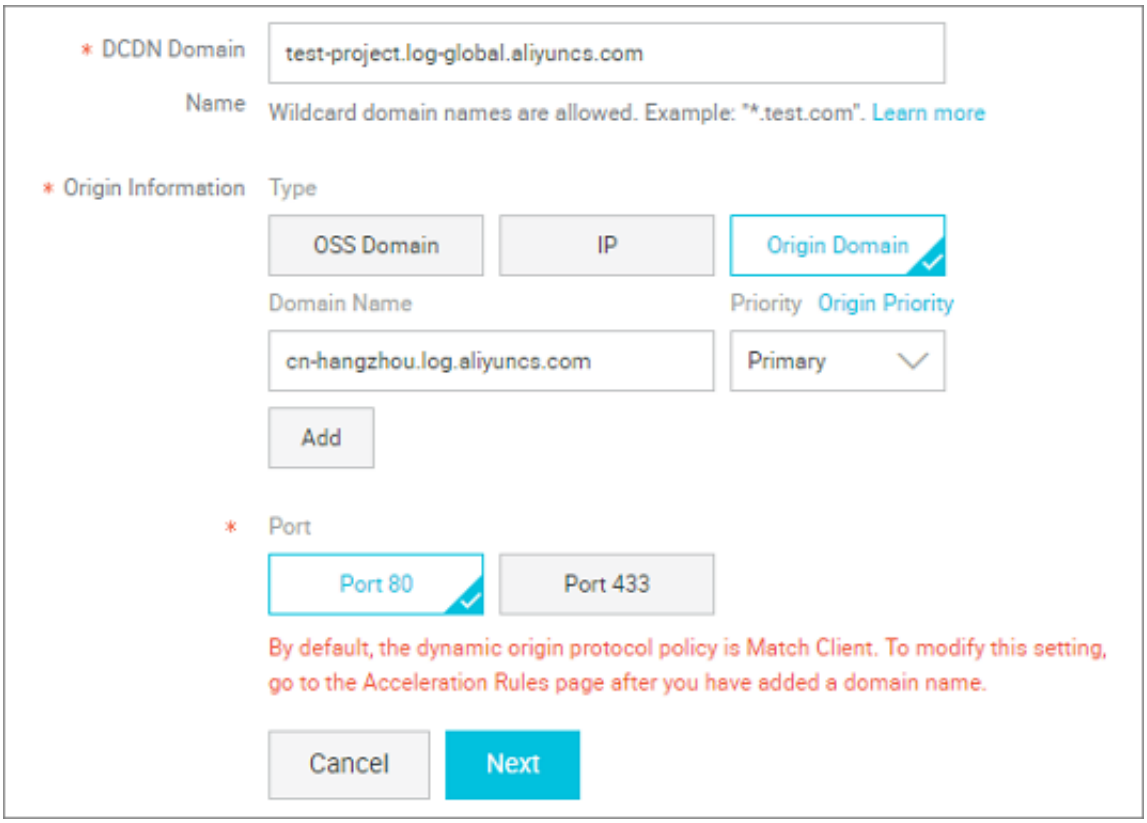
1. Log on to the *Dynamic Route for CDN Console*. Click Domain Names in the left-side navigation pane to enter the Domain Names page.
2. Click Add Domain Name in the upper left corner to enter the Add Domain Name page.

3. Enter the DCDN Domain and other information, and click Next.

Configuration	Description
Accelerated domain name	*project_name*.log-global.aliyuncs.com Replace *project_name* with your project name.
Origin site type	Select Origin Domain .
Domain name	Enter the public network endpoint for the region to which your project belongs. For information about endpoints, see *Service endpoint*.
Port	Please select port 80. If you have an HTTPS acceleration requirement, see Enable HTTPS acceleration.

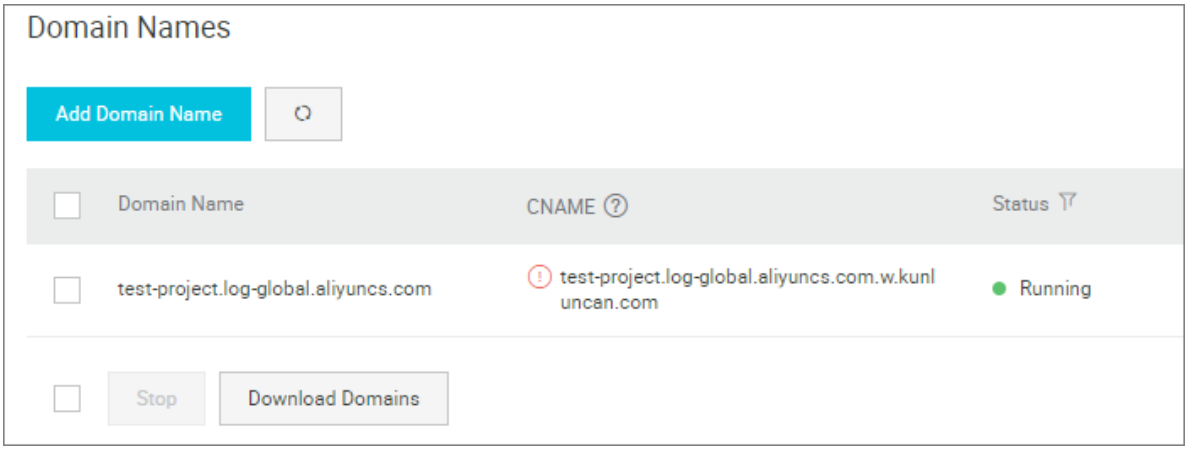
Configuration	Description
Accelerated area	By default, this configuration item is not displayed and the acceleration area is `Domesticat e accelerati on .` If you have a demand for Global Acceleration, open a ticket for the Dynamic Route for CDN product to apply for a whitelist. After your application is approved, you can select an acceleration region based on your needs.

For more information about adding domain names, see 8.

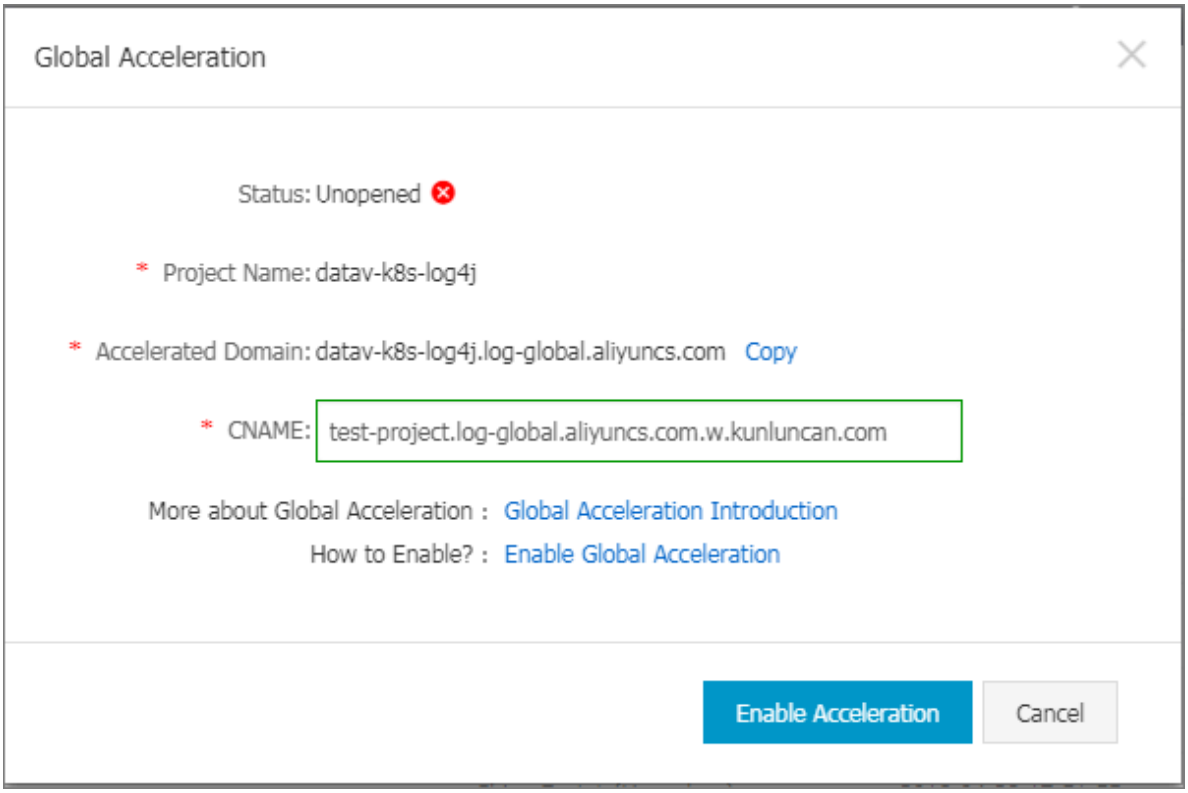


4. Go to the Domain management page as prompted.

   You can view the CNAME of each corresponding domain name in the Domain name management page.



5. Log on to the Log Service console and click Global Acceleration at the right of a specified project in the Project list.
6. Enter the CNAME corresponding to the accelerated domain name in the dialog box. Click Enable acceleration.



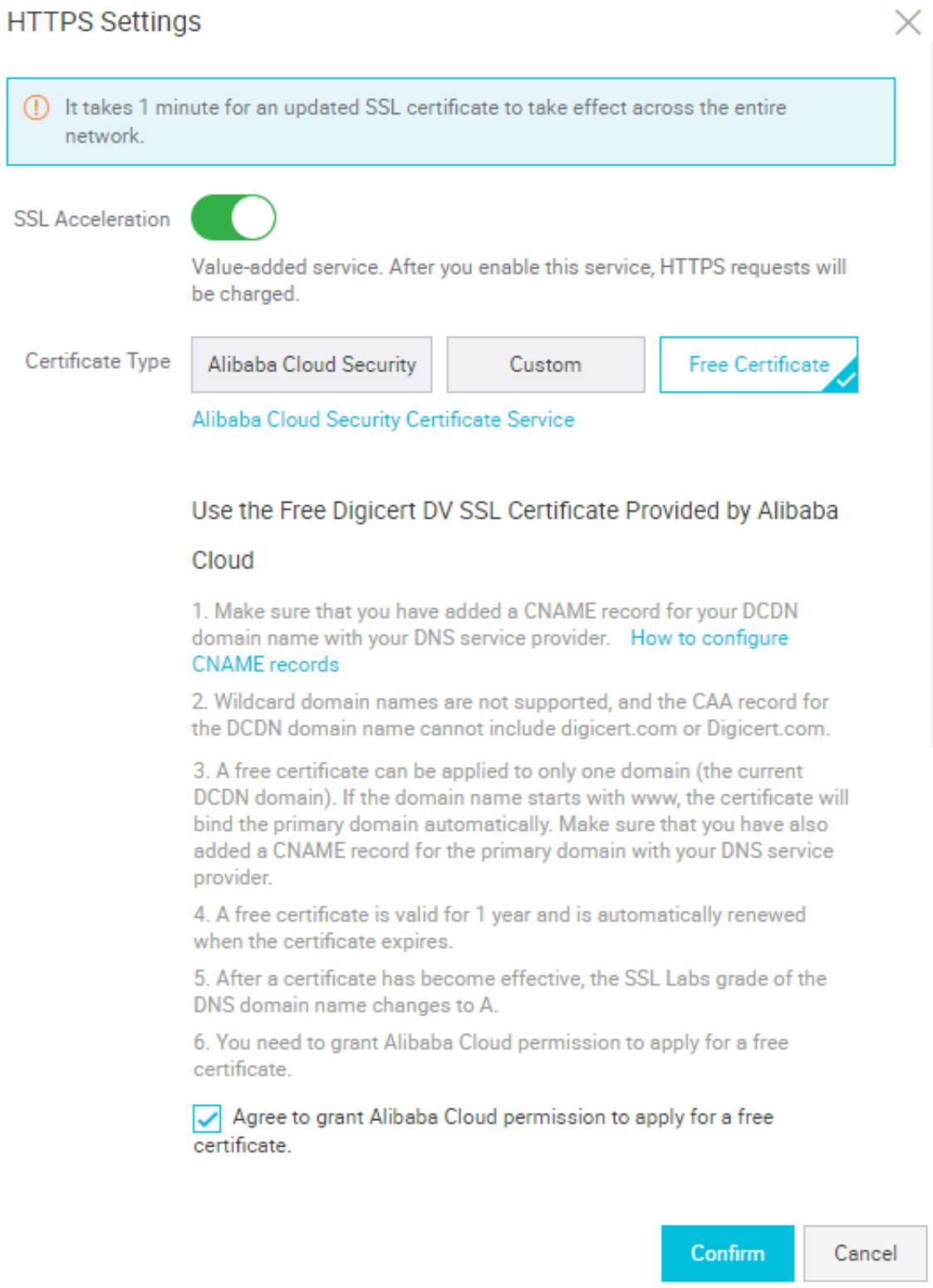
   After you complete the preceding steps, Global Acceleration for Log Service is enabled.

Enable HTTPS acceleration

After enabling HTTP acceleration, if you have HTTPS access requirements, you can use the following steps to enable HTTPS acceleration.

1. Log on to the *Dynamic Route for CDN Console*. Click Domain Names in the left-side navigation pane to enter the Domain Names page.
2. Click Configure to the right of a specified domain name.
3. Click HTTPS Settings in the left-side navigation pane and click Modify in the column of SSL Certificate to enter the HTTPS Settings page.

4. Configure SSL Acceleration and Certificate Type.
   - Enable SSL Acceleration.
   - Select Free Certificate for Certificate Type.



After the configuration is completed, select Agree to grant Alibaba Cloud permission to apply for a free certificate., and click Confirm.

Verify if the acceleration configuration takes effect

FAQ

- How to verify if the acceleration configuration takes effect?

  After the configuration is completed, you can verify if the acceleration takes effect by accessing your accelerated domain name.

  For example, if Global Acceleration is enabled for the `test-project` project, you can use `curl` to send a request to the accelerated domain name. If the following type of output is returned, the acceleration takes effect.

```
$ curl   test-project.log-global.aliyuncs.com
{"Error":{"Code":"OLSInvalid","Method","Message":"The
 script   name   is   invalid  : /","RequestId":"5B55386A2C
 E41D1F4FBC  F7E7"}}
```

- How to handle the error of `project not exist` reported in accessing accelerated domain name?

  This problem is caused usually by an invalid source site address. Log on to the Dynamic Route for CDN console and change the source site address to the public network address of the region to which your project belongs. For information about address list, see *Service endpoint*.

  Note:
  Changing the source site address has a synchronization delay of several minutes.

2.2.3 Configure Logtail collection acceleration

After global acceleration is enabled, the Logtail that is installed in global acceleration mode automatically collects logs in global acceleration mode. For the Logtail that is installed before global acceleration is enabled, you need to manually switch the acceleration mode to global acceleration by performing the steps in this topic.

Prerequisites

1. *Enable HTTP acceleration*.

2. (Optional) *Enable HTTP acceleration*.

   If you use HTTPS to access Log Service, make sure that HTTPS acceleration has been enabled and that you have configured HTTPS acceleration by following the instructions provided in *Enable HTTP acceleration*.

3. Make sure that acceleration functions normally

   by following the instructions provided in *Enable Global Acceleration*.

Before you begin

Before you configure Logtail collection acceleration, note that:

- If the Logtail is installed after global acceleration enabling, you must set the installation mode to global acceleration by following the instructions provided in *Install Logtail in Linux*. Then, the Logtail collects logs using global acceleration mode methods.
- If the Logtail is installed before global acceleration is enabled, you must switch the Logtail collection mode to global acceleration by performing the steps in this topic.

Switch the Logtail collection mode to global acceleration.

1. Stop the Logtail.
   - In Linux, run `/ etc / init . d / ilogtaild  stop` as the admin user.
   - In Windows:
     a. In Control Panel, choose System and Security > Administrative Tools.
     b. Open the Services program and locate the LogtailWorker file.
     c. Right-click the file and click Stop in the shortcut menu.
2. Modify the Logtail startup configuration file *ilogtail_c onfig . json .*

   Change the endpoint in `data_serve r_list` to `log - global . aliyuncs . com` by following the instructions provided in *Startup configuration file (ilogtail_config.json)*.

3. Start the Logtail.
   - In Linux, run `/etc/init.d/ilogtaild start` as the admin user.
   - In Windows:
     a. In Control Panel, choose System and Security > Administrative Tools.
     b. Open the Services program and locate the LogtailWorker file.
     c. Right-click the file and click Start in the shortcut menu.

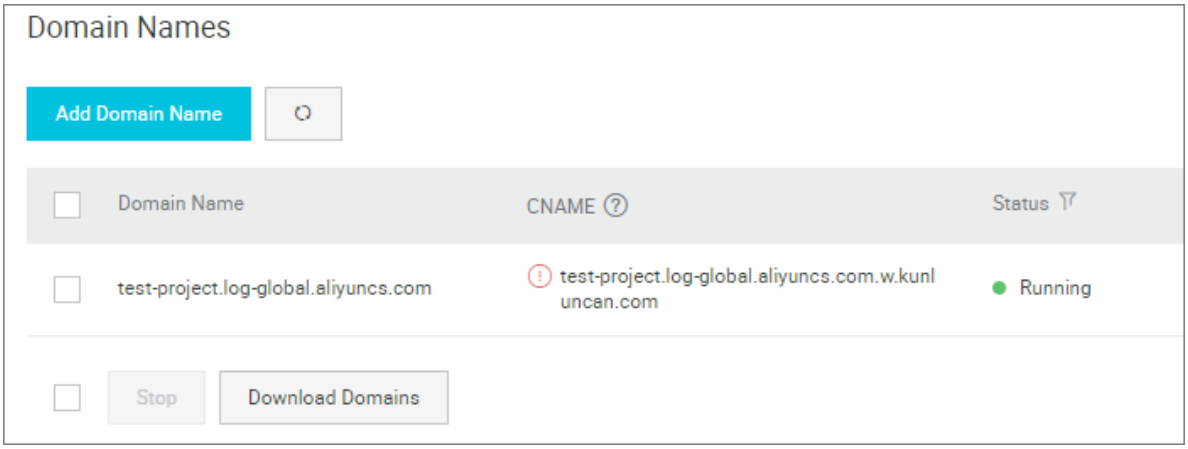
2.2.4 Disable Global Acceleration

To disable Global Acceleration for Log Service, perform the following operations.

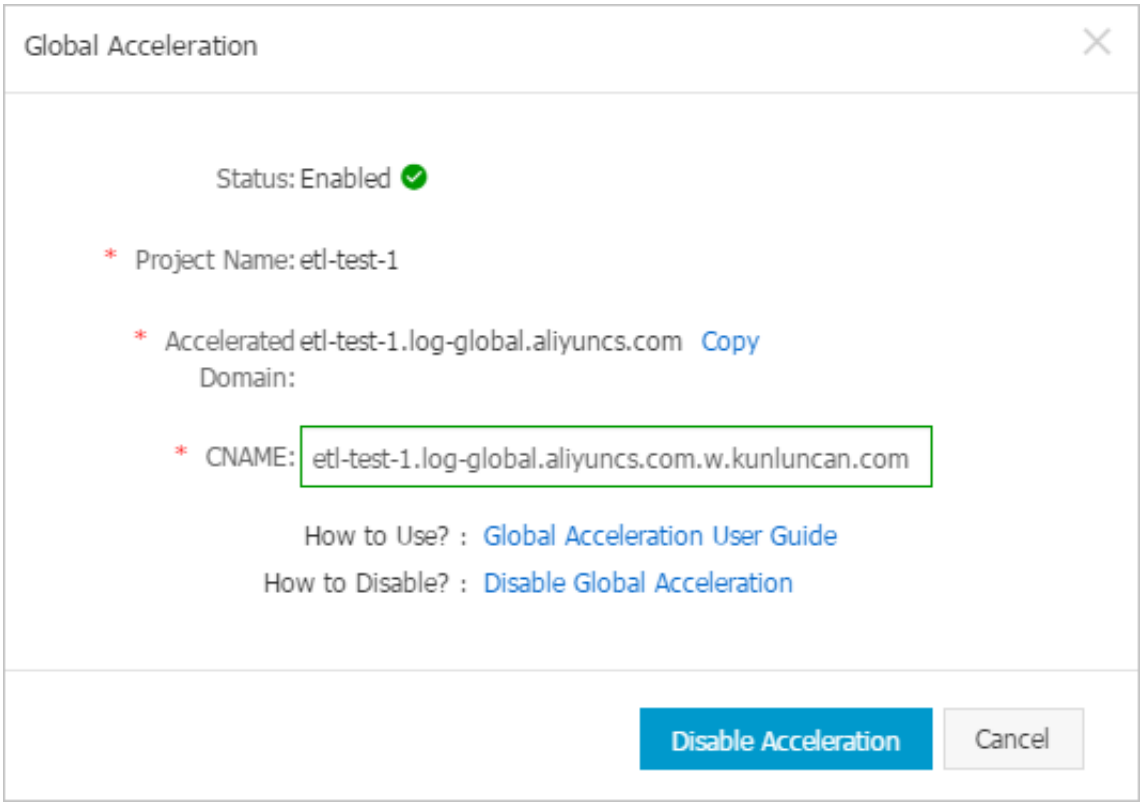
> **Note:**
> When you disable Global Acceleration, the accelerated domain name configured during provisioning becomes unavailable. Make sure that all of your clients do not upload or request data through the domain name before you disable Global Acceleration.

Disable Global Acceleration

1. Log on to the *Dynamic Route for CDN Console*. Click Domain name management in the left-side navigation pane to enter the Domain name management page.
2. View the CNAME corresponding to the domain name that is to be disabled .



3. Log on to the Log Service console. On the Project list page, click Global Acceleration at the right of a specified project.
4. Enter CNAME and click Disable acceleration.


Global Acceleration
Status: Enabled
* Project Name: etl-test-1
* Accelerated Domain: etl-test-1.log-global.aliyuncs.com Copy
* CNAME: etl-test-1.log-global.aliyuncs.com.w.kunluncan.com
How to Use? : Global Acceleration User Guide
How to Disable? : Disable Global Acceleration
Disable Acceleration
Cancel

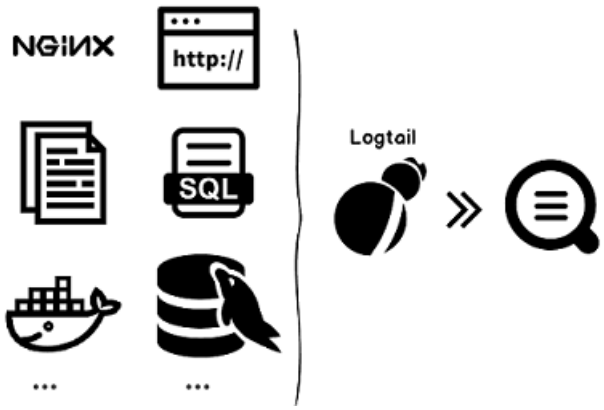
3 Logtail collection

3.1 Overview

3.1.1 Overview

The Logtail access service is a log collection agent provided by Log Service. You can use Logtail to collect logs from servers such as Alibaba Cloud Elastic Compute Service (ECS) instances in real time in the Log Service console.

Figure 3-1: Function advantages



Benefits

- Non-invasive log collection based on log files. You do not have to modify codes of any application, and log collection does not affect the operating logic of your applications.
- In addition to text log collection, more collection methods are supported, such as binlog, http, and container stdout.
- Containers are well supported. This service supports data collection in standard containers, swarm clusters, and Kubernetes clusters.
- Logtail handles exceptions occurred in the log collection process. When problems (such as the network or Log Service is abnormal, and the user data temporarily exceeds the reserved bandwidth writing limit) occur, Logtail actively retries and caches data locally to guarantee the data security.
- Centralized management capability based on Log Service. After installing Logtail, you can configure settings such as the machines from which logs are to be

collected and the collection method in Log Service in a centralized way, without logging on to the servers and configuring settings separately. For how to install Logtail, see *Install Logtail in Windows* and *Install Logtail in Linux*.

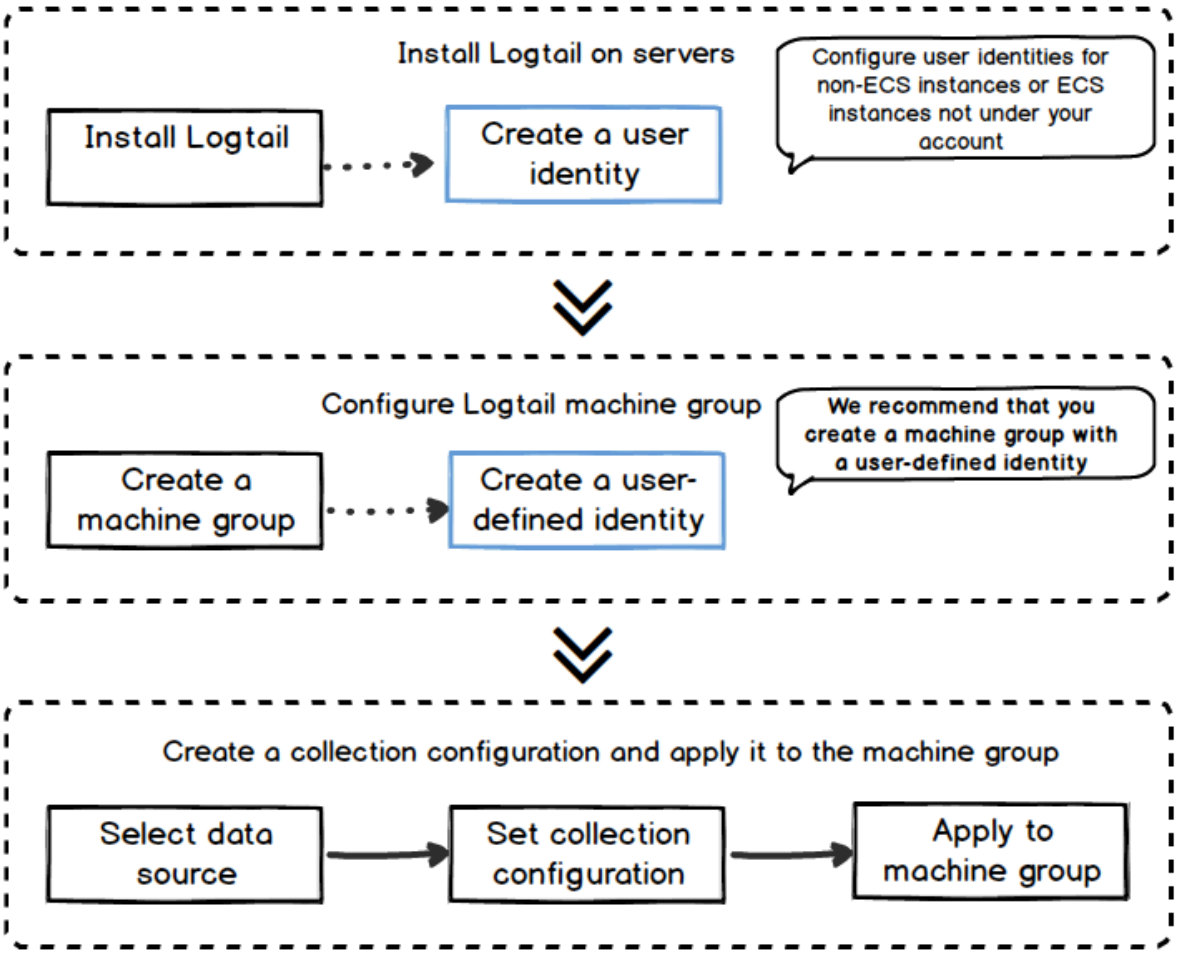
- Comprehensive self-protection mechanism. To make sure that the collection agent running on your machine does not significantly affect the performance of your services, the Logtail client strictly protects and limits the usage of CPU, memory, and network resources.

Processing capabilities and limits

See *Limits*.

Procedure

Figure 3-2: Configuration process



Follow these steps to use Logtail to collect logs from servers:

1. Install Logtail. Install Logtail on the servers from which logs are to be collected. For more information, see *Install Logtail in Windows* and *Install Logtail in Linux*
2. *Create an ID to identify a machine group*. Skip this step if you are about to collect logs from Alibaba Cloud ECS instances
3. *Create a machine group with an IP address as its identifier*. Log Service manages all the servers from which logs are to be collected by using the Logtail client in the form of machine groups. Log Service allows you to define machine groups by using IP addresses or custom identifiers. You can create a machine group as instructed when applying Logtail configurations to machine groups.
4. Create a Logtail collection configuration and apply it to the machine group. You can collect data such as *Collect text logs* and *Syslog* by creating a Logtail configuration in the data import wizard. Then, you can apply the Logtail configuration to the machine group.

After completing the preceding steps, incremental logs on servers from which logs are to be collected are actively collected and sent to the corresponding Logstore. Historical logs are not collected. You can query these logs in the console or by using APIs/SDKs. You can also query the Logtail log collection status in the console, such as check whether the collection is normal or if any error occurs.

For the complete procedure for Logtail access service in the Log Service console, see *Collect text logs* .

Container

- Alibaba Cloud Container Service Swarm cluster: see *Enable Log Service*.
- Alibaba Cloud Container Service Kubernetes cluster: see *Kubernetes log collection process*.
- Self-built Kubernetes: see*Self-built Kubernetes installation*
- Other self-built Docker clusters: see*Collect standard Docker logs*

Major concepts

- Machine group: A machine group contains one or more machines from which a type of logs is to be collected. By applying a Logtail configuration to a machine group, Log Service collects logs from all the machines in the machine group according to the same Logtail configuration. You can also manage a machine group in the Log Service console, such as creating/deleting a machine group, and

adding/removing a machine to/from a machine group. You must note that a single machine group cannot contain a mix of Windows and Linux machines, but may have machines with different versions of Windows Server or different release versions of Linux.

- Logtail client: Logtail is the agent that collects logs and runs on servers from which logs are to be collected. For how to install Logtail, see *Install Logtail in Windows* and *Install Logtail in Linux*. After installing Logtail on the server, create a Logtail configuration and then apply it to a machine group.
  - In Linux, Logtail is installed in the `/ usr / local / ilogtail` directory and starts two independent processes (a collection process and a daemon process) whose names start with ilogtail. The program running log is `/ usr / local / ilogtail / ilogtail . LOG` .
  - In Windows, Logtail is installed in the `C :\ Program Files \ Alibaba \ Logtail` directory (for 32-bit system) or the `C :\ Program Files ( x86 )\ Alibaba \ Logtail` directory (for 64-bit system). Navigate to Windows Administrative Tools > Services, you can view two Windows services: LogtailWorker and LogtailDaemon. LogtailWorker is used to collect logs and LogtailDaemon works as a daemon. The program running log is `logtail_ *. log` in the installation directory.
- Logtail configuration: Logtail configuration is a collection of policies to collect logs by using Logtail. By configuring Logtail parameters such as data source and collection mode, you can customize the log collection policy for all the machines in the machine group. A Logtail configuration is used to collect a type of logs from machines, parse the collected logs, and send them to a specified Logstore of Log Service. You can add a Logtail configuration for each Logstore in the console to enable the Logstore to receive logs collected by using this Logtail configuration.

Basic functions

The Logtail access service provides the following functions:

- Real-time log collection: Logtail dynamically monitors log files, and reads and parses incremental logs in real time. Generally, a delay of less than three seconds

exists between the time when a log is generated and the time when a log is sent to Log Service.

> Note:
> Logtail does not support collection of historical data. Logs with an interval of more than 12 hours between the time when a log is read and the time when a log is generated are discarded.

- Automatic log rotation processing: Many applications rotate log files according to the file size or date. During the rotation process, the original log file is renamed and a new blank log file is created for log writing. For example, the monitored `app.LOG` is rotated to generate `app.LOG.1` and `app.LOG.2`. You can specify the file to which collected logs are written, for example, `app.LOG`. Logtail automatically detects the log rotation process and guarantees that no log data is lost during this process.
- Multiple collection input sources: Besides text logs, Logtail supports the input sources such as syslog, HTTP, MySQL, and binlog. For more information, see Data Source in Log Service user guide.
- Compatible with open-source collection agent: Input source of Logtail can be data collected by open-source softwares , such as Logstash and Beats. For more information, see Data Source in Log Service user guide.
- Automatic handling of collection exceptionsWhen data transmission fails because of exceptions such as Log Service errors, network measures, and quota exceeding the limit, Logtail actively retries based on specific scenario. If the retry fails, Logtail writes the data to the local cache and then automatically resends the data later.
- Flexible collection policy configuration: You can use Logtail configuration to flexibly specify how logs are collected from a server. Specifically, you can select log directories and files, which support exact match or fuzzy match with wildcards, based on actual scenarios. You can customize the extraction method for log collection and the names of extracted fields. Log Service supports extracting logs by using regular expressions. The log data models of Log Service require that each log must have a precise timestamp. Therefore, Logtail provides custom log time formats, allowing you to extract the required timestamp information from log data of different formats.

- Automatic synchronization of collection configuration: Generally, after you create or update a configuration in the Log Service console, Logtail automatically accepts and brings the configuration into effect within three minutes. No collected data is lost when configuration is being updated.
- Automatic upgrade of client: After you manually install Logtail on a server, Log Service automatically performs the Operation & Maintenance (O&M) and upgrade of Logtail. No log data is lost when Logtail is being upgraded.
- Status monitoring: To prevent the Logtail client from consuming too many resources and thus affecting your services, the Logtail client monitors its consumption of CPU and memory in real time. The Logtail client is automatically restarted when its resource usage exceeds the limit to avoid affecting other operations on the machine. The Logtail client actively limits network traffic to avoid excessive bandwidth consumption.
- Data transmission with a signature: To prevent data tampering during the transmission process, the Logtail client obtains your Alibaba Cloud AccessKey (AK) and provides a signature to all log data packets to be sent.

 Note:
To maintain security of your Alibaba Cloud AK, the Logtail client uses the HTTPS tunnel to obtain your AK.

Data collection reliability

During data collection, Logtail stores the collected checkpoint information to your local directory on a regular basis. If any exception occurs (such as the server unexpectedly shuts down or a process crashes), Logtail will collect data from the last recorded checkpoint after being restarted to prevent data loss. Then, Logtail functions according to the *resource limits* listed in the configuration file. However, if a resource is occupied for five minutes more than the preset time period, Logtail will be forcibly restarted. After the force restart, some of the existing data may be regenerated.

Although Logtail uses multiple methods to guarantee data collection reliability, absolute data integrity cannot be guaranteed. Specifically, data loss may occur due to the following reasons:

- Logtail is not functioning, but logs have been rotated several times.
- The log rotation speed is exceedingly high, for example, one time per second.

- The log collection speed is slower than the log generation speed for a long period of time.

3.1.2 Logtail collection process

The Logtail client performs the following six steps to collect logs from your server: monitor files, read files, process logs, filter files, aggregate logs, and send logs.

After you install the Logtail client on your server and configure a Logtail Config, Logtail starts collecting logs to Log Service. The log collection process involves the following steps:

1. *Monitor files*
2. *Read files*
3. *Process logs*
4. *Filter logs*
5. *Aggregate logs*
6. *Send logs*



Note:

After a Logtail Config is configured for a machine group, unmodified logs on a server in the machine group will be regarded as historical files. However, Logtail does not collect historical files. If you want to collect historical logs, see *Import history logs*.

Monitor files

After you install the Logtail client on your server and configure a Logtail Config based on data sources, the Logtail Config sends logs to Logtail in real time. Then, Logtail uses the Logtail Config to monitor files.

1. Specifically, Logtail scans the log directories and files that conform to the specified file naming conventions layer by layer according to the configured log path and maximum monitoring directory depth.

   To ensure the efficiency and stability of log collection, Logtail registers event monitoring for the collection directory (namely, the *Inotify* directory on Linux or the *ReadDirectoryChangesW* directory on Windows) and performs periodic polling.
2. If the monitoring results show that unmodified log files that conform to the file naming conventions exist in the specified directory, Logtail will not collect the files

. If there are modified log files, a collection process will be triggered and Logtail will read the files.

Read files

Logtail starts to read the modified files.

1. Logtail checks the size of a file when reading the file for the first time.
   - If the file size is smaller than 1 MB, Logtail reads the file from the beginning.
   - If the file size is larger than 1 MB, Logtail reads the last 1-MB content of the file.
2. If Logtail has read the file before, Logtail reads the file from the last checkpoint.
3. Logtail can read up to 512 KB at a time. Therefore, you need to limit the log size to 512 KB.

> Note:
> If you have modified the time on your server, you need to manually restart Logtail. Otherwise, the log generation time will be incorrect and some logs may be mistakenly discarded.

Process logs

Logtail splits a log into lines, parses the log, and confirms the correctness of the time field settings.

1. Line splitting:

   If a line start regular expression has been specified in the Logtail Config, Logtail will split the log into lines according to the line start settings. In this case, Logtail processes the lines as multiple logs. If no line start regular expression has been specified, Logtail regards a data block as a log and processes it.
2. Parsing:

   Logtail uses the Logtail Config to parse the log content based on specified rules, such as regular expressions, delimiters, and JSON arrays.

   > Note:
   > An excessively complex regular expression may lead to an abnormally high CPU usage. Therefore, we recommend that you use an efficient regular expression.

3. Parsing failure handling:

   Depending on whether the *discarding logs with parsing failure* function is enabled in the Logtail Config, you can handle logs with parsing failure as follows:

   - If the function is enabled, Logtail discards the log and reports a corresponding error.
   - If the function is disabled, you need to upload the original log with its key of raw_log and Value of the log content.

4. Time field settings:

   - If the time field is not set, the log generation time is the current parsing time.
   - If the time field is set and the log generation time is:
     - Less than 12 hours from the current time, Logtail extracts the time from the parsed time field.
     - More than 12 hours from the current time, Logtail discards the log and reports a corresponding error.

Filter logs

Logtail filters logs according to the *filter settings* in the Logtail Config.

- If the filter is not set, Logtail will not filter logs but directly aggregates logs.
- If the filter is set, Logtail will traverse and verify all fields in each log.
  - Logtail collects logs that conform to filter settings, that is, all fields in filter settings can be found in the log and all the fields conform to the setting requirements.
  - Logtail does not collect logs that do not conform to filter settings.

Aggregate logs

Logtail sends log data to Log Service. To reduce the number of network requests, Logtail caches the logs for some time. Then, Logtail aggregates and packages the logs to send them to Log Service.

During caching, Logtail will immediately package logs and send them if any of the following conditions is met:

- Log aggregation lasts more than 3s.
- There are more than 4.096 logs to be aggregated.
- The target log size exceeds 512 KB.

Send logs

Logtail sends the aggregated log to Log Service. You can set the startup parameters `max_bytes_ per_sec` and `send_reque st_concurr ency` by following the instructions provided in *Configure startup parameters* to adjust the log sending rate and the maximum number of logs that can be concurrently sent. In this case, Logtail ensures that the preset values are not exceeded.

If the log sending fails, Logtail automatically retries or quits the task according to the corresponding error message.

Error message	Description	Handling method
Error code: 401	The Logtail client does not have the permission to collect data.	Logtail discards the log package.
Error code: 404	The project or Logstore specified in the Logtail Config does not exist.	Logtail discards the log package.
Error code: 403	The Shard quota exceeds the upper limit.	Wait for 3s and try again.
Error code: 500	An error occurs on the server.	Wait for 3s and try again.
Network expiration	A network connection error occurs.	Wait for 3s and try again.

3.1.3 Logtail configuration and recording files

The running of Logtail depends on a series of configuration files, which generates specific information recording files. This topic describes the basic information and paths of commonly generated files.

Configuration files:

- *Startup configuration file (ilogtail_config.json)*
- *AliUid configuration file*
- *User-defined identity file (user_defined_id)*
- *Logtail Config file (user_log_config.json)*

Recording files:

- *AppInfo recording file (app_info.json)*
- *Logtail operational log file (ilogtail.LOG)*
- *Logtail plug-in log file (logtail_plugin.LOG)*

- *Container path mapping file (docker_path_config.json)*

Startup configuration file (ilogtail_config.json)

The file is used to view or set Logtail running parameters. The file is in JSON format.

After installing Logtail, you can use the file to:

- Modify Logtail running parameters.

  You can modify common settings, such as the CPU usage threshold and resident memory usage threshold by modifying the file.
- Check whether installation commands are correct.

  In the file, `config_ser ver_addres s` and `data_serve r_list` are determined by parameters and commands used during installation. If the region specified by the parameters is different from the region where Log Service resides or the address is inaccessible, incorrect parameters or commands are used during installation. In this case, Logtail cannot collect logs, and you need to reinstall it.

 Note:

- The file must be valid JSON arrays. Otherwise, Logtail cannot be started.
- The modified file can take effect only after Logtail is restarted.

The following table lists default configuration items. For details about other configuration items, see *Configure startup parameters*.

Table 3-1: Default configuration items in the startup configuration file

Configuration item	Description
config_server_address	Address of the configuration file Logtail obtains from your server. The address is determined by the parameters and commands you use during installation. The address must be accessible, and the region specified by the parameters must be the same as the region where Log Service resides.

Configuration item	Description
data_server_list	Address of the data server, which is determined by the parameters and commands you use during installation The address must be accessible, and the region specified by the parameters must be the same as the region where Log Service resides.
cluster	Region name
endpoint	*Service endpoint*
cpu_usage_limit	CPU usage threshold, which is calculated by core
mem_usage_limit	Resident memory usage threshold
max_bytes_per_sec	Maximum amount of raw data Logtail can send. The amount will not be limited if the data sending rate exceeds 20 Mbit/s.
process_thread_count	Number of threads Logtail uses to write data to log files
send_request_concurrency	Number of data packets Logtail can send concurrently and asynchronously. By default, Logtail sends data packets asynchronously. You can set the configuration item to a larger value if the write TPS is excessively high.

File address:

- Linux: */ usr / local / ilogtail / ilogtail_c onfig . json*
- Container: The file is stored in the Logtail container, and the file address is configured through the environment variable `ALIYUN_LOG TAIL_CONFI G`. You can view the address through `Docker inspect $ { logtail_co ntainer_na me } | grep ALIYUN_LOG TAIL_CONFI G`, for example, */ Etc / ilogtail / CONF / CN - Hangzhou / FIG .*
- Windows:
  - x64: *C :\ Program Files (x86)\ Alibaba \ Logtail \ ilogtail_c onfig . json*
  - x32: *C :\ Program Files \ Alibaba \ Logtail \ ilogtail_c onfig . json*

File example:

```
$ cat  / usr / local / ilogtail / ilogtail_c  onfig . json
{
    " config_ser  ver_addres  s " : " http :// logtail . cn - hangzhou
 - intranet . log . aliyuncs . com ",
    " data_serve  r_list " :
    [
        {
            " cluster " : " ap - southeast - 2 ",
            " endpoint " : " cn - hangzhou - intranet . log . aliyuncs
 . com "
        }
    ],
    " cpu_usage_  limit " :  0 . 4 ,
    " mem_usage_  limit " :  100 ,
    " max_bytes_  per_sec " :  2097152 ,
    " process_th  read_count " :  1 ,
    " send_reque  st_concurr  ency " :  4 ,
    " streamlog_  open " :  false
}
```

AliUid configuration file

The file contains the AliUid of your Alibaba Cloud account. AliUid is used to indicate that your Alibaba Cloud account has the permissions to access your server and collect logs. You need to manually create the AliUid configuration file when collecting logs from an ECS instance that does belong to your Alibaba Cloud account or from on-premises IDCs. For more information, see *Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs*.



Note:

- This file is optional and is used only when you collect logs from an ECS instance that does belong to your Alibaba Cloud account or from on-premises IDCs.
- The file can only contain the AliUid of your Alibaba Cloud account. It cannot contain the AliUid of any RAM user account under your Alibaba Cloud account.
- The file name cannot contain any suffix.
- Logtail can be configured with multiple AliUid configuration files, but a Logtail container can be configured with only one AliUid configuration file.

File address

- Linux: `/ etc / ilogtail / users /`
- Container: The file is directly configured through the environment variable `ALIYUN_LOG  TAIL_USER_  ID` in the Logtail container. You can view the file

through `docker inspect ${logtail_container_name} | grep ALIYUN_LOG_TAIL_USER_ID`.

- Windows: `C:\LogtailData\users\`

File example

```
$ ls /etc/ilogtail/users/
1559122535 02 **** 1329232535 02 ****
```

User-defined identity file (user_defined_id)

The file is used to configure machine groups with custom identifiers. For more information, see *Create an ID to identify a machine group*.

Note:

- This file is optional and is used only when configuring machine groups with custom identifiers.
- If multiple custom identifiers are configured for a machine group, they must be separated by delimiters.

File address

- Linux: */etc/ilogtail/user_defined_id*
- Container: The file is directly configured through the environment variable `ALIYUN_LOG_TAIL_USER_DEFINED_ID` in the Logtail container. You can view the file through *docker inspect ${logtail_container_name} | grep ALIYUN_LOG_TAIL_USER_DEFINED_ID*.
- Windows: *C:\LogtailData\user_defined_id*

File example

```
$ cat /etc/ilogtail/user_defined_id
aliyun-ecs-rs1e16355
```

Logtail Config file (user_log_config.json)

The file contains Logtail Config information Logtail obtains from your server. The file is in JSON format and is updated with Logtail Config updates. The file is used to check whether Logtail Config sends logs to your server. If the file exists and the file content is up-to-date, the Logtail Config has sent logs.

Note:

- **We recommend that you do not modify the file unless you need to manually configure keys and modify database passwords.**
- **The file must be uploaded when you open a ticket.**

File address

- **Linux:** */ usr / local / ilogtail / user_log_c onfig . json*
- **Container:** */ usr / local / ilogtail / user_log_c onfig . json*
- **Windows**
  - **x64:** *C :\ Program Files (x86)\ Alibaba \ Logtail \ user_log_c onfig . json*
  - **x32:** *C :\ Program Files \ Alibaba \ Logtail \ user_log_c onfig . json*

File example

```
$ cat  / usr / local / ilogtail / user_log_c  onfig . json
{
   " metrics " : {
      "## 1 . 0 ## k8s - log - c12ba2028 ***** 939f0b $ app - java " :
 {
         " aliuid " : " 16542189 ***** 50 ",
         " category " : " app - java ",
         " create_tim  e " :  1534739165 ,
         " defaultEnd  point " : " cn - hangzhou - intranet . log .
 aliyuncs . com ",
         " delay_alar  m_bytes " :  0 ,
         " enable " :  true ,
         " enable_tag " :  true ,
         " filter_key  s " : [],
         " filter_reg  s " : [],
         " group_topi  c " : "",
         " local_stor  age " :  true ,
         " log_type " : " plugin ",
         " log_tz " : "",
         " max_send_r  ate " : - 1 ,
         " merge_type " : " topic ",
         " plugin " : {
            " inputs " : [
               {
                  " detail " : {
                     " IncludeEnv " : {
                        " aliyun_log  s_app - java " : " stdout "
                     },
                     " IncludeLab  le " : {
                        " io . kubernetes . container . name " : "
 java - log - demo - 2 ",
                        " io . kubernetes . pod . namespace " : "
 default "
                     },
                     " Stderr " :  true ,
                     " Stdout " :  true
                  },
                  " type " : " service_do  cker_stdou  t "
```

```
                }
             ]
          },
          " priority " :  0 ,
          " project_na  me " : " k8s - log - c12ba2028c *****
 ac1286939f  0b ",
          " raw_log " :  false ,
          " region " : " cn - hangzhou ",
          " send_rate_  expire " :  0 ,
          " sensitive_  keys " : [],
          " tz_adjust " :  false ,
          " version " :  1
       }
    }
}
```

AppInfo recording file (app_info.json)

The file contains various time information, such as the Logtail startup time and the time when Logtail obtains the IP address and host name. The IP address is needed when you configure *machine groups with IP addresses as identifiers*.

In normal cases, Logtail obtains the server IP address according to the following rules :

- Logtail automatically obtains the IP address if the IP address has been attached to your host through the server file `/ etc / hosts .`
- Logtail automatically obtains the IP address of the first NIC on your host if no IP address is attached to your host.

 Note:

- The file only contains internal information about Logtail. Manual modifications to the file content do not change basic Logtail information.
- If you have modified network configurations of your server, for example, host name, you need to restart Logtail to obtain the new IP address.

Table 3-2: Field description

Field	Description
UUID	Server serial number
hostname	Host name
instance_id	Randomly generated identifier for uniquely indicating Logtail

Field	Description
ip	IP address obtained by Logtail. An empty field indicates that Logtail does not obtain the IP address and cannot function normally. In this case, you need to set an IP address for your server and restart Logtail. **Note:** If the target machine group uses an IP address as an identifier, the IP address configured in the machine group must be the same as the one specified by this field. If an incorrect IP address is configured on your server, you need to modify the IP address within the machine group, wait one minute, and then check again.
logtail_version	Version of the Logtail client
os	OS version
update_time	Time when Logtail is last started

File address

- **Linux:** `/ usr / local / ilogtail / app_info . json`
- **Container:** `/ usr / local / ilogtail / app_info . json`
- **Windows**
  - **x64:** `C :\ Program Files ( x86 )\ Alibaba \ Logtail \ app_info . json`
  - **x32:** `C :\ Program Files \ Alibaba \ Logtail \ app_info . json`

File example

```
$ cat  / usr / local / ilogtail / app_info . json
{
   " UUID " : "",
   " hostname " : " logtail - ds - slpn8 ",
   " instance_i  d " : " E5F93BC6 - B024 - 11E8 - 8831 - 0A58AC1403
 9E_172 . 20 . 3 . 158_153605  3315 ",
   " ip " : " 172 . 20 . 3 . 158 ",
   " logtail_ve  rsion " : " 0 . 16 . 13 ",
   " os " : " Linux ;  3 . 10 . 0 - 693 . 2 . 2 . el7 . x86_64 ; # 1
 SMP   Tue   Sep   12   22 : 26 : 13   UTC   2017 ;  x86_64 ",
   " update_tim  e " : " 2018 - 09 - 04   09 : 28 : 36 "
```

```
}
```

Logtail operational log file (ilogtail.LOG)

The file contains running information about the Logtail client. Log levels are ranked as follows in ascending order: `INFO` , `WARN` , `ERROR` . `INFO` -type logs can be ignored.



Note:

- First, you need to *diagnose collection exceptions* and troubleshoot errors according to specific error types and Logtail operational logs.
- The file must be uploaded when you open a ticket due to Logtail collection exceptions.

File address

- **Linux:** */ usr / local / ilogtail / ilogtail . LOG*
- **Container:** */ usr / local / ilogtail / ilogtail . LOG*
- **Windows**
  - **x64:** *C :\ Program Files (x86)\ Alibaba \ Logtail \ logtail_ *. log*
  - **x32:** *C :\ Program Files \ Alibaba \ Logtail \ logtail_ *. log*

File example

```
$ tail  / usr / local / ilogtail / ilogtail . LOG
[ 2018 - 09 - 13   01 : 13 : 59 . 024679 ]    [ INFO ]    [ 3155 ]
   [ build / release64 / sls / ilogtail / elogtail . cpp : 123 ]
 change   working   dir :/ usr / local / ilogtail /
[ 2018 - 09 - 13   01 : 13 : 59 . 025443 ]    [ INFO ]    [ 3155 ]
  [ build / release64 / sls / ilogtail / AppConfig . cpp : 175 ]
 load   logtail   config   file ,  path :/ etc / ilogtail / conf / ap
 - southeast - 2 / ilogtail_c  onfig . json
[ 2018 - 09 - 13   01 : 13 : 59 . 025460 ]    [ INFO ]    [ 3155 ]
  [ build / release64 / sls / ilogtail / AppConfig . cpp : 176 ]
 load   logtail   config   file ,  detail :{
   " config_ser  ver_addres  s " : " http :// logtail . ap - southeast
 - 2 - intranet . log . aliyuncs . com ",
   " data_serve  r_list " : [
      {
         " cluster " : " ap - southeast - 2 ",
         " endpoint " : " ap - southeast - 2 - intranet . log .
 aliyuncs . com "
      }
```

```
]
```

Logtail plug-in log file (logtail_plugin.LOG)

The file contains running information about the container stdout, binlogs, http plug-in, and other plug-ins. Log levels are ranked as follows in ascending order: `INFO`, `WARN`, `ERROR`. `INFO`-type logs can be ignored.

If there is any plug-in error, for example, CANAL_RUNTIME_ALARM, when you *diagnose collection exceptions*, you can troubleshoot the error according to Logtail plug-in logs.



Note:
The file must be uploaded when you open a ticket due to plug-in exceptions.

File address

- **Linux:** `/ usr / local / ilogtail / logtail_pl ugin . LOG`
- **Container:** `/ usr / local / ilogtail / logtail_pl ugin . LOG`
- **Windows: plug-in logs are not supported.**

File example

```
$ tail / usr / local / ilogtail / logtail_pl ugin . LOG
2018 - 09 - 13 02 : 55 : 30 [ INF ] [ docker_cen ter . go : 525 ]
[ func1 ] docker fetch all : start
2018 - 09 - 13 02 : 55 : 30 [ INF ] [ docker_cen ter . go : 529 ]
[ func1 ] docker fetch all : stop
2018 - 09 - 13 03 : 00 : 30 [ INF ] [ docker_cen ter . go : 525 ]
[ func1 ] docker fetch all : start
2018 - 09 - 13 03 : 00 : 30 [ INF ] [ docker_cen ter . go : 529 ]
[ func1 ] docker fetch all : stop
2018 - 09 - 13 03 : 03 : 26 [ INF ] [ log_file_r eader . go :
221 ] [ ReadOpen ] [## 1 . 0 ## sls - zc - test - hz - pub $ docker -
stdout - config , k8s - stdout ] open file for read , file
:/ logtail_ho st / var / lib / docker / containers / 7f46afec6a
14de39b59e e9cdfbfa8a 70c2fa26f1 148b2e2f31 bd3410f5b2 d624
/ 7f46afec6a 14de39b59e e9cdfbfa8a 70c2fa26f1 148b2e2f31
bd3410f5b2 d624 - json . log offset : 40379573 status :
794354 - 64769 - 40379963
2018 - 09 - 13 03 : 03 : 26 [ INF ] [ log_file_r eader . go :
221 ] [ ReadOpen ] [## 1 . 0 ## k8s - log - c12ba2028c fb444238cd
9ac1286939 f0b $ docker - stdout - config , k8s - stdout ]
open file for read , file :/ logtail_ho st / var / lib /
docker / containers / 7f46afec6a 14de39b59e e9cdfbfa8a 70c2fa26f1
148b2e2f31 bd3410f5b2 d624 / 7f46afec6a 14de39b59e e9cdfbfa8a
70c2fa26f1 148b2e2f31 bd3410f5b2 d624 - json . log offset :
40379573 status : 794354 - 64769 - 40379963
2018 - 09 - 13 03 : 04 : 26 [ INF ] [ log_file_r eader . go :
308 ] [ CloseFile ] [## 1 . 0 ## sls - zc - test - hz - pub $ docker
- stdout - config , k8s - stdout ] close file , reason : no
read timeout file :/ logtail_ho st / var / lib / docker
/ containers / 7f46afec6a 14de39b59e e9cdfbfa8a 70c2fa26f1
```

```
148b2e2f31  bd3410f5b2  d624 / 7f46afec6a  14de39b59e  e9cdfbfa8a
70c2fa26f1  148b2e2f31  bd3410f5b2  d624 - json . log      offset :
40379963      status : 794354 - 64769 - 40379963
2018 - 09 - 13   03 : 04 : 27  [ INF ] [ log_file_r  eader . go :
308 ] [ CloseFile ] [## 1 . 0 ## k8s - log - c12ba2028c  fb444238cd
9ac1286939  f0b $ docker - stdout - config , k8s - stdout ]     close
  file ,  reason : no   read   timeout      file :/ logtail_ho  st /
var / lib / docker / containers / 7f46afec6a  14de39b59e  e9cdfbfa8a
70c2fa26f1  148b2e2f31  bd3410f5b2  d624 / 7f46afec6a  14de39b59e
e9cdfbfa8a  70c2fa26f1  148b2e2f31  bd3410f5b2  d624 - json . log
  offset : 40379963      status : 794354 - 64769 - 40379963
2018 - 09 - 13   03 : 05 : 30  [ INF ] [ docker_cen  ter . go : 525 ]
[ func1 ]  docker   fetch   all : start
2018 - 09 - 13   03 : 05 : 30  [ INF ] [ docker_cen  ter . go : 529 ]
[ func1 ]  docker   fetch   all : stop
```

Container path mapping file (docker_path_config.json)

The file is automatically created only when container files are collected. The file is used to record the mapping between the path of container files and the actual file path. The file is in JSON format.

When you *diagnose collection exceptions*, if an error indicating DOCKER_FILE_MAPPING_ALARM is reported, Logtail fails to add Docker file mapping. In this case, you can use the file to troubleshoot the error.



Note:

- The file only contains information. Any modification to the file does not take effect . The file will be automatically recreated once is deleted. This does not impact services.
- The file must be uploaded when you open a ticket due to container log collection exceptions.

File address

```
/ usr / local / ilogtail / docker_pat  h_config . json
```

File example

```
$ cat  / usr / local / ilogtail / docker_pat  h_config . json
{
   " detail " : [
      {
         " config_nam  e " : "## 1 . 0 ## k8s - log - c12ba2028c
 fb444238cd  9ac1286939  f0b $ nginx ",
         " container_  id " : " df19c06e85  4a0725ea7f  ca7e0378b0
 450f7bd312  2f94fe3e75  4d8483fd33  0d10 ",
         " params " : "{\ n    \" ID \" : \" df19c06e85  4a0725ea7f
 ca7e0378b0  450f7bd312  2f94fe3e75  4d8483fd33  0d10 \",\ n
 \" Path \" : \"/ logtail_ho  st / var / lib / docker / overlay2
 / 947db34669  5a1f65e63e  582ecfd10a  e1f57019a1  b99260b6c8
```

```
 3d00fcd189  2874 / diff / var / log \",\ n    \" Tags \" : [\ n
      \" nginx - type \",\ n       \" access - log \",\ n       \"
 _image_nam  e_ \",\ n       \" registry . cn - hangzhou . aliyuncs
 . com / log - service / docker - log - test : latest \",\ n       \"
 _container  _name_ \",\ n       \" nginx - log - demo \",\ n       \"
 _pod_name_ \",\ n       \" nginx - log - demo - h2lzc \",\ n       \"
 _namespace  _ \",\ n       \" default \",\ n       \" _pod_uid_ \",\
 n       \" 87e56ac3 - b65b - 11e8 - b172 - 00163f0086  85 \",\ n
   \" _container  _ip_ \",\ n       \" 172 . 20 . 4 . 224 \",\ n
 \" purpose \",\ n       \" test \"\ n    ]\ n }\ n "
      }
   ],
   " version " : " 0 . 1 . 0 "
}
```

3.2 Select a network type

The collected log data can be sent to Log Service through the Alibaba Cloud intranet, the Internet, or through Global Acceleration.

Network types

- Internet: Sending log data through the Internet can be limited by the network bandwidth. Additionally, network issues such as jitters, latency, and packet loss may affect the speed and stability of data transmission.
- Alibaba Cloud intranet: The Alibaba Cloud intranet supports shared bandwidth at the gigabit-level and can transmit log data more quickly and stably than the Internet. The intranet includes the Virtual Private Cloud (VPC) environment and the classic network environment.
- Global Acceleration: This network service accelerates log collection by using the edge nodes of Alibaba Cloud Content Delivery Network (CDN). Compared with the Internet, Global Acceleration provides lower transmission delay and greater stability.

Select a network type

- intranet:

  Whether your log data is transmitted through the Alibaba Cloud intranet depends on your server type and if the server and the Log Service Project are in the same

region. The Alibaba Cloud intranet can transmit log data in only the following two scenarios:

- The ECS instances of your account and the Log Service Project are in the same region.
- The ECS instances of other accounts and the Log Service Project are in the same region.

Therefore, we recommend that you create a Log Service Project in the region where your ECS instances reside, and collect logs to this Project. Then the log data of the ECS instances is written to Log Service through the Alibaba Cloud intranet, without consuming the Internet bandwidth.

 Note:
When you install a Logtail client on a server, you must select the region in which the Log Service Project resides. Otherwise, the log data cannot be collected.

- Global Acceleration:

  If your servers are located in your self-built IDCs overseas, or your servers are hosted by overseas cloud vendors, using the Internet to transmit data may cause problems such as high latency and unstable transmission. In this case, you can use *Global Acceleration* instead. *Global Acceleration* accelerates log collection by using the edge nodes of Alibaba Cloud CDN. Compared with data transmission through the Internet, Global Acceleration offers a more stable network with minimal transmission delays.

- Internet:

  We recommend that you select the Internet for the following two scenarios:

  - The server is an ECS instance, but it does not reside in the same region as the Log Service Project.
  - The server is located in your own IDC or provided by a vendors.

Server type	Reside in the same region as the Project	Configure an AliUid	Network type
ECS instances under your account	Yes	Not required	Alibaba Cloud intranet

Server type	Reside in the same region as the Project	Configure an AliUid	Network type
	No	Not required	Internet or Global Acceleration
ECS instances of other accounts	Yes	Required	Alibaba Cloud intranet
	No	Required	Internet or Global Acceleration
Cloud vendor servers or your own IDC servers	-	Required	Internet or Global Acceleration



Note:

Log Service cannot obtain owner information of the ECS instances that are under other accounts or servers. Therefore, you need to configure an AliUid for each server after you complete the Logtail client installation. Otherwise, the server heartbeat is abnormal and the server logs cannot be collected. For more information, see *Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs*.

Examples of selecting a network type

The following examples describe how to select an appropriate network in several common scenarios.



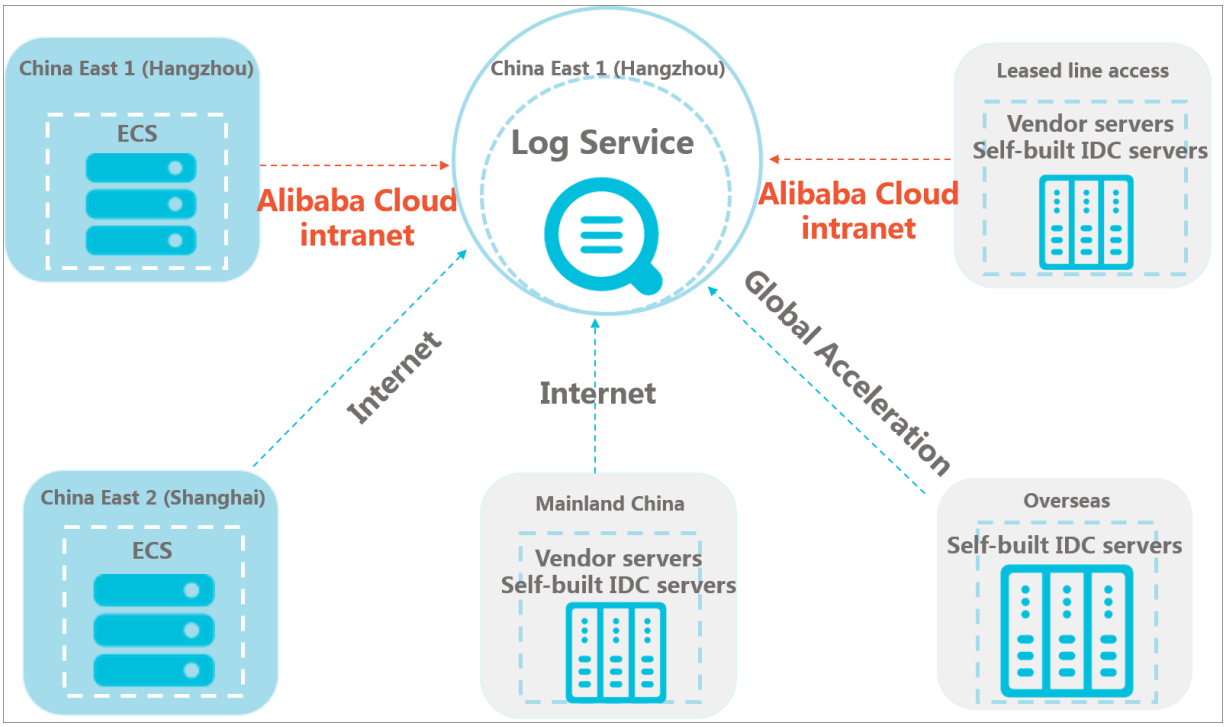
Note:

In the Global Acceleration scenario, the speed and reliability of data collection are important factors because the Log Service Project is created in the Hong Kong region but the servers are from the IDCs located worldwide. Therefore, we recommend that you select the Global Acceleration network type in the Hong Kong region when installing a Logtail client in similar scenarios. Compared with the Internet, Global Acceleration transmits log data with higher stability and performance.

Scenario	Region of the Log Service Project	Server type	Region of the ECS instance	Selected region for installing a Logtail client	Network type	Configure an AliUid
ECS and the Project are in the same region.	China East 1 (Hangzhou)	ECS of your current account	China East 1 (Hangzhou)	China East 1 (Hangzhou)	intranet	Not required
ECS and the Project are in different regions.	China East 2 (Shanghai)	ECS of your current account	China North 1 (Beijing)	China North 1 (Beijing)	Internet	Not required
Other accounts	China East 2 (Shanghai)	ECS belongs to other accounts.	China North 1 (Beijing)	China North 1 (Beijing)	Internet	Required
Server is in the local IDC.	China East 5 (Shenzhen)	Self-built IDC	-	China East 5 (Shenzhen)	Internet	Required

Scenario	Region of the Log Service Project	Server type	Region of the ECS instance	Selected region for installing a Logtail client	Network type	Configure an AliUid
Global Acceleration	Hong Kong	Self-built IDC	-	Hong Kong	Global Acceleration	Required

Figure 3-3: Examples of selecting a network type



Update configurations after a classic network is switched to a VPC

After a Logtail client is installed, you must update the network configurations if your ECS instance is switched from a classic network to a VPC. To do so, follow these steps:

1. Restart the Logtail client as the administrator.
   - Linux:

     ```
     sudo  / etc / init . d / ilogtaild   stop
     sudo  / etc / init . d / ilogtaild   start
     ```

   - Windows:

     Open Management Tool in Control Panel, open Service, right-click *LogtailWor ker* , and then select Restart.

2. Update machine group configurations.
   - Custom ID

     If a custom ID is set to define the machine group, you can directly use the VPC network without updating machine group configurations.
   - IP address

     If the ECS instance IP address is used when you define the machine group, you must replace the original IP address with the new IP address obtained by the restarted Logtail client. That is, the IP address field in the `app_info.json` file.

     The file path of `app_info.json`:

     - Linux: `/usr/local/ilogtail/app_info.json`
     - Windows x64: `C:\Program Files (x86)\Alibaba\Logtail\app_info.json`
     - Windows x32: `C:\Program Files\Alibaba\Logtail\app_info.json`

3.3 Install

3.3.1 Install Logtail in Linux

This topic describes how to install the Logtail client on Linux servers.

Supported systems

The following Linux x86-64 systems are supported:

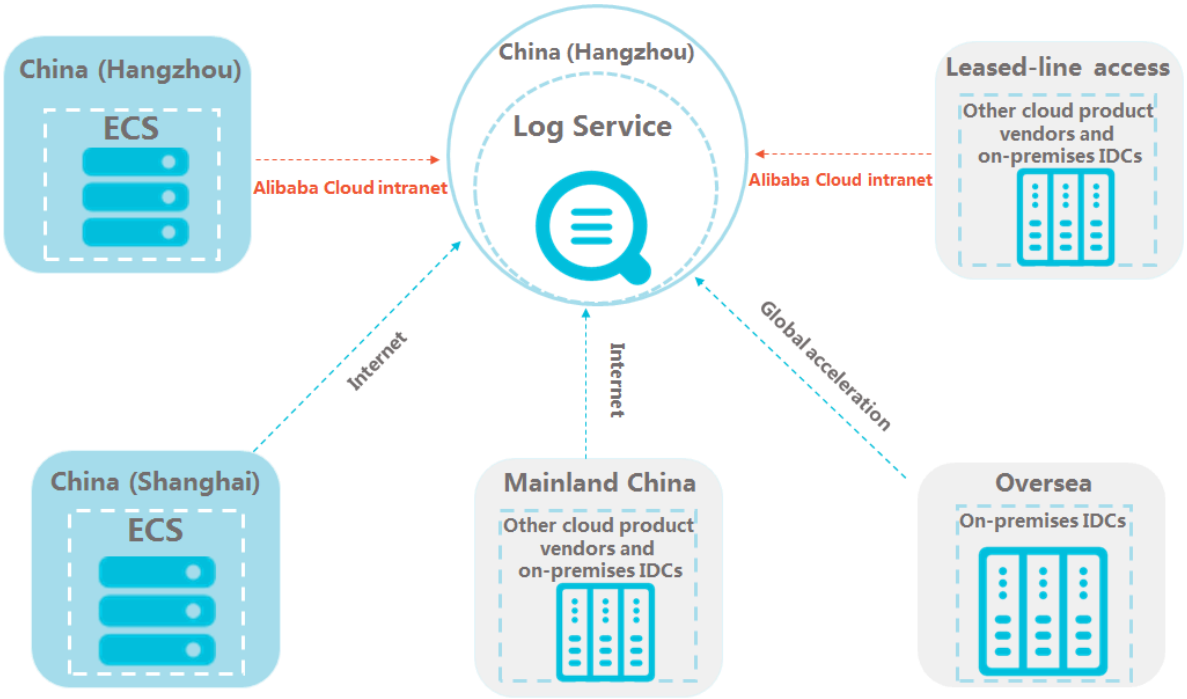
- Aliyun Linux
- Ubuntu
- Debian
- CentOS
- OpenSUSE
- Red Hat

Prerequisites

- Your account has at least one server under it.

- You have determined the required network type for log collection according to the server type and region to which the server belongs. For more information, see *Select a network type*.

Figure 3-4: Select a network type



Precautions

- If Logtail is already installed, the Logtail installer will uninstall your current version of Logtail, delete the `/ usr / local / ilogtail` directory, and reinstall Logtail. Then, Logtail is started and registered by default.
- `${ your_regio n_name }` is an *installation parameter* used for Docker and Kubernetes installation, and you can copy it from the installation parameter table.
- If the installation fails, *open a ticket* for technical support.

Installation methods

Choose one of the following installation methods according to the *network type* you selected.

- *Install Logtail through the Alibaba Cloud intranet*.
- *Install Logtail through the Internet*.
- *Install Logtail with global acceleration enabled*.

Before running the installation command, you need to replace the value of `${your_region_name}` with the actual installation parameter required. You can copy the required installation parameter based on the corresponding region from the following table.

Table 3-3: Logtail installation parameters

Region	Installation parameter
China (Hangzhou)	cn-hangzhou
China (Shanghai)	cn-shanghai
China (Qingdao)	cn-qingdao
China (Beijing)	cn-beijing
China (Zhangjiakou)	cn-zhangjiakou
China (Hohhot)	cn-huhehaote
China (Shenzhen)	cn-shenzhen
China (Chengdu)	cn-chengdu
Hong Kong	cn-hongkong
US (Silicon Valley)	us-west-1
US (Virginia)	us-east-1
Singapore	ap-southeast-1
Australia (Sydney)	ap-southeast-2
Malaysia (Kuala Lumpur)	ap-southeast-3
Indonesia (Jakarta)	ap-southeast-5
India (Mumbai)	ap-south-1
Japan (Tokyo)	ap-northeast-1
Germany (Frankfurt)	eu-central-1
UAE (Dubai)	me-east-1
UK (London)	eu-west-1

Install Logtail through the Alibaba Cloud intranet

The Alibaba Cloud intranet is a GiB-level shared network, which provides a faster and more stable data transfer than the Internet and does not consume Internet bandwidth .

The Alibaba Cloud intranet applies to Alibaba Cloud ECS servers that are deployed in the region to which the Log Service project belongs.

You can use either of the following methods to install Logtail:

- Enable automatic installation parameter selection.

  If you are unsure about the region of the ECS server, you can specify the `auto` parameter of the Logtail installer. Then, the Logtail installer will obtain the *metadata* from the server to automatically locate the region of the ECS server.

  To enable automatic installation parameter selection, follow these steps:

  1. Download the Logtail installer through the Internet by running the following command:

     Note:
     This operation requires access to the Internet and consumes about 10 KB of Internet traffic.

     ```
     wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh
     ```

  2. Use the `auto` parameter by running the following command:

     Note:
     This operation does not consume Internet traffic. The installation program will be automatically downloaded.

     ```
     ./logtail.sh install auto
     ```

- Manually install Logtail.

  You can manually install the Logtail installer through the intranet, which does not consume Internet traffic.

  To install Logtail, choose the required installation parameter and run the installation command. Specifically, you need to choose the correct parameter from the *installation parameter* table, replace `${your_region_name}` with the required parameter, and run the following installation command:

  Note:

In the installation command, `${ your_regio n_name }` indicates the region to which the Log Service project belongs. For example, the installation parameter for the China (Hangzhou) region is `cn - hangzhou` .

```
wget http://logtail-
release-${your_region_name}.oss-${your_region_name}-
```

```
internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755
 logtail.sh; ./logtail.sh install ${your_region_name}
```

Alternatively, after confirming the region to which the Log Service project belongs , you can run the corresponding installation command listed in the following table to install Logtail:

Region	Installation command
China (Hangzhou)	`wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hangzhou`
China (Shanghai)	`wget http://logtail-release-cn-shanghai.oss-cn-shanghai-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai`
China (Qingdao)	`wget http://logtail-release-cn-qingdao.oss-cn-qingdao-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao`
China (Beijing)	`wget http://logtail-release-cn-beijing.oss-cn-beijing-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing`
China (Zhangjiakou)	`wget http://logtail-release-cn-zhangjiakou.oss-cn-zhangjiakou-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiakou`
China (Hohhot)	`wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote`
China (Shenzhen)	`wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen`
China (Chengdu)	`wget http://logtail-release-cn-chengdu.oss-cn-chengdu-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu`

 **Note:**

Log Service cannot obtain owner information about non-ECS servers, so you must manually configure AliUids after installing Logtail. Otherwise, Logtail heartbeats become abnormal or Logtail cannot collect logs. For details about how to configure AliUids, see *Configure AliUids for ECS servers under other Alibaba Cloud accounts and on-premises IDCs*.

To install Logtail, choose the required installation parameter and run the installation command. Specifically, you need to choose the correct parameter from the *installation parameter* table, replace `${ your_regio n_name }` with the required parameter, and run the following installation command:

 **Note:**

In the installation command, `${ your_regio n_name }` indicates the region to which the Log Service project belongs. For example, the installation parameter for the China (Hangzhou) region is `cn - hangzhou` .

```
wget http://logtail-
release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/
```

```
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh
 install ${your_region_name}-internet
```

Alternatively, after confirming the region to which the Log Service project belongs, you can run the corresponding installation command listed in the following table to install Logtail:

Region	Installation command
China (Hangzhou)	`wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hangzhou-internet`
China (Shanghai)	`wget http://logtail-release-cn-shanghai.oss-cn-shanghai.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai-internet`
China (Qingdao)	`wget http://logtail-release-cn-qingdao.oss-cn-qingdao.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao-internet`
China (Beijing)	`wget http://logtail-release-cn-beijing.oss-cn-beijing.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing-internet`
China (Zhangjiakou)	`wget http://logtail-release-cn-zhangjiako-u.oss-cn-zhangjiako-u.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiako-u-internet`
China (Hohhot)	`wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote-internet`
China (Shenzhen)	`wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen-internet`
China (Chengdu)	`wget http://logtail-release-cn-chengdu.oss-cn-chengdu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu-internet`

In this case, you can enable the *global acceleration* function. Global acceleration accelerates log collection with the help of Alibaba Cloud CND edge nodes and provides greater advantages in terms of data transfer effectiveness and network stability for log collection through the Internet.

To install Logtail, choose the required installation parameter and run the installation command. Specifically, you need to choose the correct parameter from the *installation parameter* table, replace `${ your_regio n_name }` with the required parameter, and run the following installation command:

 **Note:**
In the installation command, `${ your_regio n_name }` indicates the region to which the Log Service project belongs. For example, the installation parameter for the China (Hangzhou) region is `cn - hangzhou` .

```
wget http://logtail-
release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/
```

```
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh
 install ${your_region_name}-acceleration
```

Alternatively, after confirming the region to which the Log Service project belongs, you can run the corresponding installation command listed in the following table to install Logtail:

Region	Installation command
China (Beijing)	`wget http://logtail-release-cn-beijing.oss-cn-beijing.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing-acceleration`
China (Qingdao)	`wget http://logtail-release-cn-qingdao.oss-cn-qingdao.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao-acceleration`
China (Hangzhou)	`wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hangzhou-acceleration`
China (Shanghai)	`wget http://logtail-release-cn-shanghai.oss-cn-shanghai.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai-acceleration`
China (Shenzhen)	`wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen-acceleration`
China (Zhangjiakou)	`wget http://logtail-release-cn-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiakou-acceleration`
China (Hohhot)	`wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote-acceleration`
China (Chengdu)	`wget http://logtail-release-cn-chengdu.oss-cn-chengdu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu-acceleration`

```
{
   " UUID " : " 0DF18E97 - 0F2D - 486F - B77F -*********",
   " hostname " : " david *******",
   " instance_i  d " : " F4FAFADA - F1D7 - 11E7 - 846C - 00163E3034
 9E_ ********* _151512954  8 ",
   " ip " : "**********",
   " logtail_ve  rsion " : " 0 . 16 . 0 ",
   " os " : " Linux ;  2 . 6 . 32 - 220 . 23 . 2 . ali1113 . el5 .
 x86_64 ; # 1   SMP   Thu   Jul   4   20 : 09 : 15   CST   2013 ;
 x86_64 ",
   " update_tim  e " : " 2018 - 01 - 05   13 : 19 : 08 "
}
```

Upgrade Logtail

You can use the Logtail installer (logtail.sh) to upgrade Logtail. The installer automatically chooses an appropriate upgrade method according to the Logtail settings.

> **Note:**
> During the upgrade, Logtail is stopped, and only necessary files are overwritten. The configuration file, the Checkpoint file, and all logs are retained.

Upgrade Logtail by running the following command:

```
#  Download   the   Logtail   installer .
 wget   http :// logtail - release - cn - hangzhou . oss - cn -
 hangzhou . aliyuncs . com / linux64 / logtail . sh  - O   logtail .
 sh ;  chmod   755   logtail . sh
#  Run   the   upgrade   command .
 sudo  ./ logtail . sh   upgrade
```

Response:

```
#  The   upgrade   succeeded .
 Stop   logtail   successful  ly .
 ilogtail   is   running
 Upgrade   logtail   success
{
   " UUID " : "***",
   " hostname " : "***",
   " instance_i  d " : "***",
   " ip " : "***",
   " logtail_ve  rsion " : " 0 . 16 . 11 ",
   " os " : " Linux ;  3 . 10 . 0 - 693 . 2 . 2 . el7 . x86_64 ; # 1
 SMP   Tue   Sep   12   22 : 26 : 13   UTC   2017 ;  x86_64 ",
   " update_tim  e " : " 2018 - 08 - 29   15 : 01 : 36 "
}

#  The   upgrade   failed :  The   current   version   is   the
 latest   version .
```

```
[ Error ]:     Already   up   to   date .
```

Manually start and stop Logtail

- Start Logtail as the admin user by running the following command:

```
/ etc / init . d / ilogtaild   start
```

- Stop Logtail as the admin user by running the following command:

```
/ etc / init . d / ilogtaild   stop
```

Uninstall Logtail

Download the Logtail installer logtail.sh, and then uninstall Logtail by running the following command:

```
wget   http :// logtail - release - cn - hangzhou . oss - cn -
hangzhou . aliyuncs . com / linux64 / logtail . sh  - O   logtail .
sh
chmod   755   logtail . sh ; ./ logtail . sh   uninstall
```

3.3.2 Install Logtail in Windows

This topic describes how to install the Logtail client on Windows servers.

Supported systems

The following Windows systems are supported:

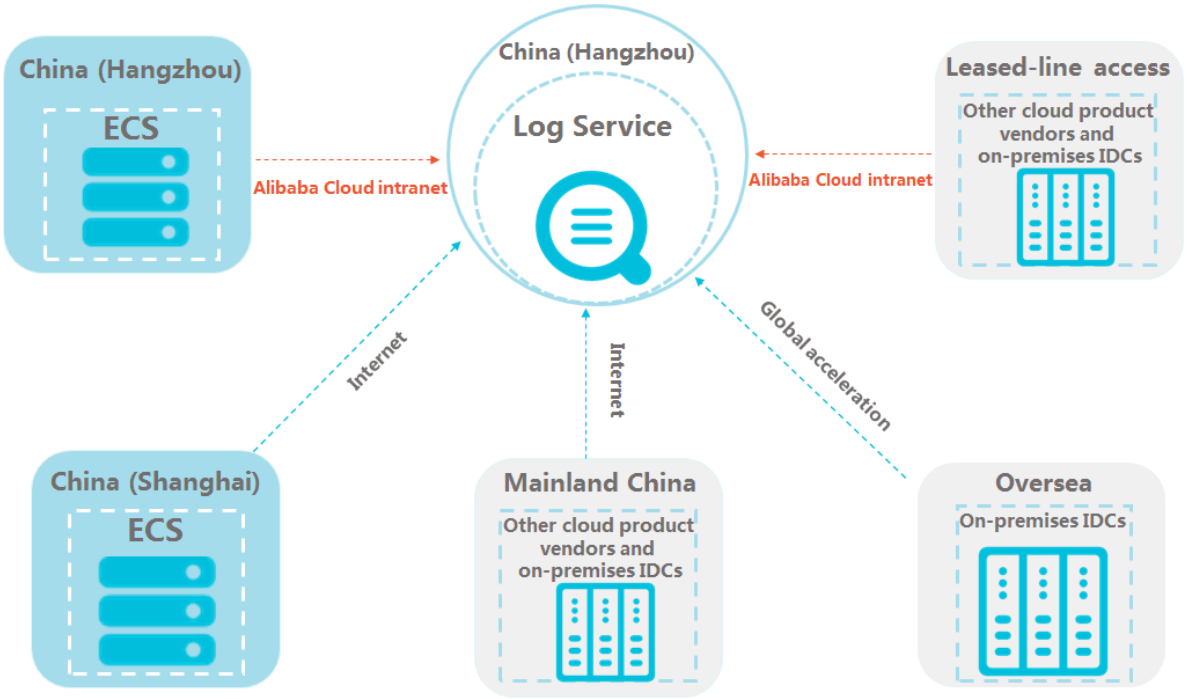
- Windows 7 (Client) 32-bit
- Windows 7 (Client) 64-bit
- Windows Server 2008 32-bit
- Windows Server 2008 64-bit
- Windows Server 2012 64-bit
- Windows Server 2016 64-bit

Prerequisites

- Your account has at least one server under it.

- You have determined the required network type for log collection according to the server type and region to which the server belongs. For more information, see *Select a network type*.

Figure 3-5: Select a network type



Install Logtail

1. Download the required Logtail installation package as follows:
   - For users in Mainland China: *Logtail installation package*
   - For users in all other areas: *Logtail installation package*
2. Decompress `logtail_in staller . zip` to the current directory.

3. **Install Logtail according to the region to which the Log Service project belongs.**

   Run Windows PowerShell or CMD as the admin user to enter the `logtail_installer` directory, where you decompress the Logtail installation package. Then, run the installation command according to the region and network type.

   The following table describes installation commands you can use according to the network type.

Region	Alibaba Cloud intranet	Internet	Global acceleration
China (Qingdao)	`.\ logtail_installer . exe install cn - qingdao`	`.\ logtail_installer . exe install cn - qingdao - internet`	`.\ logtail_installer . exe install cn - qingdao - accelerati on`
China (Beijing)	`.\ logtail_installer . exe install cn - beijing`	`.\ logtail_installer . exe install cn - beijing - internet`	`.\ logtail_installer . exe install cn - beijing - accelerati on`
China (Zhangjiakou)	`.\ logtail_installer . exe install cn - zhangjiako u`	`.\ logtail_installer . exe install cn - zhangjiako u - internet`	`.\ logtail_installer . exe install cn - zhangjiako u - accelerati on`
China (Hohhot)	`.\ logtail_installer . exe install cn - huhehaote`	`.\ logtail_installer . exe install cn - huhehaote - internet`	`.\ logtail_installer . exe install cn - huhehaote - accelerati on`
China (Hangzhou)	`.\ logtail_installer . exe install cn - hangzhou`	`.\ logtail_installer . exe install cn - hangzhou - internet`	`.\ logtail_installer . exe install cn - hangzhou - accelerati on`

Region	Alibaba Cloud intranet	Internet	Global acceleration
China (Shanghai)	`.\ logtail_installer.exe install cn-shanghai`	`.\ logtail_installer.exe install cn-shanghai-internet`	`.\ logtail_installer.exe install cn-shanghai-acceleration`
China (Shenzhen)	`.\ logtail_installer.exe install cn-shenzhen`	`.\ logtail_installer.exe install cn-shenzhen-internet`	`.\ logtail_installer.exe install cn-shenzhen-acceleration`
China (Chengdu)	`.\ logtail_installer.exe install cn-chengdu`	`.\ logtail_installer.exe install cn-chengdu-internet`	`.\ logtail_installer.exe install cn-chengdu-acceleration`
Hong Kong	`.\ logtail_installer.exe install cn-hongkong`	`.\ logtail_installer.exe install cn-hongkong-internet`	`.\ logtail_installer.exe install cn-hongkong-acceleration`
US (Silicon Valley)	`.\ logtail_installer.exe install us-west-1`	`.\ logtail_installer.exe install us-west-1-internet`	`.\ logtail_installer.exe install us-west-1-acceleration`
US (Virginia)	`.\ logtail_installer.exe install us-east-1`	`.\ logtail_installer.exe install us-east-1-internet`	`.\ logtail_installer.exe install us-east-1-acceleration`
Singapore	`.\ logtail_installer.exe install ap-southeast-1`	`.\ logtail_installer.exe install ap-southeast-1-internet`	`.\ logtail_installer.exe install ap-southeast-1-acceleration`

Region	Alibaba Cloud intranet	Internet	Global acceleration
Australia (Sydney)	`.\ logtail_installer . exe install ap - southeast - 2`	`.\ logtail_installer . exe install ap - southeast - 2 - internet`	`.\ logtail_installer . exe install ap - southeast - 2 - acceleration`
Malaysia (Kuala Lumpur)	`.\ logtail_installer . exe install ap - southeast - 3`	`.\ logtail_installer . exe install ap - southeast - 3 - internet`	`.\ logtail_installer . exe install ap - southeast - 3 - acceleration`
Indonesia (Jakarta)	`.\ logtail_installer . exe install ap - southeast - 5`	`.\ logtail_installer . exe install ap - southeast - 5 - internet`	`.\ logtail_installer . exe install ap - southeast - 5 - acceleration`
India (Mumbai)	`.\ logtail_installer . exe install ap - south - 1`	`.\ logtail_installer . exe install ap - south - 1 - internet`	`.\ logtail_installer . exe install ap - south - 1 - acceleration`
Japan (Tokyo)	`.\ logtail_installer . exe install ap - northeast - 1`	`.\ logtail_installer . exe install ap - northeast - 1 - internet`	`.\ logtail_installer . exe install ap - northeast - 1 - acceleration`
Germany (Frankfurt)	`.\ logtail_installer . exe install eu - central - 1`	`.\ logtail_installer . exe install eu - central - 1 - internet`	`.\ logtail_installer . exe install eu - central - 1 - acceleration`
UAE (Dubai)	`.\ logtail_installer . exe install me - east - 1`	`.\ logtail_installer . exe install me - east - 1 - internet`	`.\ logtail_installer . exe install me - east - 1 - acceleration`

Region	Alibaba Cloud intranet	Internet	Global acceleration
UK (London)	`.\ logtail_install eu-west-1`	`.\ logtail_install eu-west-1 internet`	`.\ logtail_install eu-west-1 acceleration`

> **Note:**
> When you use Logtail on a server deployed in an on-premises IDC or provided by another cloud product vendor, Log Service cannot obtain the owner information about ECS servers under other Alibaba Cloud accounts or other types of servers. In this case, after installing Logtail, you need to manually configure AliUids for the servers by following the instructions provided in *Configure AliUids for ECS servers under other Alibaba Cloud accounts and on-premises IDCs*. Otherwise, Logtail heartbeats become abnormal, or Logtail cannot collect logs.

Installation path

By default, Logtail is installed in the specified path, which cannot be changed. In this path, you can *view the Logtail version* in the `app_info.json` file or *uninstall Logtail*.

The installation path is:

- `C:\Program Files\Alibaba\Logtail` in Windows 32-bit systems
- `C:\Program Files (x86)\Alibaba\Logtail` in Windows 64-bit systems

> **Note:**
> Logtail is a 32-bit program. For Windows 64-bit systems, Logtail will be installed in the `Program Files (x86)` directory.

View the Logtail version

Logtail is automatically installed to the *default directory*. You can open the `app_info.json` file to view the `logtail_version` field, which indicates your Logtail version.

The following examples shows that the Logtail version is 1.0.0.0:

```
{
 " logtail_ve  rsion " : " 1 . 0 . 0 . 0 "
}
```

Upgrade Logtail

- Automatic upgrade

  In normal cases, Windows supports automatic Logtail upgrades.

- Manual upgrade

  You need to manually upgrade Logtail if the source version is earlier than 1.0.0.0 and the target version is 1.0.0.0 or later. For more information about how to manually upgrade Logtail, see *Install Logtail*.

  > **Note:**
  > Manually upgrading Logtail means that Logtail will be automatically uninstalled and then reinstalled. In this case, files in the original installation path will be deleted. We recommend that you back up the files before manually upgrading Logtail.

Manually start and stop Logtail

In the Control Panel, choose System and Security > Administrative Tools, and then open the Services program.

Find the target service according to your Logtail version:

- For Logtail 0.x.x.x: LogtailWorker
- For Logtail 1.0.0.0 and later: LogtailDaemon

Then, perform the following operations as needed:

- Manually start Logtail: Right-click Logtail and click Start in the shortcut menu.
- Stop Logtail: Right-click Logtail and click Stop in the shortcut menu.
- Restart Logtail: Right-click Logtail and click Restart in the shortcut menu.

Uninstall Logtail

Run Windows PowerShell or CMD as the admin user to enter the `logtail_installer` directory, namely, the directory for decompressing the installation package you have downloaded. Then, run the following command:

```
.\ logtail_install . exe uninstall
```

After Logtail is deleted successfully, the Logtail *installation path* will be deleted. However, some residual configuration information will be retained in the `C:\LogtailData` directory. You can manually delete the information as needed. The information includes the following:

- `checkpoint` : contains checkpoint information of all agents (for example, the Windows event log agent).
- `logtail_check_point` : contains major checkpoint information of Logtail.
- `users` : contains all AliUids.

3.3.3 Configure startup parameters

This topic describes the Logtail startup configuration parameters. You can configure the startup parameters by following this topic when you have any special requirements.

Scenarios

In the following scenarios, you must configure the Logtail startup configuration parameters:

- The metadata information of each file, such as file signature, collection location, and file name, must be maintained in the memory.
- Therefore, the memory usage may be high if a large number of log files are to be collected.
- The CPU usage is high because the volume of log data is large and the traffic sent to Log Service is heavy.

Startup configuration

- File path

```
/ usr / local / ilogtail / ilogtail_c  onfig . json
```

- File format

  JSON

- File sample (which only shows partial configuration items)

```
{
    ...
    " cpu_usage_  limit " :  0 . 4 ,
    " mem_usage_  limit " :  100 ,
    " max_bytes_  per_sec " :  2097152 ,
    " process_th  read_count " :  1 ,
    " send_reque  st_concurr  ency " :  4 ,
    " buffer_fil  e_num " :  25 ,
    " buffer_fil  e_size " :  20971520 ,
    " buffer_fil  e_path " : "",
    ...
}
```

Common configuration parameters

Parameter name	Parameter description	Value
cpu_usage_limit	The CPU usage threshold. Calculated per core. In most cases, the single-core processing capability is about 24 MB/s in simple mode and about 12 MB/s	Double type. The minimum value is 0.1, and the maximum value is the number of CPU cores of the current machine. The default value is 2. For example, the value 0.4 indicates the CPU usage of Logtail is limited to 40% of single-core CPUs. Logtail restarts automatically when the threshold is exceeded.

Parameter name	Parameter description	Value
mem_usage_limit	The usage threshold of resident memory. To collect more than 1,000 distinct files, properly increase the threshold value.	Int type. Measured in MBs. The minimum value is 128, and the maximum value is the current machine effective memory value. The default value is 2048. For example, the value 100 indicates the memory usage of Logtail is limited to 100 MB. Logtail restarts automatically when the threshold is exceeded.
max_bytes_per_sec	The traffic limit on the raw data sent by Logtail, more than 20MB/s stream is not limited.	Int type. Measured in bytes per second. The range is 1024 - 52428800, the default value is 20971520. For example, the value 2,097,152 indicates the data transfer rate of Logtail is limited to 2 MB/s.
process_thread_count	The number of threads that Logtail processes written data of log files. Generally supports a write speed of 24 MB/s in simple mode and 12 MB/s in full mode. By default, it is not required to adjust this value, but you can increase the threshold value when necessary.	Int type. Measured in units. The range is 1 - 64, the default value is 1.

Parameter name	Parameter description	Value
send_request_concurrency	The number of asynchronous concurrency. By default, Logtail sends data packets asynchronously. You can set a larger asynchronous concurrency value if the write TPS is large. Can be supported with a single concurrency of 0.5 Mb/s ~ It is based on the network delay to calculate the throughput of 1 Mb/s network. Note: Quantity based on the condition that one concurrency supports 0.5–1 MB/s network throughout. The actual concurrency quantity varies with network delay.	Int type. Measured in units. The range is 1 - 1000, default value is 20.

Parameter name	Parameter description	Value
buffer_file_num	When a network exception occurs or the writing quota is exceeded, Logtail writes the logs that are parsed in real time to a local file (located in the installation directory) as a cache and then tries to resend the logs to Log Service after the recovery. This parameter indicates the maximum number of cached files.	Int type. Measured in units. The range is 1 - 100, default value is 25.
buffer_file_size	The maximum number of bytes that a cached file allows. The (buffer_file_num * buffer_file_size) indicates the maximum disk space available for cached files.	Int type. Measured in bytes. The range is 1048576 - 104857600, the default is 20971520 Bytes (20 MB).
buffer_file_path	The directory that stores cached files. After modifying this parameter value, you must manually move the files named in the format of *logtail_buffer_file_** in the old cache directory to the new cache directory so that Logtail can read the cached files and delete them after sending logs.	The default value is null, indicating the cached files are stored in the Logtail installation directory (*/usr/local/ilogtail*).

Parameter name	Parameter description	Value
bind_interface	The name of the network card that is bound to the local machine, such as eth1 (only Linux versions are supported).	The parameter is empty by default. The available network card is bound automatically. If this parameter is configured, Logtail will force to use this network card to upload logs.

Parameter name	Parameter description	Value
check_point_filename	The full path stored by the checkpoint file, which is used to customize the checkpoint storage location of Logtail. We recommend that Docker users modify this file storage address and mount the directory where the checkpoint file resides to the host. Otherwise, duplicate collection occurs when the container is released because the checkpoint information is missing. For example, configure the check_point_filename in Docker as /data/logtail/check_point.dat, and add -v /data/docker1/logtail:/data/logtail in the Docker startup command to mount the /data/docker1/logtail directory of the host to the /data/logtail directory of Docker.	The default value is */ tmp / logtail_check_point* .

Parameter name	Parameter description	Value
user_config_file_path	The full path stored by the checkpoint file, which is used to customize the checkpoint storage location of Logtail. We recommend that Docker users modify this file storage address and mount the directory where the checkpoint file resides to the host. Otherwise, duplicate collection occurs when the container is released because the checkpoint information is missing. For example, in Docker, configure user_config_file_path as */data/logtail/user_log_config.json*, and add -v /data/docker1/logtail:/data/logtail in the Docker startup command to mount the */data/docker1/logtail* directory of the host to the */data/logtail* directory of Docker.	The default file path is the directory where the binary process resides, and the file name is *user_log_config.json*.

Parameter name	Parameter description	Value
discard_old_data	Whether to discard historical logs. If the parameter is set to true, the logs that were generated before 12 hours will be discarded.	The parameter values are of the bool type, and the default value is true.
working_ip	The IP address of your host reported by Logtail. If the parameter value is empty, Logtail automatically obtains your host IP address.	The default value is empty.
working_hostname	The host name reported by Logtail. If the parameter value is empty, Logtail automatically obtains your host name.	The parameter values are of the string type, and the default value is empty.
max_read_buffer_size	The maximum size of a log. If the size of a single log exceeds 512 KB, you can adjust the parameter value.	The parameter values are of the long type, and the default value is 524288 bytes (512 KB).



Note:

- The preceding table only lists the common startup parameters that need your attention. If *ilogtail_config.json* has parameters that are not listed in the table, the default values are applied.
- Add or modify the values of configuration parameters as per your needs. Unnecessary configuration parameters do not need to be added to *ilogtail_config.json*.

Modify configurations

1. Configure *ilogtail_config.json* as per your needs.

   Confirm the modified configurations are in the valid JSON format.

2. Restart Logtail to apply the modified configurations.

```
/ etc / init . d / ilogtaild   stop
/ etc / init . d / ilogtaild   start
/ etc / init . d / ilogtaild   status
```

3.4 Machine Group

3.4.1 Overview

Log Service uses machine groups to manage all the servers whose logs are collected by Logtail clients.

A machine group is a virtual group that contains multiple servers. If you want the logs of multiple servers to be collected by Logtail clients with the same configuration, you can add the servers to a machine group and apply the Logtail configuration to the machine group.

You can define a machine group by using either of the following identification types:

- *IP address*: Add the IP addresses of all the servers to the machine group. Each server in the group can be identified by using its unique IP address.
- *Custom ID*: Customize an ID for the machine group and use this same custom ID for each server of the machine group.



Note:

- Before adding a server of other cloud vendors or your local IDC, or adding an ECS instance of other accounts to a machine group, you must set an AliUid for the server or instance. For more information, see *Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs*.
- You cannot add Windows servers and Linux servers to the same machine group.

IP address-based machine group

You can add multiple servers to a machine group by adding their IP addresses to the machine group. Then you can configure the Logtail clients on all the servers at the same time.

- If you use ECS servers that are not bound to hostnames, and the network types of these ECS servers remain unchanged, you can use their private IP addresses to define the machine group.

- In other cases, use the server IP address obtained automatically by the Logtail client when you define a machine group. The IP address of each server is recorded in the IP address field of the `app_info.json` server file on the server.

  > Note:
  > `app_info.json` is the file that records the internal information of the Logtail client. The internal information includes the server IP addresses obtained by the Logtail client automatically. Manually modifying the IP address field of this file does not change the IP addresses obtained by the Logtail client.

A Logtail client automatically obtains a server IP address by using the following methods:

- If the IP address of a server has been bound with its host name in the `/etc/hosts` server file, the Logtail client automatically obtains the IP address.
- If the IP address of a server has not been bound with its host name, the Logtail automatically obtains the IP address of the first Network Interface (NI) on the server.

> Note:
> Whether the Alibaba Cloud intranet is used for data collection does not depend on whether you use a private IP address to define a machine group. Your server log data can be collected to Log Service through the Alibaba Cloud intranet only when you use an ECS instance of Alibaba Cloud and you have selected Alibaba Cloud intranet (Classic Network and VPC) when installing a Logtail on the instance.

For more information, see *Create a machine group with an IP address as its identifier*.

Custom ID-based machine group

In addition to IP addresses, custom IDs can also be used to define machine groups.

We recommend that you use a machine group defined by a custom ID in the following scenarios:

- In a custom network, for example a VPC, different servers may have the same IP address. In that case, Log Service cannot manage the Logtail clients on the servers. Using a custom ID to define a machine group can eliminate such a problem.

- Multiple servers in a machine group can use one custom ID to implement machine group auto scaling. If you set the same custom ID for a new server, the Log Service identifies the new server automatically and adds it to the machine group.

Typically, the system consists of multiple modules. Each module can be expanded horizontally. That is, multiple servers can be added to each module. By creating a machine group separately for each module, you can collect logs by module. Therefore, you need to create a custom ID for each module, and set the machine group ID for the servers of each module. For example, a common website consists of an HTTP request processing module, a cache module, a logic processing module, and a storage module. The custom IDs can be set as http_module for the HTTP request processing module, cache_module for the cache module, logic_module for the logic processing module, and store_module for the storage module.

For more information, see *Create an ID to identify a machine group*.

3.4.2 Create a machine group with an IP address as its identifier

Log Service now supports machine groups with IP addresses as identifiers. After adding the server IP addresses obtained by Logtail to the target machine groups, you can use the same Logtail Config to collect logs from the servers.

Prerequisites

- You have created a project and a Logstore.
- You own at least one server. If the server is an Alibaba Cloud ECS server, make sure that the server is deployed in the region to which the project belongs.
- You have installed Logtail on the server. For details about how to install Logtail on a server, see *Install Logtail in Linux* and *Install Logtail in Windows*.
- For servers of other cloud providers, on-premises IDCs, and ECS servers under other Alibaba Cloud accounts, confirm that you have configured AliUids for them. For details about how to configure AliUids, see *Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs*.

Context

You can collect logs through the Alibaba Cloud intranet even if you have not configured an intranet IP address. Server logs can be collected and sent to Log Service through the Alibaba Cloud intranet only when you use an Alibaba Cloud ECS server,

the ECS server is deployed in the region to which the project belongs, and you select Alibaba Cloud intranet (Classic Network and VPC) when installing Logtail.

Procedure

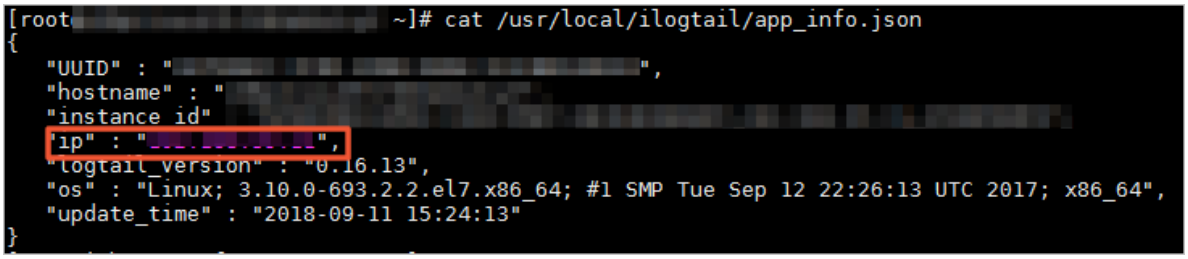
1. View the IP address of the server. The IP address is automatically generated by Logtail.

   The IP address is recorded in the ip field in the `app_info.json` file.

   You can view the file in the server with Logtail installed. The file path is:

   - In Linux: `/usr/local/ilogtail/app_info.json`
   - In Windows x64: `C:\Program Files (x86)\Alibaba\Logtail\app_info.json`
   - In Windows x32: `C:\Program Files\Alibaba\Logtail\app_info.json`

   The following figure shows an example IP address of a Linux server.

2. Log on to the *Log Service console*, and then click the target project name.
3. Choose LogHub - Collect > Logtail Machine Group to view the Machine Groups list.
4. In the upper-right corner, click Create Machine Group.

   Alternatively, after creating a Logtail Config in the data access wizard, click Create Machine Group on the Apply to Machine Group page.
5. Create a machine group.

   a) Enter a Name.

   The name must be 3 to 128 characters in length and can contain lowercase letters, numbers, hyphens (-), and underscores (_). It must start and end with a lowercase letter or number.

   b) invalid content

   Note:

Exercise caution when you set a name for the machine group because the name cannot be changed once it is set.

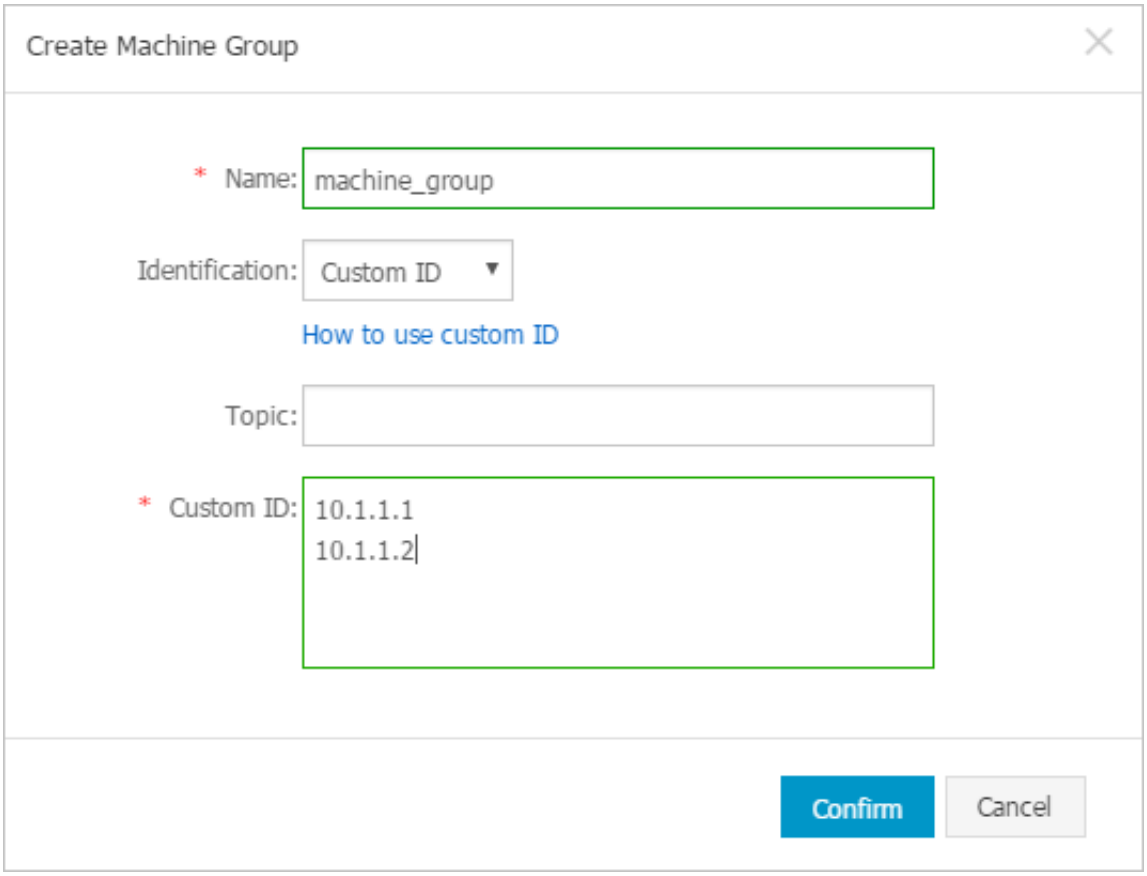
c) Set Identification to IP Address.

d) Enter an IP address.

The IP address is the server IP address you obtained from *1*.

Note:

- You need to obtain the server IP address by following the instructions provided in *1*.
- When multiple servers exist in the machine group, you need to use line breaks to separate the IP addresses of the servers.
- You cannot add both Windows servers and Linux servers to the machine group.



6. Optional: Enter a Topic.

   For more information about machine group topics, see *Set a log topic*.

7. Click Confirm.

Result

You can view the newly created machine group in the Machine Groups list.



3.4.3 Create an ID to identify a machine group

In addition to IP addresses, you can also use custom IDs as identifiers of machine groups.

The advantages of using custom IDs as identifiers for machine groups are described in the following scenarios:

- In a custom network, such as a VPC, IP addresses conflicts may occur among server. As a result, Log Service cannot manage Logtail. In this case, custom IDs can be added to the servers to prevent IP address conflicts.
- A custom ID can be configured for multiple servers to achieve elastic scaling of a machine group. Specifically, you can add the same custom ID to new servers so that Log Service can automatically identify the servers and add them to the machine group.

Procedure

1. Set custom IDs on the servers.
   - For Logtail in Linux:

     Set custom IDs by using the `/ etc / ilogtail / user_defin ed_id` file.

     The following is an example:

     ```
     # vim / etc / ilogtail / user_defin ed_id
     ```

     Enter `userdefine d` in this file.
   - For Logtail in Windows:

     Set custom IDs by using the `C :\ LogtailDat a \ user_defin ed_id` file.

     The following is an example:

     ```
     C :\ LogtailDat a > more user_defin ed_id
     userdefine d_windows
     ```

   Note:

- You cannot add Linux and Windows servers into the same machine group. Therefore, you must set different custom IDs for Linux and Windows servers.
- When multiple custom IDs are configured for a server, you need to use line breaks to separate them.
- If the `/etc/ilogtail/` or `C:\LogtailData\a` directory or the `/etc/ilogtail/user_defined_id` or `C:\LogtailData\a\user_defined_id` file does not exist, manually create one.

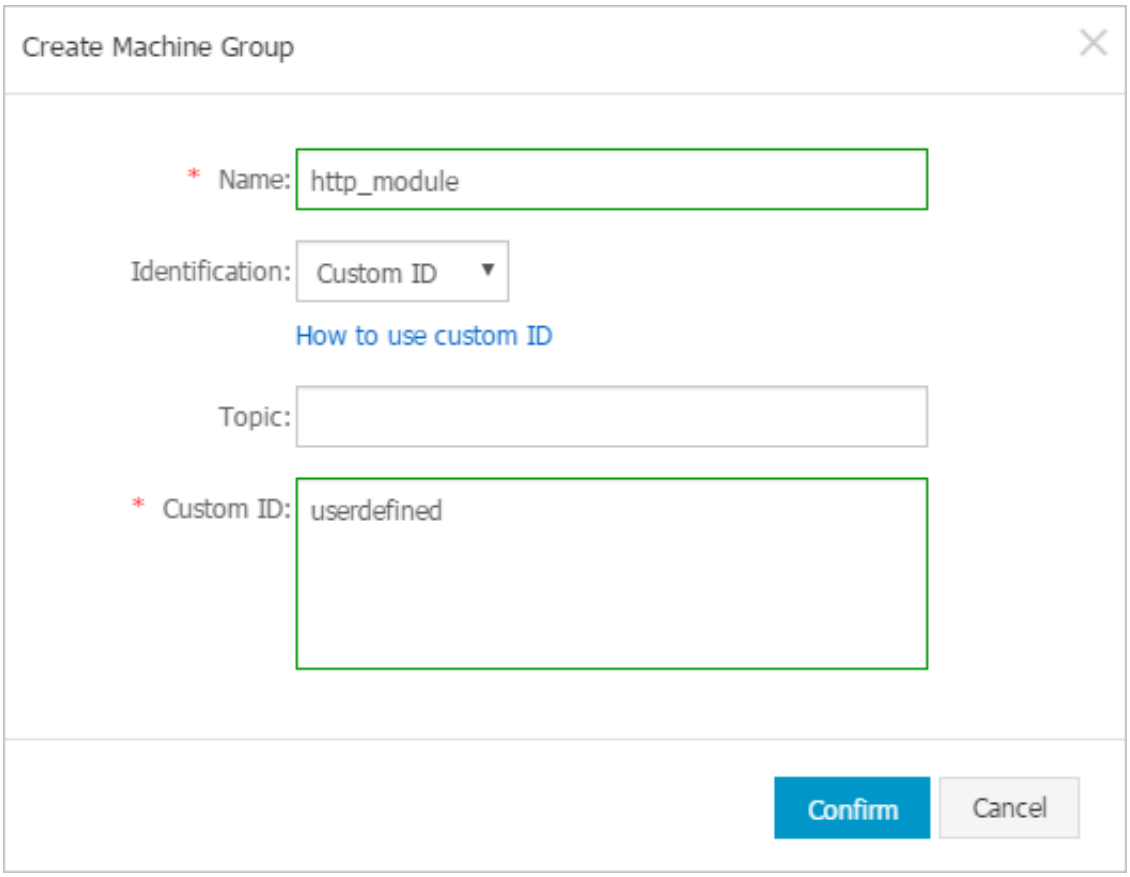
2. Create a machine group.
   a. Log on to the *Log Service console*, and then click the target project name.
   b. In the left-side navigation pane, click Logtail Machine Group.
   c. On the Machine Groups page, click Create Machine Group the upper-right corner.
   d. Set the machine group configurations.
      - Name: Enter a name.

        The name must be 3 to 128 characters in length and can contain lowercase letters, numbers, hyphens (-), and underscores (_). It must start and end with a lowercase letter or number.

         Note:

Exercise caution when you set a name for the machine group because the name cannot be changed once it is set.

- Identification: Select Custom ID.
- (Optional) Topic: Enter a topic. For more information about machine group topics, see *Set a log topic*.
- Custom ID: Enter the custom ID obtained from step 1.

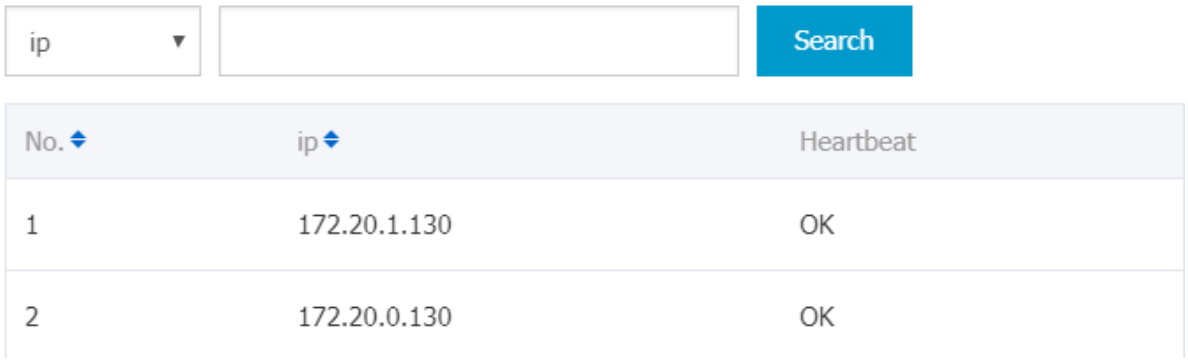


e. Click Confirm.

Note:
To scale out a server group, you only need to add the custom ID to the new server.

3. Check the status of the machine group.

   On the Machine Groups page, locate the target machine group, and click Status in the Actions column. Then, you can view the list containing all servers using the same custom ID and their heartbeat information.



Disable custom IDs

If you want use server IP addresses as machine group identifiers, delete the user_defined_id file. The setting takes effect within one minute.

- In Linux:

```
rm  - f  / etc / ilogtail / user_defin  ed_id
```

- In Windows:

```
del   C :\ LogtailDat  a \ user_defin  ed_id
```

Effective time

By default, addition, deletion, and modification of the user_defined_id file take effect within one minute.

If you want the setting to take effect immediately, run the following command and restart Logtail:

- In Linux:

```
/ etc / init . d / ilogtaild   stop
/ etc / init . d / ilogtaild   start
```

- In Windows:

  Choose Control Panel > Administrative Tools > Services. Then, right-click the LogtailWorker service and choose Restart from the shortcut menu.

Example

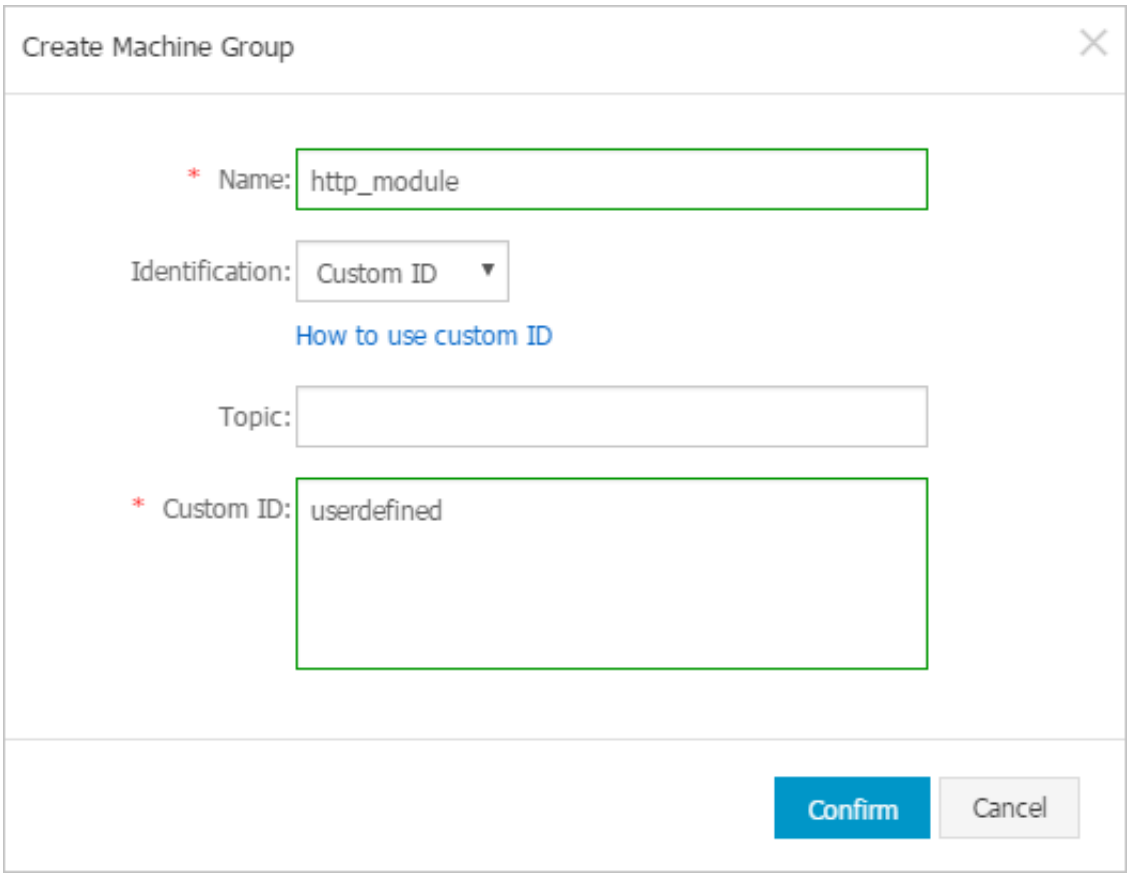
The system is composed of multiple modules, each of which can contain multiple servers. For example, a common website can be divided into a frontend HTTP request processing module, a cache module, a logic processing module, and a storage module. Each module can be individually expanded. Therefore, you need to enable Log Service to collect logs from new servers in real time.

1. Create a custom ID.

   After installing the Logtail client, use custom IDs as machine group identifiers. In this example, there are four identifiers (indicating the four modules): http_module , cache_module, logic_module, and store_module

2. Create a machine group.

   Set Identification to the actual custom ID of the machine group. The following figure uses the http_module machine group as an example.



3. In the Machine Group Status dialog box, view the list containing all servers using the same custom ID and their heartbeat information.
4. Add the custom ID to the 10.1.1.3 server if the server is added to the machine group. Then, you can view the new server in the Machine Group Status dialog box.

3.4.4 Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs

If Logtail is installed on ECS servers under other Alibaba Cloud accounts, provided by other cloud vendors, or located in on-premises IDCs, you must configure AliUids for the servers so that they can be added into machine groups for log collection.

Context

If the target server for log collection through Logtail is purchased by another Alibaba Cloud account or provided by another cloud vendor, you need to install Logtail on the server and configure an AliUid for it. By doing so, you grant your Alibaba Cloud account the permissions to access and collect logs from the server. Otherwise, the server does not receive heartbeat information and cannot collect logs.

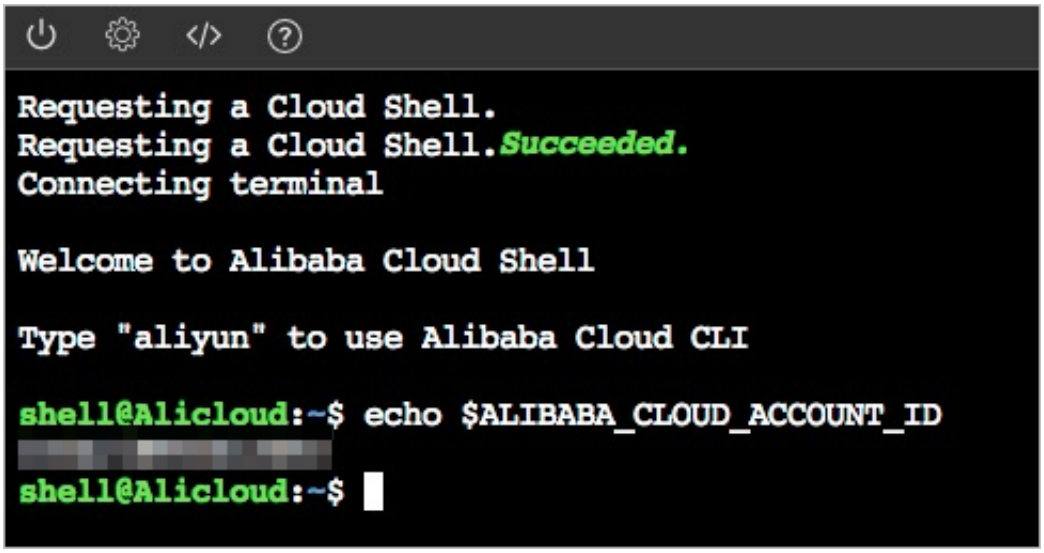
Prerequisites

- **The target server for log collection is under another Alibaba Cloud account, provided by another cloud vendor, or located in an on-premises IDC.**
- **The Logtail client is installed on the server.**

  For more information, see *Install Logtail in Linux* **and** *Install Logtail in Windows* **as needed.**

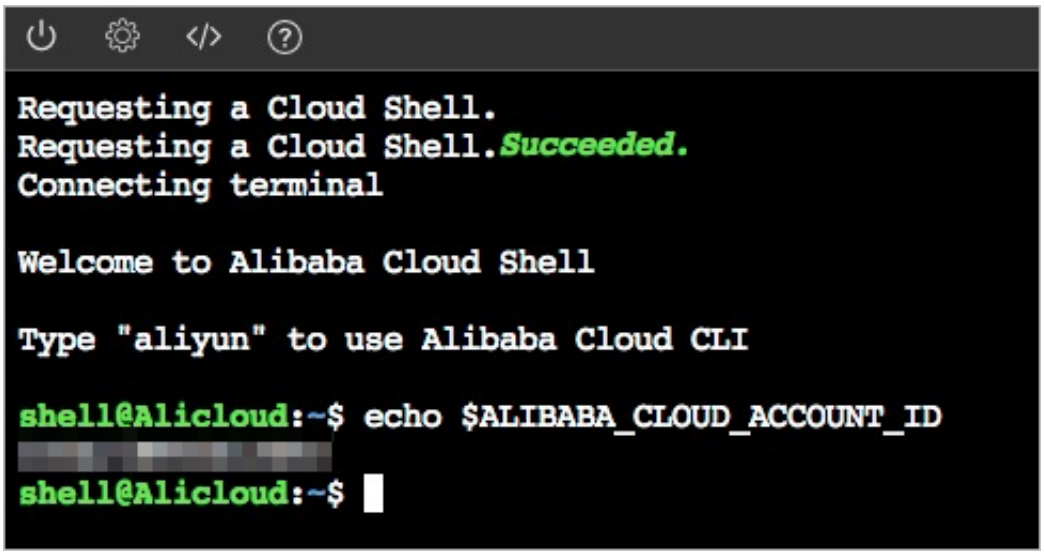
Procedure

1. View the Alibaba Cloud account ID, namely, the AliUid.
   a. Log on to the *Log Service console*.
   b. In the upper-right corner, click the icon. For more information, see *Cloud Shell*.
   c. Run the `echo $ALIBABA_CLOUD_ACCOUNT_ID` command to obtain your AliUid.



You can view the AliUid of the account to which the Log Service project belongs on the Account Management page.

Figure 3-6: View your Alibaba Cloud account ID



2. Configure an AliUid for the server.

   - In Linux:

     Create a file named after the AliUid in the `/ etc / ilogtail / users` directory. If the directory does not exist, you need to create one. You can configure multiple AliUids for a single server by running a command similar to the following:

     ```
     touch  / etc / ilogtail / users / 1 **************
     touch  / etc / ilogtail / users / 1 **************
     ```

     If you do not need Logtail to collect data to your Log Service project, you can delete the AliUid:

     ```
     rm  / etc / ilogtail / users / 1 **************
     ```

   - In Windows:

     Create a file named after the AliUid in the `C :\ LogtailDat  a \ users` directory.

     If you want to delete the AliUid, you can simply delete this file. (`C :\ LogtailDat  a \ users \ 1 **************` is used as an example file.)

     Note:

     - After an AliUid is configured for a server, the Alibaba Cloud account has the permission to collect logs from the server by using Logtail. You need to delete unnecessary AliUid files from the server in a timely manner.
     - Addition and deletion of an AliUid take effect within 1 minute.

3.4.5 Create a Logtail configuration

The Logtail client provides an easy way to collect logs from Elastic Compute Service (ECS) instances in the Log Service console. After installing the Logtail client, you must create a log collection configuration for the Logtail client. For how to install Logtail, see *Install Logtail in Linux* and *Install Logtail in Windows*. You can create and modify the Logtail configurations of LogStores in the LogStore list.

Create a Logtail configuration

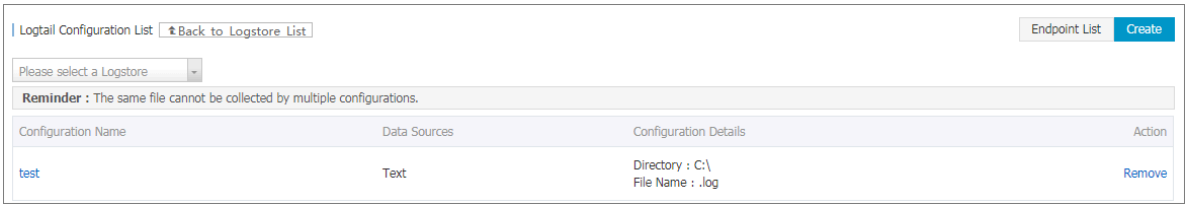
For how to create a Logtail configuration in the Log Service console, see *Collect text logs* and *Syslog*.

View Logtail configuration list

1. Log on to the Log Service console.
2. On the Project List page, click the project name, to enter the Logstore List page.
3. On the Logstore List page, click Manage at the right of the Logstore. Logtail Configuration List page appears.

   All the configurations of this Logstore are displayed on the page, including the configuration name, data sources, and configuration details. When the data source is Text, the file path and file name are displayed under Configuration Details.

   Figure 3-7: Logtail configuration list



> **Note:**
> A file can be collected by only one configuration.

Modify a Logtail configuration

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Manage at the right of the Logstore. The Logtail Configuration List page appears.
4. Click the name of the Logtail configuration to be modified.

   You can modify the log collection mode and specify the machine group to which the modified mode is applied. The configuration modification process is the same as the configuration creation process.

Delete a Logtail configuration

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Manage at the right of the Logstore. The Logtail Configuration List page appears.

4. Click Remove at the right of the Logtail configuration to be deleted.

   After the configuration is deleted successfully, it is unbound from the machine groups that applied this configuration and Logtail stops collecting the log files of the deleted configuration.

   > Note:
   > You must delete all the Logtail configurations in a Logstore before deleting the Logstore.

3.4.6 Manage a machine group

Log Service manages all the Elastic Compute Service (ECS) instances whose logs need to be collected by using the Logtail client in the form of machine groups. You can go to the Machine Groups page by clicking a project name on the Project List page and then clicking LogHub - Collect > Logtail Machine Group in the left-side navigation pane on the Logstore List page. You can create, modify, and delete a machine group, view the machine group list and status, manage the configurations, and use the machine group identification in the Log Service console.

Create a machine group

You can define a machine group by using:

- IP: Define the machine group name and add the intranet IP addresses of a group of machines. For more information, see *Create a machine group with an IP address as its identifier*.
- User-defined identity: Define an identity for the machine group and configure the identity on the corresponding machine for association. For more information, see *Create an ID to identify a machine group*.

For how to create a machine group, see *Create a machine group with an IP address as its identifier*.

View machine group list

1. Log on to the Log Service console.

2. On the Logstore List page, click Logtail Machine Group in the left-side navigation pane. The Machine Groups page appears.

   View all of the machine groups in the project.

   Figure 3-8: View a list of machine groups



Modify a machine group

After creating a machine group, you can adjust the ECS instances in the machine group as per your needs.

> **Note:**
> The machine group name cannot be modified after the machine group is created.

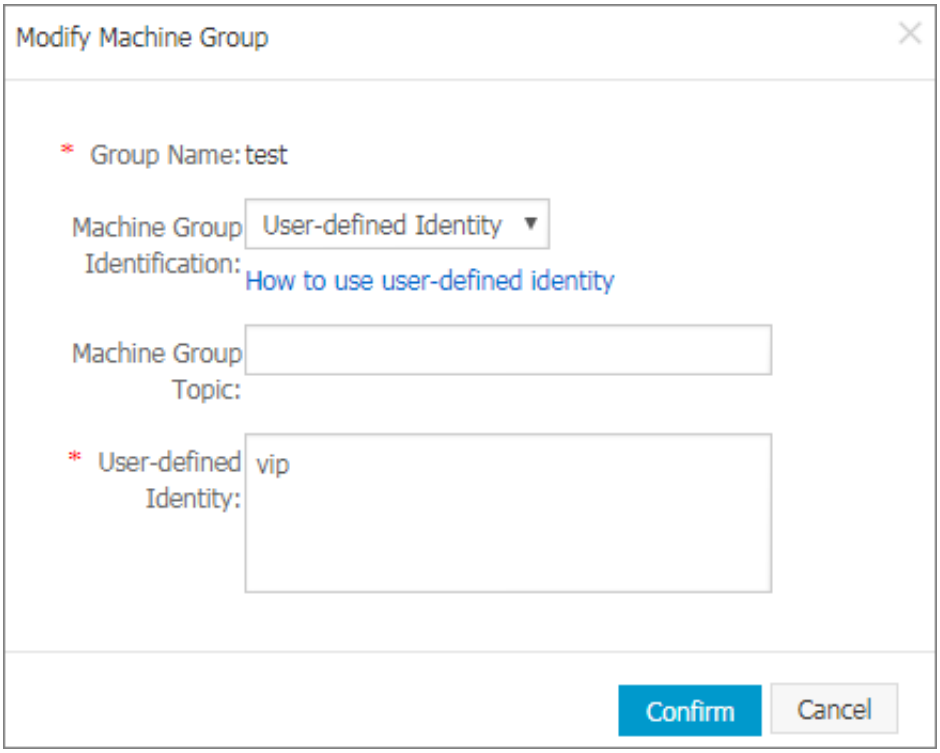
1. Log on to the Log Service console.
2. On the Logstore List page, click Logtail Machine Group in the left-side navigation pane. The Machine Groups page appears.

   All machine groups under the project are displayed.
3. Click Modify at the right of the machine group.
4. Modify the configurations and then click Confirm.

> **Note:**

The machine group name cannot be modified.

Figure 3-9: Modify a Machine Group



View status

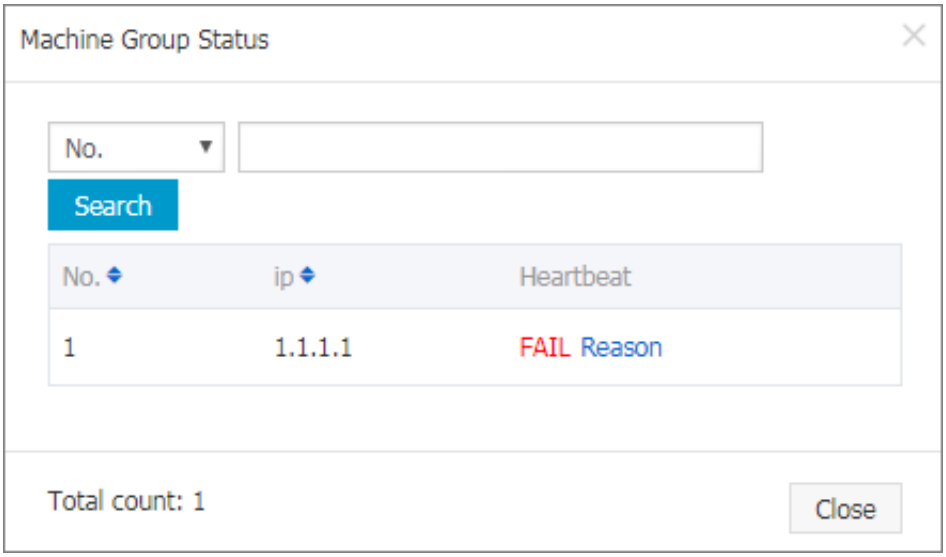
To verify that the Logtail client is successfully installed on all ECS instances in a machine group, view the heartbeat status of the Logtail client.

Note:
After a machine is added to a machine group, the change to the hearbeat status of the machine group takes about two minutes to take effect. Then, you can view the hearbeat status.

1. Log on to the Log Service console.
2. On the Project List page, click the project name. On the Logstore List page, click LogHub - Collect > Logtail Machine Group in the left-side navigation pane. The Machine Groups page appears.

3. Find the target machine group, and then click Machine Status.
   - If the Logtail client is installed on each ECS instance, and Logtail can normally communicate with Log Service, the heartbeat status is displayed as `OK`.
   - If Logtail cannot communicate with Log Service, the heartbeat status is displayed as `FAIL`. Furthermore, if the heartbeat status retains as `FAIL`, troubleshoot the exception by following the prompted messages on the page and the instructions provided by *What do I do if a Logtail machine has no heartbeat?*. If the problem persists, open a tick for support.

Figure 3-10: View the machine group status



Managing configurations

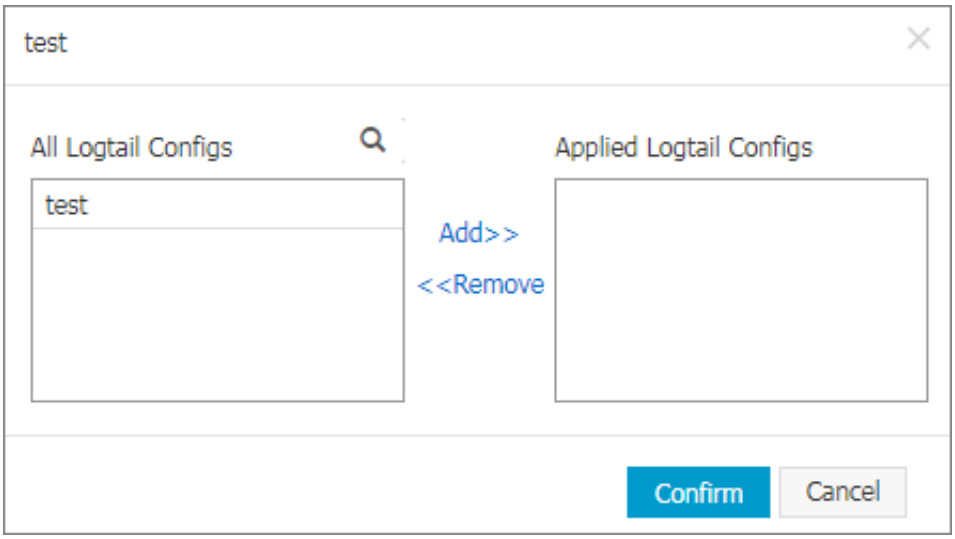
Log Service manages all the servers whose logs need to be collected by using machine groups. One important management item is the collection configuration of the Logtail client. For more information, see *Collect text logs* and *Syslog*. You can apply or delete a Logtail configuration to/from a machine group to decide what logs are collected, how the logs are parsed, and to which Logstore the logs are sent by the Logtail on each ECS instance.

1. Log on to the Log Service console.
2. On the Logstore List page, click Logtail Machine Group in the left-side navigation pane. The Machine Groups page appears.
3. Click Config at the right of the machine group.

4. Select the Logtail configuration and click Add or Remove to add or remove the configuration to/from the machine group.

   After a Logtail configuration is added, it is issued to the Logtail client on each ECS instance in the machine group. After a Logtail configuration is removed, it is removed from the Logtail client.

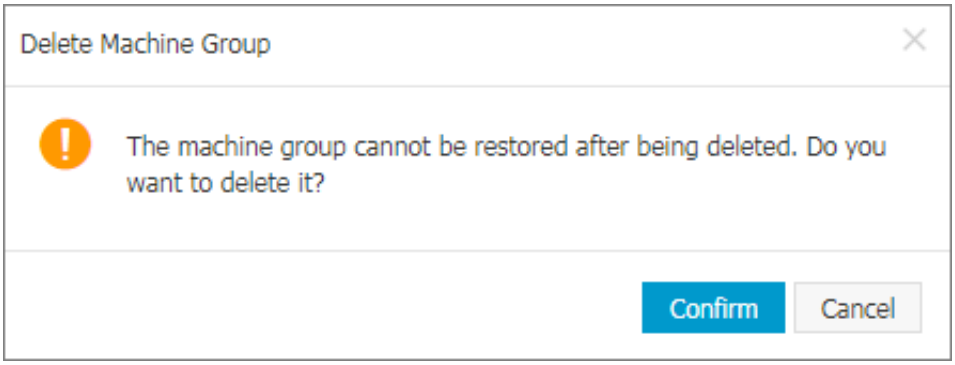
Figure 3-11: Managing Machine Group configurations



Delete a machine group

1. Log on to the Log Service console.
2. On the Project List page, click the project name. On the Logstore List page, click LogHub - Collect > Logtail Machine Group in the left-side navigation pane. The Machine Groups page appears.
3. Click Delete at the right of the machine group.
4. Click Confirm in the appeared dialog box.

Figure 3-12: Delete a machine group



3.5 Text logs

3.5.1 Collect text logs

The Logtail client helps Log Service users collect text logs from Elastic Compute Service (ECS) instances or local servers in the console.

Prerequisites

- You must install Logtail before collecting logs. For installation methods, see *Install Logtail in Linux* and *Install Logtail in Windows*.
- To collect logs from ECS instances or local servers, you must open ports 80 and 443.

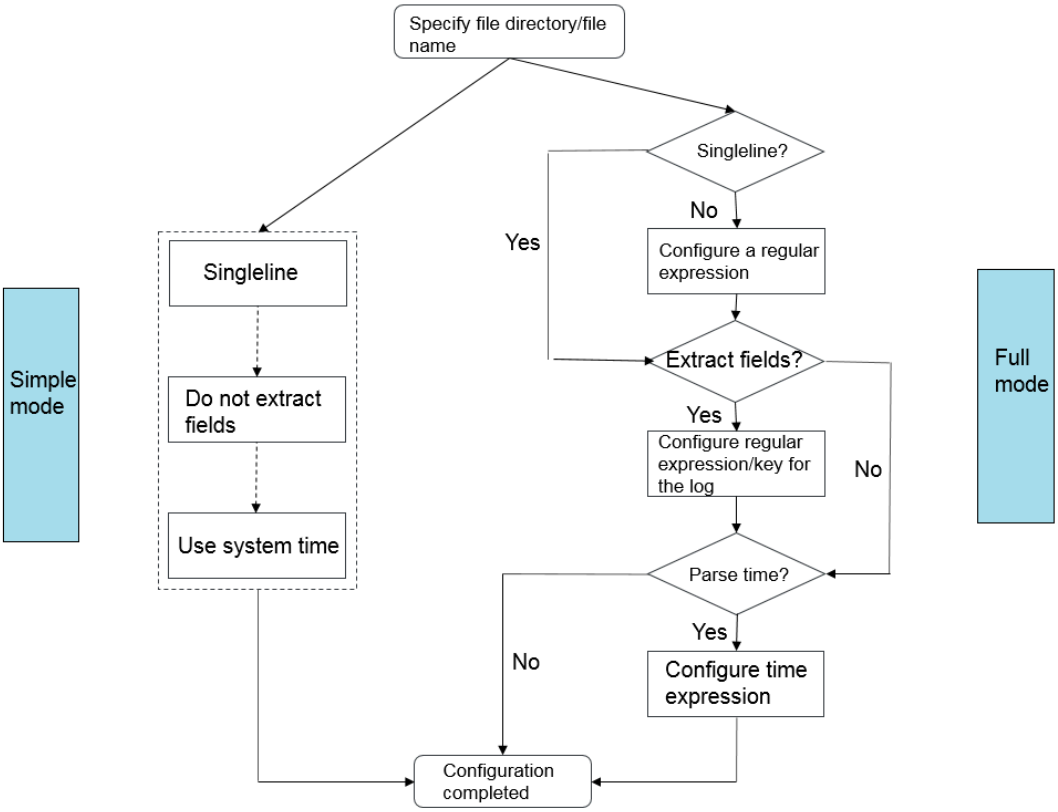
Limits

- A file can only be collected using one configuration. To collect a file with multiple configurations, we recommend you use the soft link. For example, to collect files under `/ home / log / nginx / log` with two configurations, you can use the original path for one configuration, and run the command `ln - s / home / log / nginx / log / home / log / nginx / link_log` to create a soft link of this folder, and then use the soft link path for the other configuration.
- For more information about operating systems supported by the Logtail client, see *Overview*.
- The ECS instances of the classic network or Virtual Private Cloud (VPC) and the Log Service project must belong to the same region. If your source data is transmitted by Internet (similar to IDC), you can select the region that the Log Service resides in based on the region description.

Configuration process of log collection

The following are simple mode and full mode examples. The configuration process as follows:

Figure 3-13: Log collection configuration process



Log collection modes

Logtail supports simple mode, delimiter mode, JSON mode, full mode, and other log collection methods.

- Simple mode

  Currently, simple mode is the single-line mode. By default, one line of data is a log, and two logs are separated by a line break in the log file. The system does not extract log fields (that is, the regular expression (.*) by default), and uses the current server system time as the generated log time. To configure more settings, you can change the configuration to full mode to adjust the settings. For

more information on how to change the Logtail configuration, see *Create a Logtail configuration*.

In the simple mode, specify the file directory and file name. Then, the Logtail collects logs by each line and uses the system time.

- Delimiter mode

  Logtail can collect delimiter logs through the delimiter mode. For more information, see *Delimiter logs*.

- JSON mode

  You can select JSON mode to collect *JSON logs*.

- Full mode

  To configure more personalized field extraction settings for log contents (such as cross-line logs and field extraction), select Full Mode.

  Log Service provides a log sample-based regular expression generation function in the data collection wizard. However, multiple manual tests to fit the log samples are required because of the different log samples. For more information about how to test the regular expressions, see *How do I test regular expressions?*

Procedure

1. Click Project name to enter the Logstore List.
2. Select Logstore, and click the Wizard at the right side of the Logstore.
3. Select the data source.

   Select Text under Other Sources and then click Next to go to the Configure Data Source step.
4. Specify the Configuration Name.

   The configuration name can be 3–63 characters long, contain lowercase letters , numbers, hyphens (-), and underscores (_), and must begin and end with a lowercase letter or number.

    Note:
   The configuration name cannot be modified after the configuration is created.

5. Specify the log directory and the file name.

   The directory structure must be a full path or a path that contains wildcards.

   > Note:
   > Only `*` and `?`can be used as wildcards in the directory.

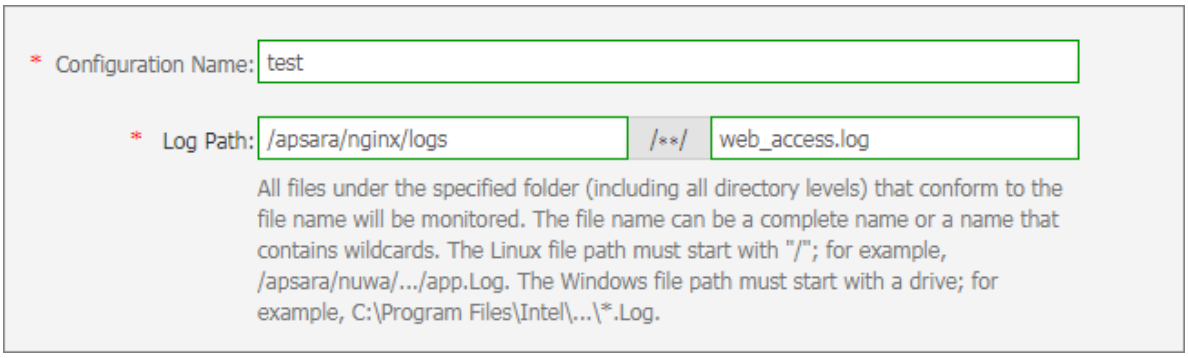
   The log file name must be a complete file name or a name that contains wildcards. For the rules of file names, see *Wildcard matching*.

   The search mode of log files is the multi-level directory matching mode, namely , under the specified folder (including directories of all levels), all the files that conform to the file name can be monitored.

   - `/ apsara / nuwa / … /*. log` means the files whose suffix is `. log` and exist in the `/ apsara / nuwa` directory (including its recursive subdirectories).
   - `/ var / logs / app_ * … /*. log *` means the files whose file name contains `. log` and exist in all of the directories that conform to the `app_ *` mode (including their recursive subdirectories) under the `/ var / logs` directory.

   > Note:
   > A file can only be collected by one configuration.

   Figure 3-14: Specify the Directory and file name



6. Set collection mode. The following uses the full mode as an example.
   a. Enter the Log Sample.

      The purpose of providing a log sample is facilitating the Log Service console in automatically extracting the regex matching mode in logs. Be sure to use a log from the actual environment.

   b. Disable Singleline.

      By default, the single-line mode is used, that is, two logs are separated by a line break. To collect cross-line logs (such as Java program logs), you must disable Singleline and then configure the Regular Expression.

   c. Modify. the Regular Expression.

      You can select to automatically generate the regular expression or manually enter the regular expression. After entering the log sample, click Auto Generate to automatically generate the regular expression. If failed, you can switch to the manual mode to enter the regular expression for verification.

   d. Enable Extract Field.

      To analyze and process fields separately in the log content, use the Extract Field function to convert the specified field to a key-value pair before sending it to Log Service. Therefore, you must specify a method for parsing the log content, that is, a regular expression.

      The Log Service console allows you to specify a regular expression for parsing the log content in two ways. The first option is to automatically generate a regular expression through simple interactions. You can select the field to be extracted in the log sample and then click Generate RegEx to automatically generate the regular expression in the Log Service console.

      In this way, you can generate the regular expression without writing it on your own. You can also manually enter a regular expression. Click Manually Input Regular Expression to switch to the manual input mode. After entering the regular expression, click Validate to validate whether or not the entered regular

expression can parse and extract the log sample. For more information, see *How do I test regular expressions?*

No matter the regular expression for parsing the log content is automatically generated or manually entered, you must name each extracted field, that is, set keys for the fields.

Figure 3-15:



e. Set Use System Time.

Default settings Use System Time is set by default. If disabled, you must specify a certain field (value) as the time field during field extraction and name this field `time`. After selecting a `time` field, you can click Auto Generate

in Time Format to generate a method to parse this time field. For more information on log time format, see *Text logs - Configure time format*.

f. Enable Drop Failed to Parse Logs as needed.

This option specifies whether to upload the logs with parsing failure to Log Service.

When this option is enabled, the logs with parsing failure will not be uploaded to Log Service. When the option is disabled, the raw log will be uploaded to Log Service when log parsing fails. The key of the raw log is `__raw_log_ _` and the value is the log content.

7. (Optional) Set Advanced Options as needed and click Next.

If you have no special requirements, retain the default settings.

Configuration item	Desceiption
Upload Original Log	Select whether or not to upload the original log. If enabled, the new field is added by default to upload the original log.
Topic Generation Mode	· Null - Do not generate topic: The default option, which indicates to set the topic as a null string and you can query logs without entering the topic. · Machine Group Topic Attributes: Used to clearly differentiate log data generated in different frontend servers. · File Path Regular: With this option selected, you must enter the Custom RegEx to use the regular expression to extract contents from the path as the topic. Used to differentiate log data generated by users and instances. Used to differentiate log data generated by users and instances.
Custom RegEx	After selecting File Path Regular as Topic Generation Mode, you must enter your custom regular expression.
Log File Encoding	· utf8: Use UTF-8 encoding. · gbk: Use GBK encoding.
Maximum Monitor Directory Depth	Specify the maximum depth of the monitored directory when logs are collected from the log source, that is, at most how many levels of logs can be monitored. The range is 0–1000, and 0 indicates to only monitor the current directory level.

Configuration item	Desceiption
Timeout	A log file has timed out if it does not have any update within a specified time. You can configure the following settings for Timeout. · Never Time out: Specify to monitor all log files persistent ly and the log files never time out. · 30 minute timeout: A log file has timed out and is not monitored if it does not have any update within 30 minutes.
Filter Configurat ion	Only logs that completely conform to the filter conditions can be collected. For example: · collect logs that conform to a condition: `Key : level Regex : WARNING | ERROR` indicates to only collect logs that are in the WARNING or ERROR level. · ***filter logs that do not conform to a condition*** : - `Key : level Regex :^(?!. *( INFO | DEBUG ))`, indicates to not collect logs whose level is INFO or DEBUG. - `Key : url Regex :. *^(?!.*( healthchec k )). *`, indicates to filter logs with healthcheck in the url. Such as logs in which key is url and value is `/ inner / healthchec k / jiankong . html` will not be collected. For similar examples, see*regex-exclude-word* and *regex-exclude-pattern*.

8. Click Next after completing the configurations.

   If you have not created a machine group, you must create one first. For how to create a machine group, see Create a machine *Create a machine group with an IP address as its identifier*group.

   Note:

- It takes up to three minutes for the Logtail configuration to take effect, so be patient.
- To collect IIS access logs, see *Use Logstash to collect IIS logs*.
- After creating the Logtail configuration, you can view the Logtail configuration list, modify the Logtail configuration, or delete the Logtail configuration. For more information, see *Create a Logtail configuration*.

Figure 3-16: Applying the configuration to the machine group



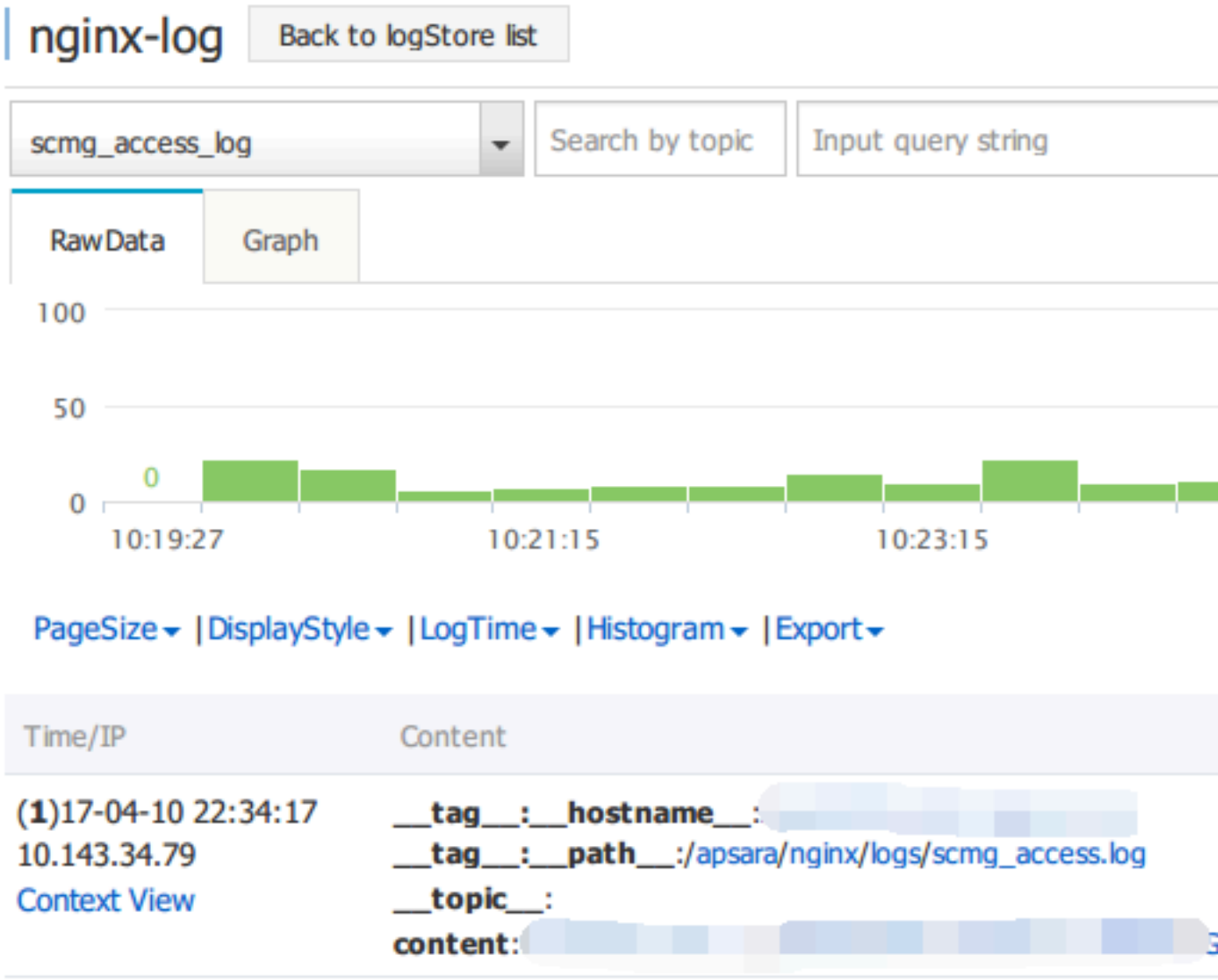
Log Service starts to collect logs after completing the configurations.

Subsequent operations

After completing the preceding configurations, you can configure the Search, Analysis, and Visualization and Shipper & ETL as instructed on the page.

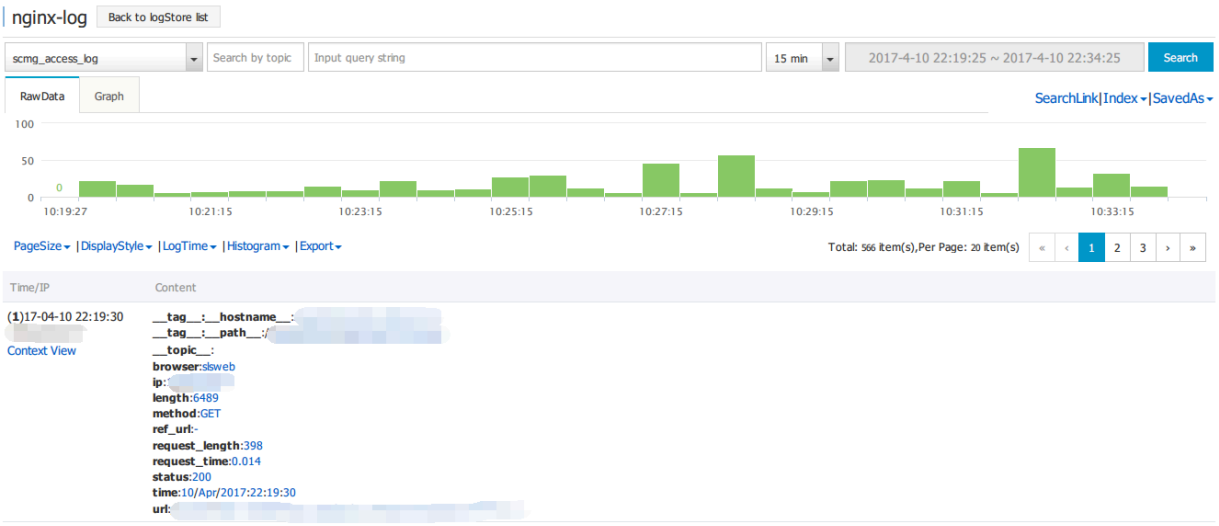
Logs collected to Log Service in the simple mode are as follows. All the contents of each log are displayed under the key named content.

Figure 3-17: Preview



Logs collected to Log Service in the full mode are as follows. The contents of each log are collected to Log Service according to the configured key-value.

Figure 3-18: Preview



Logtail configuration items

You must complete the configuration items when configuring Logtail. The descriptions and limits of the commonly used configuration items are as follows.

Configuration item	Description
Log path	Make sure that the log monitoring directory and the log file name match with the files on the machine. The directory does not support fuzzy match and must be set to an absolute path, while the log file name supports fuzzy match. The path that contains wildcards can match with directories of multiple levels, that is, under the specified folder (including directories of all levels), all the files that conform to the file name can be monitored.
Log file name	The name of the file from which logs are collected, which is case-sensitive and can contain wildcards, for example, `*.log`. The file name wildcards in Linux include *, “?”, and [···].

Local Storage	Whether or not to enable the local cache to temporarily store logs that cannot be sent because of short-term network interruption.
First-line log header	Specifies the starting header of a multiline log by specifying a regular expression. Lines cannot be used to separate individual logs when multiline log is collected (such as the stack information in application logs). In this case, you must specify the start line of a multi-line log. When this line is discovered, this indicates the last log has ended and a new log has begun. Therefore, you must specify a matching rule for the starting header, that is, a regular expression here.
Log parsing expression	Defines how to extract a piece of log information and convert it to a log format supported by Log Service. The user must specify a regular expression to extract the required log field informatio n and define the name of each field to be extracted.
Log time format	Defines how to parse the time format of the timestamp string in log data. For more information, see *Text logs - Configure time format*.

Writing method of logs

In addition to using Logtail to collect logs, Log Service also provides APIs and SDKs to help you write logs.

- APIs to write logs

  Log Service provides RESTful APIs to help you write logs. You can use the *PutLogs* API to write data. For more information on a complete API reference, see *Overview*.

- Use SDKs to write logs

  In addition to APIs, Log Service also provides SDKs in multiple languages (Java, .NET, PHP, and Python) to help you write logs. For more information on a complete SDK reference, see SDK reference *Overview* .

3.5.2 Configure and parse text logs

Specify log line separation method

A full access log is typically a row by line, such as the nginx's access log, each log is split with line breaks. For example, the following two access logs:

```
10 . 1 . 1 . 1  - - [ 13 / Mar / 2016 : 10 : 00 : 10  + 0800 ] " GET
 /  HTTP / 1 . 1 "  0 . 011   180   404   570  "-" " Mozilla / 4 . 0
( compatible ;  MSIE   6 . 0 ;  Windows   NT   5 . 1 ;  360se )"
10 . 1 . 1 . 1  - - [ 13 / Mar / 2016 : 10 : 00 : 11  + 0800 ] " GET
 /  HTTP / 1 . 1 "  0 . 011   180   404   570  "-" " Mozilla / 4 . 0
( compatible ;  MSIE   6 . 0 ;  Windows   NT   5 . 1 ;  360se )"
```

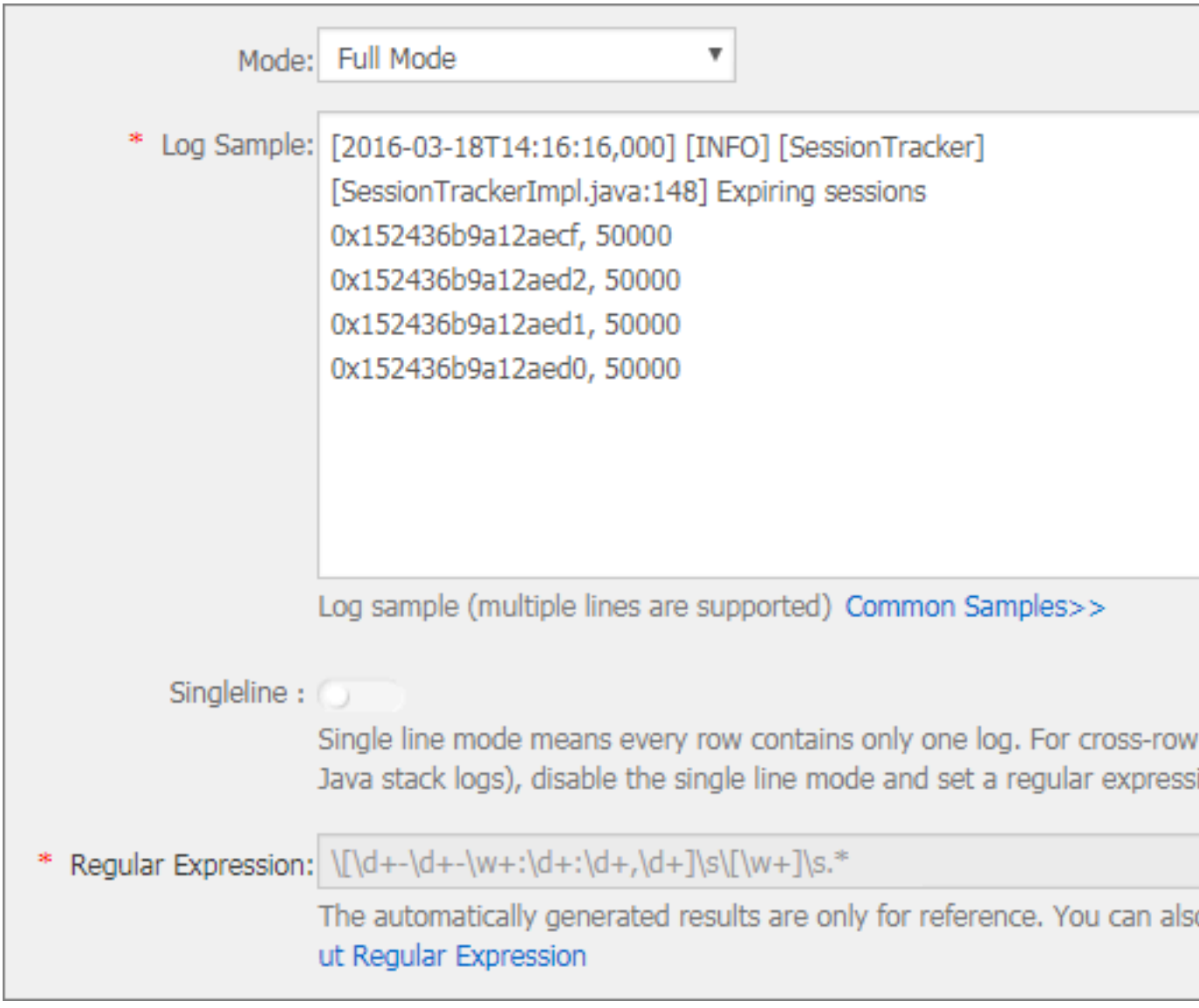
For Java applications, a program log usually spans several lines. The characteristic log header is used to separate two logs. For example, see the following Java program log:

```
[ 2016 - 03 - 18T14 : 16 : 16 , 000 ] [ INFO ] [ SessionTra  cker ] [
 SessionTra  ckerImpl . java : 148 ]  Expiring   sessions
 0x152436b9  a12aecf ,  50000
 0x152436b9  a12aed2 ,  50000
 0x152436b9  a12aed1 ,  50000
```

```
0x152436b9  a12aed0 ,  50000
```

The preceding Java log has a starting field in the time format. The regular expression is \[\d+-\d+-\w+:\d+:\d+,\d+]\s.* *. You can complete the configurations in the console as follows.

Figure 3-19: Full mode parsing Regular Expression



Extract log fields

According to the *Log Service data models*, a log contains one or more key-value pairs. To extract specified fields for analysis, you must set a regular expression. If log content does not need to be processed, the log can be considered as a key-value pair.

For the access log in the previous example, you can choose to extract a field or not.

- When fields are extracted

  Regular expression: `(\ S +)\ s -\ s -\ s \[(\ S +)\ s [^]]+]\ s "(\ w +).*`, Extracted contents: `10 . 1 . 1 . 1` , `13 / Mar / 2016 : 10 : 00` and `GET` .

- When fields are not extracted

  Regular expression: `(. *)`, Extracted contents: `10 . 1 . 1 . 1 - - [ 13 / Mar / 2016 : 10 : 00 : 10 + 0800 ] " GET / HTTP / 1 . 1 " 0 . 011 180 404 570 "-" " Mozilla / 4 . 0 ( compatible ; MSIE 6 . 0 ; Windows NT 5 . 1 ; 360se )""`

Specify log time

According to the Log Service data models, a log must have a time field in UNIX timestamp format. Currently, the log time can be set to the system time when Logtail collects the log or the time field in the log content.

For the access log in the previous example:

- Extract the time field in the log content `Time : 13 / Mar / 2016 : 10 : 00 : 10` Time expression: `% d /% b /% Y :% H :% M :% S`
- The system time when the log is collected Time: Timestamp when the log is collected.

3.5.3 Text logs - Configure time format

As described in the core concepts of Log Service, each log in Log Service has a timestamp when this log happened. Logtail must extract the timestamp string of each log and parse it into a timestamp when collecting logs from your log files. Therefore, you must specify the timestamp format of the log for Logtail.

In Linux, Logtail supports all time formats provided by the strftime function. Logtail can parse and use the timestamp strings that can be expressed in the log formats defined by the strftime function.

In reality, the timestamp strings of logs have multiple formats. To make configuration easier, Logtail supports the following common log time formats.

Format	Meaning	Example
%a	The abbreviation of a day in a week.	Fri

Format	Meaning	Example
%A	The full name of a day in a week.	Friday
%b	The abbreviation of a month.	Jan
%B	The full name of the month.	January
%d	The day of the month in decimal format [01,31].	07, 31
%h	The abbreviation of a month. Same as `% b`.	Jan
%H	The hour in 24-hour format.	22
%I	The hour in 12-hour format.	11
%m	The month in decimal format.	08
%M	The minute in decimal format [00, 59].	59
%n	A line break.	A line break
%p	AM or PM locally.	AM/PM
%r	Time in 12-hour format, which is equivalent to `% I :% M :% S % p`.	11:59:59 AM
%R	Time expressed in hour and minute, which is equivalent to `% H :% M`.	23:59
% S	The second in decimal format [00, 59].	59
%t	Tab character.	Tab character
%y	The year without century in decimal format [00, 99].	04; 98
%Y	The year in decimal format.	2004; 1998
%C	The century in decimal format [00, 99].	16

Format	Meaning	Example
%e	The day of the month in decimal format [1, 31]. A single digit is preceded by a space.	7, 31
%j	The day of the year in decimal format [00, 366].	365
%u	The day of the week in decimal format [1, 7]. 1 represents Monday.	2
%U	The week number of the year (Sunday as the first day of the week) [00, 53].	23
%V	The week number of the year (Monday as the first day of the week) [01, 53]. If the week at the beginning of January has no less than four days, this week is the first week of the year. Otherwise, the next week is considered as the first week of the year.	24
%w	The day of the week in decimal format [0, 6]. 0 represents Sunday.	5
%W	The week number of the year (Monday as the first day of the week) [00,53].	23
%c	Standard date and time representation.	To specify more information such as long date and short date, we recommend that you use the preceding supported formats for more precise expression.

Format	Meaning	Example
%x	Standard date representation.	To specify more information such as long date and short date, we recommend that you use the preceding supported formats for more precise expression.
%X	Standard time representation.	To specify more information such as time, we recommend that you use the preceding supported formats for more precise expression.
%s	Unix timestamp.	1476187251

3.5.4 Import history logs

Logtail only collects incremental logs by default. If you want to import history logs, use the history log importing feature of Logtail.

Prerequisites

- Your Logtail must be v0.16.15 (Linux) or v1.0.0.1 (Windows) and later. If your Logtail is an earlier version, you must upgrade it to the latest version.
- Target history logs do not have to be covered by the collection configuration. If a log file has been collected by Logtail, Logtail will collect the log file again after you import history logs.
- The maximum interval between generating and importing local configurations is one minute.
- Due to the special action of loading local configurations, Logtail notifies you of this action by sending `LOAD_LOCAL _EVENT_ALA RM` to your server.
- If you want to import a large number of log files, you must modify the startup parameters for Logtail. That is, increase the upper limit of the Logtail CPU and memory. We commend that you increase the CPU to 2.0 or any other greater specification, and increase the memory to 512 MB or any other greater specification. For more information, see *Configure startup parameters*.

Context

Logtail collects logs based on the events that are detected by listening on or performing round robin for log modifications. Logtail can also load local configurat ions, and trigger log collections. Logtail collects history logs by loading local configurations.

You must perform the operations for importing history logs in the Logtail installation directory, which varies by operating systems.

- Linux operating system: `/ usr / local / ilogtail .`
- Windows operating system
  - 32-bit Windows operating system: `C :\ Program Files \ Alibaba \ Logtail`
  - 64-bit Windows operating system: `C :\ Program Files ( x86 )\ Alibaba \ Logtail`

Procedure

1. Create collection configurations

   Configure the collection and apply the configuration to the machine group. If you only want to configure the collection for importing history logs, you can set a collection directory that does not exist. For more information about the collection configuration, see *Collect text logs*.

2. Gets the configuration unique identity.

   Obtain a unique identifier for the configuration from the `user_log_c onfig . json` file in the Logtail installation directory. For more information, see *Logtail installation directory*. For the Linux operating system, run the `grep` command in this directory. For the Windows operating system, use a tool (for example, the text tool) to open the file.

   To view the identifier in the Linux operating system, do the following:

```
grep  "##" / usr / local / ilogtail / user_log_c  onfig . json  |
awk  '{ print  $ 1 }'
                  "## 1 . 0 ## log - config - test $ multi "
                  "## 1 . 0 ## log - config - test $ ecs - test "
                  "## 1 . 0 ## log - config - test $ metric_sys
tem_test "
                  "## 1 . 0 ## log - config - test $ redis -
status "
```

3. Add local events.

   Save local events to JSON file `local_even t . json` in the Logtail installation directory (for more information, see *Logtail installation directory*). You must save the local events by using the following format:

```
[
  {
    " config " : "${ your_confi  g_unique_i  d }",
    " dir " : "${ your_log_d  ir }",
    " name " : "${ your_log_f  ile_name }"
   },
  {
   ...
   }
   ...
]
```

   - Configuration items

Configuration items	Description:	Examp
Config	Indicates the configuration unique identifier that is obtained in step 2.	##1.0#
dir	Indicates the folder where logs are located. Note: The folder cannot end in `/`.	`/ dat`
name	Indicates a log name that support wildcards.	`acce` `log`

   Note:
   To prevent Logtail from loading invalid JSON files, we recommend that you save local event configurations to a temporary file, and after editing the temporary file, copy the content to `local_even t . json` .

   - Configuration example

     For the Windows operating system, directly use a tool (for example, the text tool) to modify the file `local_even t . json` . For the Linux operating system, add local events as follows:

```
$  cat  / usr / local / ilogtail / local_even  t . json
[
  {
```

```
    " config ": "## 1 . 0 ## log - config - test $ ecs - test ",
    " dir ": "/ data / log ",
    " name ": " access . log *"
  },
  {
    " config ": "## 1 . 0 ## log - config - test $ tmp - test ",
    " dir ": "/ tmp ",
    " name ": " access . log . 2017 - 08 - 09 "
  }
]
```

- How can I check whether Logtail has loaded the configuration?

  After you save local file `local_even t . json`, Logtail loads this local configuration file to the memory within one minute, and clears the content in `local_even t . json`.

  You can check whether Logtail has read local events by following these methods:

  - Check whether the content in `local_even t . json` has been cleared. If cleared, Logtail has read the local configurations.
  - Check whether the file *`ilogtail . LOG`* in the Logtail installation directory (for more information, see *Logtail installation directory*) includes `process local event` keywords. If the content in `local_even t . json` has been cleared, but these keywords cannot be found, the local configuration file may be invalid and has been filtered out.
  - Search the *Query diagnosed errors* result for the `LOAD_LOCAL _EVENT_ALA RM` alarm.

- Logtail has loaded the configuration, but still cannot collect any data. How can I deal with this issue?

  This issue may be caused by the following reasons:

  - The configuration is invalid.
  - The `config` item is available in the local configuration.
  - The target log is not located in the specified path in the collection configuration.
  - The target log has been collected.

3.5.5 Set a log topic



Note:

No topic can be set for a syslog.

Topic generation modes

You can set a topic when using Logtail to collect logs or using APIs/SDKs to upload data. Currently, the following topic generation modes are supported in the console: Null - Do not generate topic, Machine Group Topic Attributes, and File Path Regular.

- Null - Do not generate topic

  The default log topic generation mode is Null - Do not generate topic when you configure Logtail to collect text logs in the console, that is, the topic is a null string, and you can query logs without entering a topic.

- Machine Group Topic Attributes

  Machine Group Topic Attributes mode is used to clearly differentiate log data generated in different servers. If the log data of different servers are stored in the same file path and file name, you can divide the machines into different machine groups to distinguish the log data of different servers by topic. To do this, set different topic attributes for different machine groups when creating machine groups, and set the Topic Generation Mode to Machine Group Topic Attributes . Apply the previously created Logtail configuration to those machine groups to complete the configuration.

  With this mode selected, Logtail uploads the topic attribute of the machine group where the current machine belongs as the topic name to Log Service when reporting data. When querying logs, you must specify a topic, that is, you must specify the topic attribute of the target machine group as the query condition.

- File Path Regular

  This mode is used to differentiate log data generated by users and instances. If service logs are stored under different directories based on users or instances but their subdirectories and log file names are the same, Log Service cannot clearly differentiate which user or instance generates the logs when collecting log files. In this case, you can set the Topic Generation Mode to File Path Regular, enter a regular expression of file paths, and set the topic as the instance name.

  With this mode selected, Logtail uploads the instance name as the topic name to Log Service when reporting data. Different topics are generated according to your

directory structure and configuration. You must specify the topic name as the instance name when querying logs.

For example, the following directory setting stores logs in different directories according to the users to which the logs belong:

```
/ logs
  | - / userA / serviceA
    | -  service . log
  | - / userB / serviceA
    | -  service . log
  | - / userC / serviceA
    | -  service . log
```

For files that are named `service . log` and stored in `/ logs` in th server, the users or instances from which the logs of these files were generated cannot be distinguished. You can set Topic Generation Mode as File Path Regular, and enter `\/(.*)\/ serviceA \/.*` to extract the instance names. This method can generate an unique topic for logs in each directory: userA, userB, and userC. Then, you can query a log by specifying its topic.

 Note:
You must escape `/` in the regular expression for a file directory.

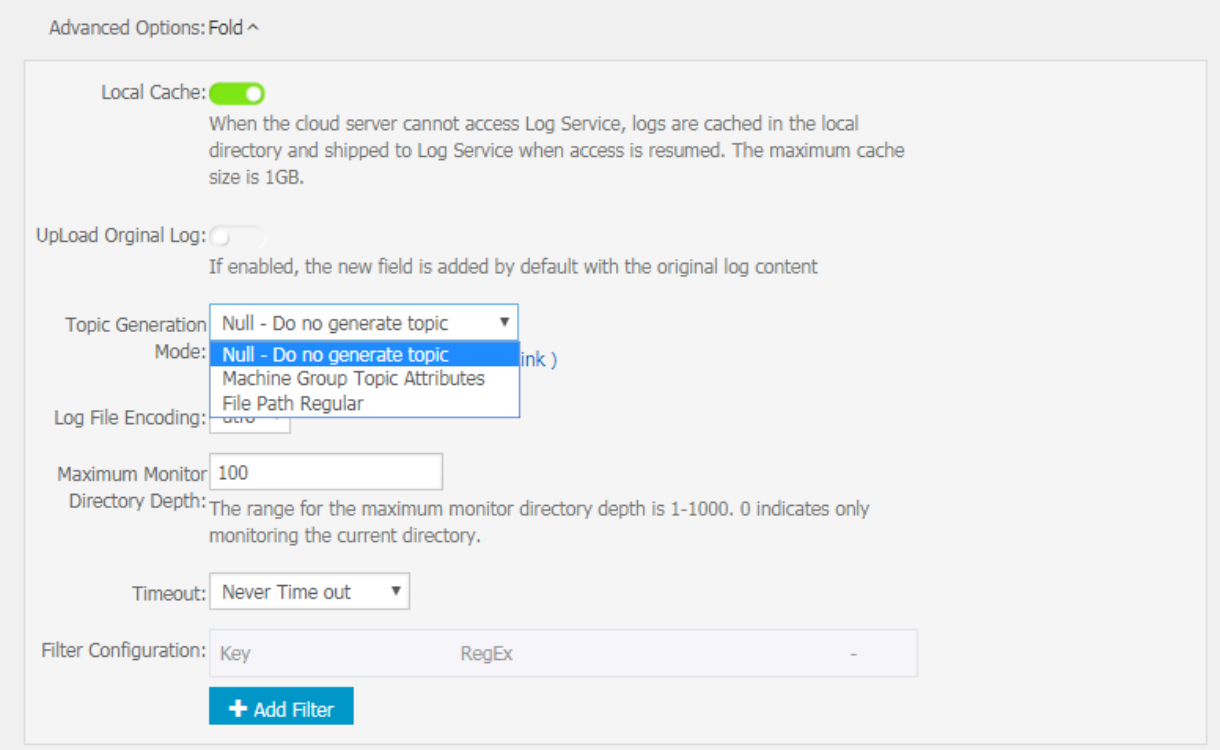
Set log topic

1. Configure Logtail in the console by following the *Collect text logs*.

   To set the topic generation mode to Machine Group Topic Attributes, configure the Machine Group Topic when creating/modifying a machine group.

2. Expand Advanced Options in the data import wizard and select a topic generation mode from the Topic Generation Mode drop-down list.

Figure 3-20: Set log topic



Modify log topic

To change the log topic generation mode, modify the Topic Generation Mode option directly in the data import wizard.

Note:
The modified configuration only applies to the data collected after the modification takes effect.

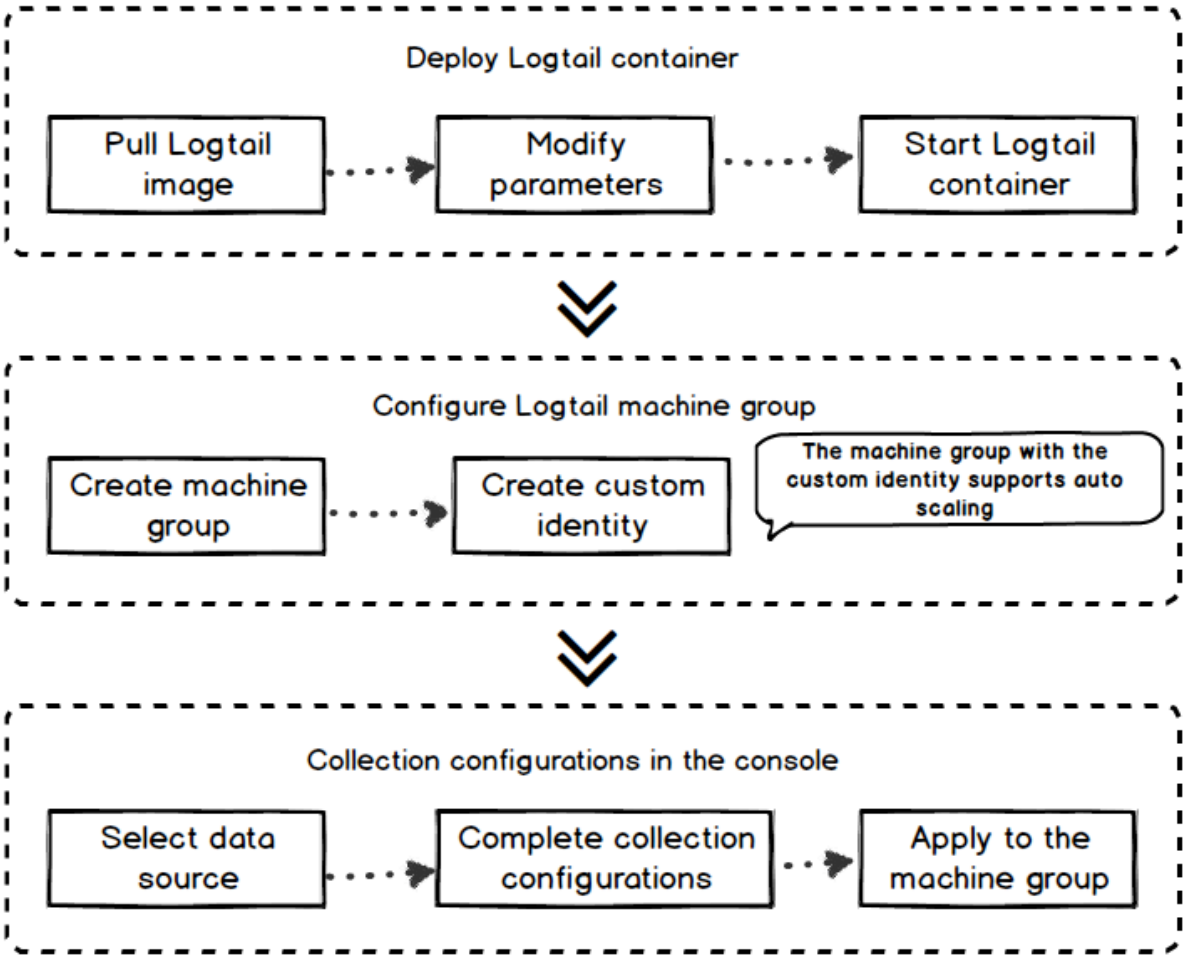
3.6 Container log collection

3.6.1 Collect standard Docker logs

Logtail supports collecting standard Docker logs and uploading these logs together with the container-related metadata information to Log Service.

Configuration process

Figure 3-21: Configuration process



1. Deploy a Logtail container.
2. Configure the Logtail container.

   In the logging service control panel, create a machine group with a custom ID. In the future this container will not need further O&M to expand or contract.
3. Create a new configuration for collection on the server side.

   Create collection configurations in the Log Service console. All the collection configurations are for the server side. No local configuration is needed.

Step 1. Deploy the Logtail container

1. **Pull the Logtail image.**

```
docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail
```

2. **Start the Logtail container.**

   Replace the following three parameters in the startup template: `${your_region_name}`, `${your_aliyun_user_id}`, and `${your_machine_group_user_define_id}`.

```
docker run -d -v /:/logtail_host:ro -v /var/run/docker.sock:/var/run/docker.sock --env
ALIYUN_LOG_TAIL_CONFIG=/etc/ilogtail/conf/${your_region_name}/ilogtail_config.json
--env ALIYUN_LOG_TAIL_USER_ID=${your_aliyun_user_id} --env
ALIYUN_LOG_TAIL_USER_DEFINED_ID=${your_machine_group_user_define_id} registry.cn-hangzhou.aliyuncs.com/log-service/logtail
```



Note:

Please perform any of the following configurations before the configuration parameters, otherwise, there may be an error in removing the other container, `container text file busy`。

- **Centos version 7.4 and later sets up FS. may_detach_mounts = 1, see Bug 1468249, bug 1441737, and issue 34538 for instructions.**

- Give logtail a `privileged` permission, and add `--privileged` to the startup parameter. For more information, see *docker run command* .

Parameter	Description
`${your_region_name}`	This parameter is determined by the region and network type of the project. Set the parameter to an appropriate value according to the network type. Valid values: - For the Internet: `region-internet`. For example, the value for the China (Hangzhou) region is `cn-hangzhou-internet`. - For Alibaba Cloud intranet: `region`. For example, the value for the China (Hangzhou) region is `cn-hangzhou`. In this parameter, region is a *Table 1*. Set it to the region to which the project belongs.
`${your_aliyun_user_id}`	User ID, replace it with the ID of your main Alibaba Cloud account. The user identification. Replace it with the ID of your Alibaba Cloud main account, which is in the string format. For how to check the ID, see 2.1 in *Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs*.
`${your_machine_group_user_define_id}`	The user-defined identity of your cluster machine group. If user-defined identity is not enabled yet, enable userdefined-id by following the corresponding steps in *Create an ID to identify a machine group*.

```
docker run -d -v /:/logtail_host:ro -v /var/run/
docker.sock:/var/run/docker.sock
--env ALIYUN_LOG_TAIL_CONFIG=/etc/ilogtail/conf/
cn_hangzhou_u/ilogtail_config.json --env
```

```
 ALIYUN_LOG  TAIL_USER_  ID = 1654218 ******-- env   ALIYUN_LOG
TAIL_USER_  DEFINED_ID = log - docker - demo   registry . cn -
hangzhou . aliyuncs . com / log - service / logtail
```



Note:

You can customize the startup parameter configurations of Logtail containers if the following conditions are met:

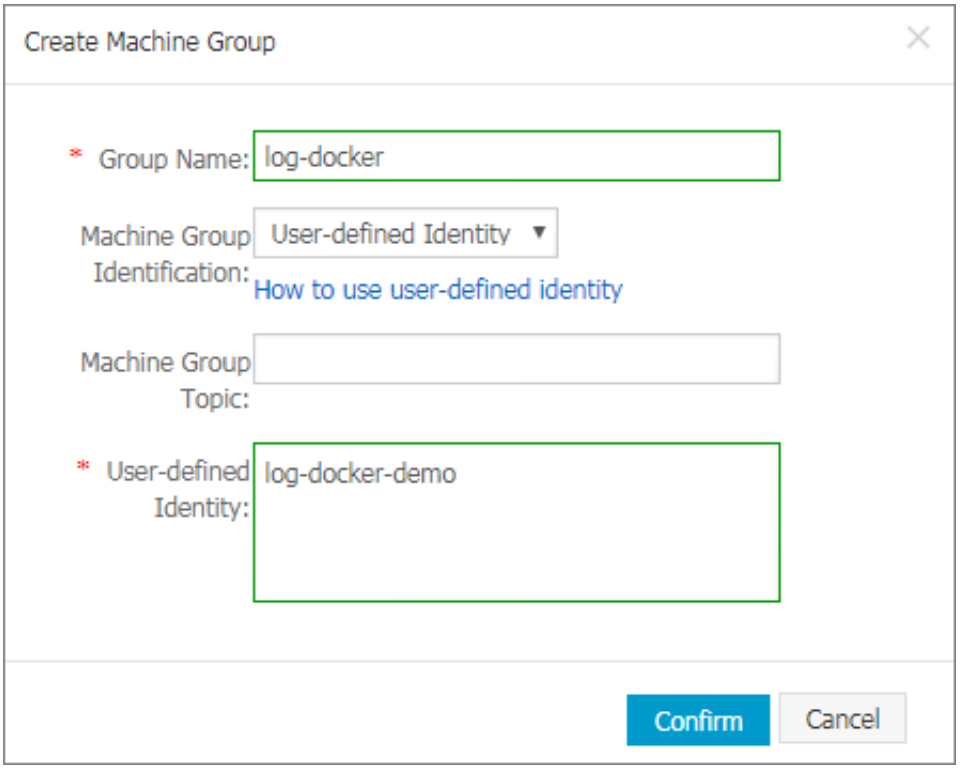
1. **You have the following three environment variables when starting the Logtail containers:** `ALIYUN_LOG  TAIL_USER_  DEFINED_ID` , `ALIYUN_LOG  TAIL_USER_  ID` , **and** `ALIYUN_LOG  TAIL_CONFI  G` .
2. **The domain socket of Docker is mounted to** `/ var / run / docker . sock` .
3. **To collect stdout, container logs, or host files, you must mount the root directory to the Logtail container directory** `/ logtail_ho  st` .
4. **If there is a error log** `The  parameter  is  invalid  :  uuid = none` **in Logtail logs** *`/ usr / local / ilogtail / ilogtail . LOG`* **, create a** *`product_uu  id`* **file on the host machine, then you enter any legal UUID (for example,** `169E98C9 - ABC0 - 4A92 - B1D2 - AA6239C0D2  61` **), and mount the file to the** *`/ sys / class / dmi / id / product_uu  id`* **path of the Logtail container.**

Step 2. Configure machine group

1. **Activate Log Service, and create Project and Logstore. For more information, see** *Preparation*.
2. **Click** *Create a machine group with an IP address as its identifier* **on the Machine Groups page in the Log Service console.**

3. Select User-defined Identity from the Machine Group Identification drop-down list. Enter the `ALIYUN_LOG_TAIL_USER_DEFINED_ID` configured in the previous step in the User-defined Identity field.

   Figure 3-22: Configuring the machine group



Click Confirm to create the machine group. One minute later, click Machine Status at the right of the machine group on the Machine Groups page to view the heartbeat status of the deployed Logtail container. For more information, see View status in *Manage a machine group*.

Step 3. Create collection configurations

Create Logtail collection configurations in the console as needed. For how to create collection configurations, see:

- *Container text log (recommended)*
- *Container standard output (recommended)*
- *Host text file*

By default, the root directory of a host is mounted in the `/logtail_host` directory of the Logtail container. You need to prefix the configuration path with /logtail_host. For example, to collect data in the `/home/logs/app_log/`

directory of the host, you must set the log path on the configuration page to `/ logtail_ho st / home / logs / app_log /`.

- *Syslog*

Other operations

- **Check the running status of the Logtail container**

  You can run the `docker exec ${ logtail_co ntainer_id } / etc / init . d / ilogtaild status` command to check the running status of Logtail.

- **Check version number, IP, and startup time of Logtail**

  You can run the `docker exec ${ logtail_co ntainer_id } cat / usr / local / ilogtail / app_info . json` command to check the information related to Logtail.

- **Check Logtail running logs**

  Logtail running logs are stored in the `/ usr / local / ilogtail /` directory. The file name is `ilogtail . LOG` . The rotation file is compressed and stored as `ilogtail . LOG . x . gz` .

  For example:

```
[ root @ iZbp17enxc 2us3624wex h2Z ilogtail ]# docker exec
 a287de895e 40 tail - n 5 / usr / local / ilogtail / ilogtail
 . LOG
[ 2018 - 02 - 06 08 : 13 : 35 . 721864 ] [ INFO ] [ 8 ] [ build /
 release64 / sls / ilogtail / LogtailPlu gin . cpp : 104 ] logtail
   plugin Resume : start
[ 2018 - 02 - 06 08 : 13 : 35 . 722135 ] [ INFO ] [ 8 ] [ build /
 release64 / sls / ilogtail / LogtailPlu gin . cpp : 106 ] logtail
   plugin Resume : success
[ 2018 - 02 - 06 08 : 13 : 35 . 722149 ] [ INFO ] [ 8 ] [ build /
 release64 / sls / ilogtail / EventDispa tcher . cpp : 369 ] start
   add existed check point events , size : 0
[ 2018 - 02 - 06 08 : 13 : 35 . 722155 ] [ INFO ] [ 8 ] [ build /
 release64 / sls / ilogtail / EventDispa tcher . cpp : 511 ] add
   existed check point events , size : 0 cache size : 0
 event size : 0 success count : 0
[ 2018 - 02 - 06 08 : 13 : 39 . 725417 ] [ INFO ] [ 8 ] [ build /
 release64 / sls / ilogtail / ConfigMana ger . cpp : 3776 ] check
   container path update flag : 0 size : 1
```

  The container stdout is not of reference significance, ignore the following stdout output:

```
start umount useless mount points , / shm $|/ merged $|/
mqueue $
```

```
 umount : / logtail_ho  st / var / lib / docker / overlay2 /
 3fd0043af1  74cb0273c3  c7869500fb  e2bdb95d13  b1e110172e
 f57fe840c8  2155 / merged :  must   be   superuser   to   unmount
 umount : / logtail_ho  st / var / lib / docker / overlay2 /
 d5b10aa193  99992755de  1f85d25009  528daa749c  1bf8c16edf
 f44beab6e6  9718 / merged :  must   be   superuser   to   unmount
 umount : / logtail_ho  st / var / lib / docker / overlay2 /
 5c3125dadd  acedec29df  72ad0c52fa  c800cd56c6  e880dc4e8a
 640b1e16c2  2dbe / merged :  must   be   superuser   to   unmount
......
 xargs :  umount :  exited   with   status   255 ;  aborting
 umount   done
 start   logtail
 ilogtail   is   running
 logtail   status :
 ilogtail   is   running
```

- **Restart Logtail**

  See the following example to restart Logtail:

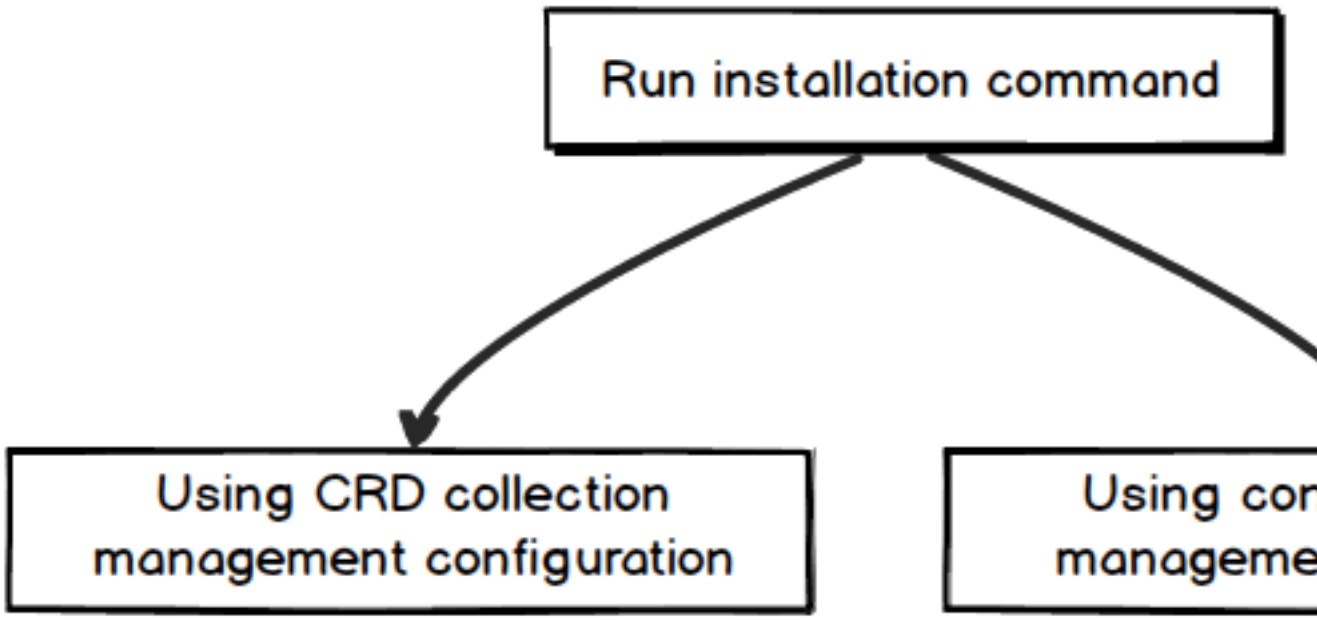
```
[ root @ iZbp17enxc  2us3624wex  h2Z   ilogtail ]#  docker   exec
 a287de895e  40  / etc / init . d / ilogtaild   stop
 kill   process   Name :  ilogtail   pid :  7
 kill   process   Name :  ilogtail   pid :  8
 stop   success
[ root @ iZbp17enxc  2us3624wex  h2Z   ilogtail ]#  docker   exec
 a287de895e  40  / etc / init . d / ilogtaild   start
 ilogtail   is   running
```

3.6.2 Kubernetes log collection process

Log Service uses Logtail to collect Kubernetes cluster logs and manages collection configuration through custom resource definition (CRD). This document describes how to install and use Logtail to collect Kubernetes cluster logs.

Configuration process

Figure 3-23: Configuration process



1. Run the installation command to install the alibaba-log-controller Helm package.
2. Choose the CRD or console to manage collection configuration as required.

Step 1 Install Logtail

Installation of managed Kubernetes clusters in Container Service

For more information about how to install Logtail in a managed Kubernetes cluster, see the Manually install Log Service components section in *Use Log Service to collect Kubernetes cluster logs*.

Installation of Kubernetes on Alibaba Cloud Container Service

Procedure

1. Log on to the master node of your Alibaba Cloud Container Service Kubernetes. For more information, see *Access Kubernetes clusters by using SSH*.
2. Replace `${ your_k8s_c luster_id }` in the following command with your Kubernetes cluster ID and run the command:

```
wget   http :// logtail - release - cn - hangzhou . oss - cn -
hangzhou . aliyuncs . com / kubernetes / alicloud - log - k8s -
install . sh  - O   alicloud - log - k8s - install . sh ;  chmod
```

```
    744  ./ alicloud - log - k8s - install . sh ;  sh  ./ alicloud -
  log - k8s - install . sh  ${ your_k8s_c  luster_id }
```

After installation, Log Service automatically creates a Log Service project in the same region of your Kubernetes cluster. The name of the created project is `k8s - log -${ your_k8s_c  luster_id }`. Under the project, machine group `k8s - group -${ your_k8s_c  luster_id }}` is created automatically.



Note:

- The `config - operation - log` Logstore, which is used to store operational logs of alibaba-log-controller, is automatically created under the `k8s - log - ${ your_k8s_c  luster_id }` project. You cannot delete this Logstore because it is required for alibaba-log-controller troubleshooting.
- To store the collected logs in an existing project, run the `sh ./alicloud-log-k8s-install.sh${your_k8s_cluster_id}${your_project_name}` installation command and confirm that the project belongs to the region where your Kubernetes cluster is deployed.

Installation example

After successful execution, the following information is displayed:

```
[ root @ iZbp ****** biaZ  ~]#  wget   http :// logtail - release -
 cn - hangzhou . oss - cn - hangzhou . aliyuncs . com / kubernetes
 / alicloud - log - k8s - install . sh  - O   alicloud - log - k8s
 - install . sh ;  chmod   744  ./ alicloud - log - k8s - install
 . sh ;  sh  ./ alicloud - log - k8s - install . sh   c12ba20
 ************** 86939f0b
....
....
....
 alibaba - cloud - log / Chart . yaml
 alibaba - cloud - log / templates /
 alibaba - cloud - log / templates / _helpers . tpl
 alibaba - cloud - log / templates / alicloud - log - crd . yaml
 alibaba - cloud - log / templates / logtail - daemonset . yaml
 alibaba - cloud - log / templates / NOTES . txt
 alibaba - cloud - log / values . yaml
 NAME :    alibaba - log - controller
 LAST   DEPLOYED :  Wed   May   16   18 : 43 : 06   2018
 NAMESPACE :  default
 STATUS :  DEPLOYED
 RESOURCES :
==>  v1beta1 / ClusterRol  eBinding
 NAME                      AGE
 alibaba - log - controller    0s
==>  v1beta1 / DaemonSet
 NAME       DESIRED    CURRENT    READY    UP - TO - DATE    AVAILABLE
    NODE   SELECTOR    AGE
```

```
 logtail    2          2          0        2             0           <
 none >          0s
==>  v1beta1 / Deployment
 NAME                      DESIRED    CURRENT    UP - TO - DATE
 AVAILABLE    AGE
 alibaba - log - controller    1          1          1             0
          0s
==>  v1 / Pod ( related )
 NAME                                     READY    STATUS
    RESTARTS    AGE
 logtail - ff6rf                            0 / 1      ContainerC
 reating    0           0s
 logtail - q5s87                            0 / 1      ContainerC
 reating    0           0s
 alibaba - log - controller - 7cf6d7dbb5 - qvn6w    0 / 1
 ContainerC  reating    0           0s
==>  v1 / ServiceAcc  ount
 NAME                      SECRETS    AGE
 alibaba - log - controller    1          0s
==>  v1beta1 / CustomReso  urceDefini  tion
 NAME                                     AGE
 aliyunlogc  onfigs . log . alibabaclo  ud . com    0s
==>  v1beta1 / ClusterRol  e
 alibaba - log - controller    0s
[ SUCCESS ]  install   helm   package  :  alibaba - log - controller
 success .
```

You can run `helm Status alibaba - log - controller` to check the current states of pods. If all states are successful, installation is successful.

After successful installation, log on to the Log Service console. The Log Service project automatically created is displayed on the console. (If you have many projects, search the keyword `k8s - log` .)

Self-built Kubernetes installation

Prerequisites

1. **The Kubernetes cluster must be version 1.8 or later.**
2. **Helm 2.6.4 or later has been installed.**

Installation procedure

1. **In the Log Service console, create a project. The project name must begin with `k8s - log - custom -`.**
2. **In the following command, replace the parameters with your own, and run the command.**

```
wget   http :// logtail - release - cn - hangzhou . oss - cn -
hangzhou . aliyuncs . com / kubernetes / alicloud - log - k8s -
custom - install . sh ;  chmod   744  ./ alicloud - log - k8s -
custom - install . sh ;  sh  ./ alicloud - log - k8s - custom -
```

```
install . sh  { your - project - suffix } { region - id } { aliuid
} { access - key - id } { access - key - secret }
```

The parameters and their descriptions are as follows:

Name	Description
{your-project-suffix}	The maid-later part of the project name that you created in the second step. `k8s - log - custom -` that you have created in the second step. For example, the created project is `k8s - log - custom - xxxx` , then you mast enter `xxxx` .
{regionId}	The ID of the region where your project is located. You can view the *Service endpoint*, for example, the region ID of `China East 1 ( Hangzhou )` is `cn - hangzhou`
{aliuid}	User ID (AliUid). You need to replace the parameter with your Alibaba Cloud account ID AliUid.  Note: AliUid is a string of characters. For more information about how to view your AliUid, see step 1 in *Configure AliUids for ECS servers under other Alibaba Cloud accounts or on-premises IDCs*.
{access-key-id}	Your account access key ID. We recommend that you use the sub-account AccessKey and grant AliyunLogFullAccess permission. For more information, see *Authorization - Overview*.

Name	Description
{access-key-secret}	Your account access key secret. We recommend that you use the sub-account AccessKey and grant AliyunLogFullAccess permission. For more information, see *Authorization - Overview*.

After installation, Log Service automatically creates a machine group in the project. The machine group name is `k8s - group -${ your_k8s_c luster_id }`.



Note:

- Logstore `config - operation - log` is automatically created in the project k8s-log-${your_k8s_cluster_id}. Do not delete this Logstore.
- After self-built kubernetes installation, Logtail is granted `privileged` permissions to avoid the error during the deletion of other pods `container text file busy` error during the deletion of other pods. For more information, see *bug 1468249*, *bug 1441737*, and *issue 34538*.

Installation example

The output of the successful execution is as follows:

```
[ root @ iZbp1dsxxx xxqfbiaZ ~]# wget http :// logtail - release
 - cn - hangzhou . oss - cn - hangzhou . aliyuncs . com / kubernetes
 / alicloud - log - k8s - custom - install . sh ; chmod 744 ./
 alicloud - log - k8s - custom - install . sh ; sh ./ alicloud - log
 - k8s - custom - install . sh xxxx cn - hangzhou 165xxxxxxx
 x050 LTAxxxxxxx xxxx AIxxxxxxxx xxxxxxxxxx xxxxxxxxxx xe
....
....
....
 NAME : alibaba - log - controller
 LAST DEPLOYED : Fri May 18 16 : 52 : 38 2018
 NAMESPACE : default
 STATUS : DEPLOYED
 RESOURCES :
==> v1beta1 / ClusterRol eBinding
 NAME AGE
 alibaba - log - controller 0s
==> v1beta1 / DaemonSet
 NAME DESIRED CURRENT READY UP - TO - DATE
 AVAILABLE NODE SELECTOR AGE
 logtail - ds 2 2 0 2 0
 < none > 0s
==> v1beta1 / Deployment
```

```
 NAME                        DESIRED    CURRENT    UP - TO - DATE
 AVAILABLE    AGE
 alibaba - log - controller    1          1          1             0
          0s
==>  v1 / Pod ( related )
 NAME                                      READY    STATUS
    RESTARTS    AGE
 logtail - ds - 7xf2d                           0 / 1      ContainerC
 reating    0           0s
 logtail - ds - 9j4bx                           0 / 1      ContainerC
 reating    0           0s
 alibaba - log - controller - 796f8496b6 - 6jxb2    0 / 1
 ContainerC  reating    0           0s
==>  v1 / ServiceAcc  ount
 NAME                        SECRETS    AGE
 alibaba - log - controller    1          0s
==>  v1beta1 / CustomReso  urceDefini  tion
 NAME                                      AGE
 aliyunlogc  onfigs . log . alibabaclo  ud . com    0s
==>  v1beta1 / ClusterRol  e
 alibaba - log - controller    0s
[ INFO ]  your   k8s   is   using   project  :  k8s - log - custom
 - xxx ,  region  :  cn - hangzhou ,  aliuid  : ***************,
 accessKeyI  d  :  LTA **********
[ SUCCESS ]  install   helm   package  :  alibaba - log - controller
 success .
```

You can use the `helm status alibaba - log - controller` to view the current pod status. If all the statuses are successful, the installation is complete.

Log on to the Log Service console after installation. You can view the automatically created Log Service project. If you have many projects, search by the keyword `k8s - log` .

Step 2 Configure

Log collection supports the console configuration mode by default. Meanwhile, CRD configuration mode for the Kubernetes microservice development is also provided . You can use kubectl to manage the configuration. The comparison of the two configurations is as follows:

-	CRD Mode	Console mode
Operational complexity	Low	Medium
Function	Supports advanced configuration with the exception of Console mode	Medium
Complexity	Medium	Low
Network connection	Connect to the Kubernetes cluster	Connect to the Internet

-	CRD Mode	Console mode
Integration with deployment components	Supported	Not supported
Authentication method	Kubernetes authorization	Cloud account authentication

We recommend you use the CRD method for collection configuration management, as this method is better integrated with the Kubernetes deployment and publishing process.

Manage collection configurations on the console

Create Logtail collection configurations on the console as required. For configuration steps, see:

- *Container text log (recommended)*
- *Container standard output (recommended)*
- *Host text file*

  By default, the root directory of the host is mounted to the `/ logtail_ho st` directory of the Logtail container. You must add this prefix when configuring the path. For example, to collect data in the `/ home / logs / app_log /` directory of the host, you must set the log path on the configuration page to `/ logtail_ho st / home / logs / app_log /`.
- *Syslog*

Acquisition configuration through CRD Management

For the Kubernetes microservice development model, the logging service also provides a way to configure the CRD, you can directly use kubectl to manage the configuration, the integration of this approach with the Kubernetes deployment and release process is more complete.

For more information, see *Configure Kubernetes log collection on CRD*.

Other operations

DaemonSet deployment migration procedure

If you previously deployed the Log Service logtail by using the WebSphere set method that you used earlier, you will not be able to use CRD for configuration management. You can migrate to a new version in the following ways:

 **Note:**

During the upgrade, some logs are duplicated. The CRD management configuration can be used only for the configuration created using the CRD. The historical configuration does not support the CRD management mode because the historical configuration is created using a non-CRD mode.

1. Install in the form of a new version, the installation command last adds a parameter for the Log Service Project name that was used by your previous kubernetes cluster.

   For example, the project name was `k8s - log - demo` , the cluster ID was `c12ba2028c xxxxxxxxxx 6939f0b` , then the installation command is

   ```
   wget http :// logtail - release - cn - hangzhou . oss - cn -
   hangzhou . aliyuncs . com / kubernetes / alicloud - log - k8s -
   install . sh - O alicloud - log - k8s - install . sh ; chmod
     744 ./ alicloud - log - k8s - install . sh ; sh ./ alicloud -
   log - k8s - install . sh c12ba2028c xxxxxxxxxx 6939f0b k8s -
   log - demo
   ```

2. After successful installation, in the Log Service console apply the historical collection configuration to the new machine group `k8s - group -${ your_k8s_c luster_id }`.
3. In a minute, the historical collection configuration is bind to the historical machine group.
4. After the log collection is normalized, you can delete the previously installed Logtail DaemonSet.

Use multiple clusters in the same Log Service project

You can use multiple clusters to collect logs to the same Log Service project. When installing other clusters Log Service components, you must replace `${ your_k8s_c luster_id }` in the installation parameters with the clusters ID you installed for the first time.

For example, you now have three clusters with IDs: abc001, abc002, and abc003. The installation parameters for the three clusters, `${ your_k8s_c luster_id }`, must all be filled as `abc001` .

 **Note:**

This method does not support Kubernetes multi-cluster sharing across regions.

Logtail container logs

Logtail logs are stored in the `/ usr / local / ilogtail /` directory in the Logtail container, the file name is `ilogtail . LOG` and `ilogtail . plugin`, the container stdout does not have the reference significance, so you can ignore the following stdout output:

```
start umount useless mount points , / shm $|/ merged $|/
mqueue $
umount : / logtail_ho st / var / lib / docker / overlay2 /
3fd0043af1 74cb0273c3 c7869500fb e2bdb95d13 b1e110172e
f57fe840c8 2155 / merged : must be superuser to unmount
umount : / logtail_ho st / var / lib / docker / overlay2 /
d5b10aa193 99992755de 1f85d25009 528daa749c 1bf8c16edf
f44beab6e6 9718 / merged : must be superuser to unmount
umount : / logtail_ho st / var / lib / docker / overlay2 /
5c3125dadd acedec29df 72ad0c52fa c800cd56c6 e880dc4e8a
640b1e16c2 2dbe / merged : must be superuser to unmount
......
xargs : umount : exited with status 255 ; aborting
umount done
start logtail
ilogtail is running
logtail status :
ilogtail is running
```

View the status of log related components in the Kubernetes cluster

```
helm status alibaba - log - controller
```

alibaba-log-controller failed to start

Make sure that you perform the installation as follows:

1. The installation command is executed on the master node of the kubernetes Cluster
2. The installation command parameter is entered in the cluster ID.

If the installation fails due to these problems, use `helm del -- purge alibaba - log - controller r` to remove the installation package and perform the installation again.

If the installation failure persists, open a ticket to contact technical support engineers of Log Service.

Check the status of the Logtail DaemonSet in the Kubernetes cluster

You can run the command `kubectl get ds - n kube - system` to check the running status of Logtail.



Note:

The default namespace of Logtail is `kube - system` .

Check the version number, IP address, and startup time of Logtail

An example is as follows:

```
[ root @ iZbp1dsu6v  77zfb40qfb  iaZ  ~]#  kubectl   get   po  - n
 kube - system  |  grep   logtail
 NAME              READY       STATUS      RESTARTS     AGE
 logtail - ds - gb92k     1 / 1         Running     0            2h
 logtail - ds - wm7lw     1 / 1         Running     0            4d
[ root @ iZbp1dsu6v  77zfb40qfb  iaZ  ~]#  kubectl   exec   logtail
 - ds - gb92k  - n   kube - system   cat  / usr / local / ilogtail /
 app_info . json
{
   " UUID " : "",
   " hostname " : " logtail - ds - gb92k ",
   " instance_i  d " : " 0EBB2B0E - 0A3B - 11E8 - B0CE - 0A58AC1404
 02_172 . 20 . 4 . 2_15178109  40 ",
   " ip " : " 172 . 20 . 4 . 2 ",
   " logtail_ve  rsion " : " 0 . 16 . 2 ",
   " os " : " Linux ;  3 . 10 . 0 - 693 . 2 . 2 . el7 . x86_64 ; # 1
 SMP   Tue   Sep   12   22 : 26 : 13   UTC   2017 ;  x86_64 ",
   " update_tim  e " : " 2018 - 02 - 05   06 : 09 : 01 "
}
```

View the operational log for Logtail

Logtail running logs are stored in the `/ usr / local / ilogtail /` directory.

The file name is `ilogtail . LOG` . The rotation file is compressed and stored as `ilogtail . LOG . x . gz` .

An example is as follows:

```
[ root @ iZbp1dsu6v  77zfb40qfb  iaZ  ~]#  kubectl   exec   logtail
 - ds - gb92k  - n   kube - system   tail  / usr / local / ilogtail /
 ilogtail . LOG
[ 2018 - 02 - 05   06 : 09 : 02 . 168693 ] [ INFO ] [ 9 ] [ build /
 release64 / sls / ilogtail / LogtailPlu  gin . cpp : 104 ]  logtail
 plugin   Resume : start
[ 2018 - 02 - 05   06 : 09 : 02 . 168807 ] [ INFO ] [ 9 ] [ build /
 release64 / sls / ilogtail / LogtailPlu  gin . cpp : 106 ]  logtail
 plugin   Resume : success
[ 2018 - 02 - 05   06 : 09 : 02 . 168822 ] [ INFO ] [ 9 ] [ build /
 release64 / sls / ilogtail / EventDispa  tcher . cpp : 369 ]  start
 add   existed   check   point   events ,  size : 0
[ 2018 - 02 - 05   06 : 09 : 02 . 168827 ] [ INFO ] [ 9 ] [ build /
 release64 / sls / ilogtail / EventDispa  tcher . cpp : 511 ]  add
   existed   check   point   events ,  size : 0   cache   size : 0
 event   size : 0   success   count : 0
```

```
```

Restart Logtail of a Pod

An example is as follows:

```
[ root @ iZbp1dsu6v  77zfb40qfb  iaZ  ~]#  kubectl   exec   logtail -
 gb92k  - n   kube - system  / etc / init . d / ilogtaild   stop
 kill   process   Name :  ilogtail   pid :  7
 kill   process   Name :  ilogtail   pid :  9
 stop   success
[ root @ iZbp1dsu6v  77zfb40qfb  iaZ  ~]#  kubectl   exec   logtail -
 gb92k  - n   kube - system  / etc / init . d / ilogtaild   start
 ilogtail   is   running
```

3.6.3 Container text logs

Logtail supports collecting text logs generated in containers and uploading the collected logs together with the container metadata to Log Service.

Function features

Compared with the basic log file collection, Docker file collection has the following features:

- Set the log path within the container, no need to care about the mapping from this path to the host.
- Supports using labels to specify containers to be collected.
- Supports using labels to exclude specific containers.
- Supports environments to specify containers to be collected.
- Supports environments to exclude specific containers.
- Supports multiline logs (such as Java stack logs).
- Supports automatic tagging for container data.
- Supports automatic tagging for Kubernetes containers.

Limits

- Policy of stopping collection: When the container is stopped, Logtail stops collecting logs from the container after listening to the `die` event of the container (with a delay of 1–3 seconds). If a collection delay occurs during this time, it is possible to lose part of the logs before the stop.
- Docker storage drives: Currently, only overlay and overlay2 drives are supported. For other drive types, you must mount the log directory to your local PC.

- Logtail running methods: Logtail must be run as a container and follow the Logtail deployment method.
- Label: The label is the label information in the Docker inspect, not the label in the Kubernetes configuration.
- Environment: The environment is the environment information configured in the container startup.

Procedure

1. Deploy and configure the Logtail container.
2. Set the collection configuration in Log Service.

1. Logtail deployment and configuration

- Kubernetes

  For more information on Kubernetes log collection, see *Collect Kubernetes logs*.

- Other container management methods

  For more information on other container management methods such as Swarm and Mesos, see *Collect standard Docker logs*.

2. Collection configuration in Log Service

1. On the Logstore List page, click the Data Import Wizard icon to enter the configuration process.
2. Select a data source.

   Select Docker File under Third-Party Software and then click Next.
3. Configure the data source.

Configuration item	Required	Description
Docker file	Yes	Confirm if the target file being collected is a Docker file.

Configuration item	Required	Description
Label whitelist	Optional	LabelKey is required. If LabelValue is not empty, only containers whose label includes LabelKey = LabelValue are collected. If LabelValue is empty, all the containers whose label includes the LabelKey are collected.  Note: a. Key-value pairs have an OR relationship between each other, that is, a container is collected if its label includes any of the key-value pairs. b. Here the label is the label information in Docker inspect.
Label blacklist	Optional	LabelKey is required. If LabelValue is not empty, only containers whose label includes LabelKey = LabelValue are excluded. If LabelValue is empty, all the containers whose label includes the LabelKey are excluded.  Note: a. Key-value pairs have an OR relationship between each other, that is, a container is excluded if its label includes any of the key-value pairs. b. Here the label is the label information in Docker inspect.

Configuration item	Required	Description
Environment whitelist	Optional	EnvKey is required. If EnvValue is not empty, only containers whose environment includes EnvKey=EnvValue are collected. If EnvValueis empty, all the containers whose environment includes the EnvKey are collected.  Note: · Key-value pairs have an OR relationship between each other, that is, a container is collected if its environment includes any of the key-value pairs. · Here the environment is the environment information configured in the container startup.
Environment blacklist	Optional	EnvKeyis required. If EnvValue is not empty, only containers whose environment includes EnvKey=EnvValue are excluded. If EnvValue is empty, all the containers whose environment includes EnvKey are excluded.  Note: a. Key-value pairs have an OR relationship between each other, that is, a container is collected if its environment includes any of the key-value pairs. b. Here the environment is the environment information configured in the container startup.
Other configurations	-	For other collection configurations and parameter descriptions, see *Collect text logs*.

4. Description

   · In this topic, labels refer to label information contained in Docker inspect.

   · The namespace and container name in Kubernetes are mapped to labels `io . kubernetes . pod . namespace` and `io . kubernetes . container . name` in a docker. For example, the Pod you have created belongs to the backend-prod namespace, and the container name is worker-server. In this case,

you can configure two whitelist labels `io . kubernetes . pod . namespace : backend - prod` and `io . kubernetes . container . name : worker - server` to specify that only logs in the worker-server container can be collected.

- We recommend that you use only the `io . kubernetes . pod . namespace` and `io . kubernetes . container . name` labels in Kubernetes. For other scenarios, you can use an environment whitelist or blacklist.

5. Apply to the machine group.

   On the Apply to Machine Group page, select the Logtail machine group to be collected and click Apply to Machine Group to apply the configuration to the selected machine group. If you have not created a machine group, click Create Machine Group to create one.

6. Complete the process of accessing container text logs.

   To configure the Search, Analysis, and Visualization function and the Shipper & ETL function, complete the settings as instructed on the page.

Configuration example

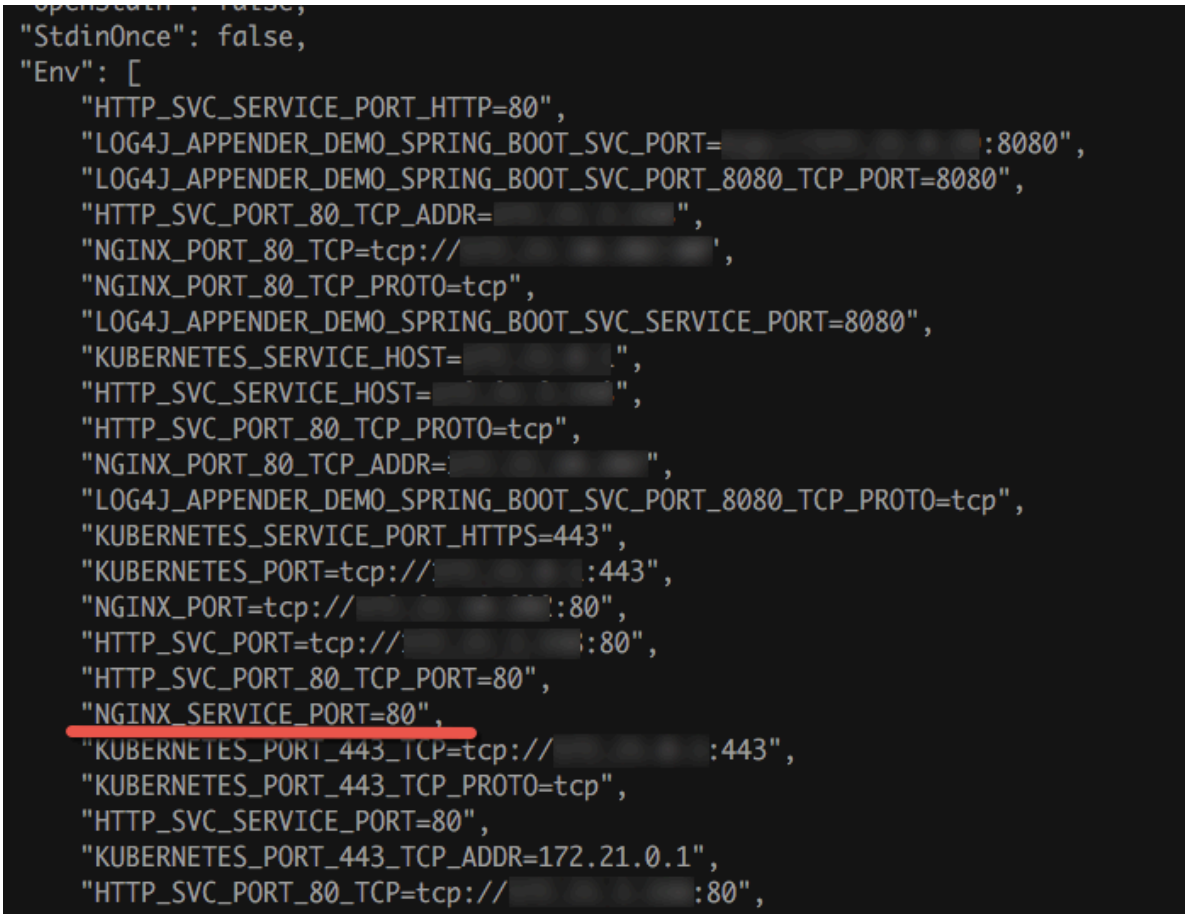
- Environment configuration

  Collect the logs of containers whose environment is `NGINX_PORT _80_TCP_PORT = 80` , and is not `POD_NAMESP ACE = kube - system` . The log file path is `/ var / log / nginx / access . log` , and the logs are parsed in the simple mode.

   **Note:**

The environment is the environment information configured in the container startup.

Figure 3-24: Example of Environment Configuration



The data source configuration in this example is as follows. For other collection configurations and parameter descriptions, see *Collect text logs*.

- Label configuration

  Collect the logs of containers whose label is `io . kubernetes . container . name = nginx` , and is not type=pre . The log file path is `/ var / log / nginx / access . log` , and the logs are parsed in the simple mode.

  Note:

The label is the label information in the Docker inspect, not the label in the Kubernetes configuration.

Figure 3-25: Example label Mode



The data source configuration in this example is as follows. For other collection configurations and parameter descriptions, see *Collect text logs*.

Default field

Normal Docker

The following fields are uploaded by each log by default.

Field	Description
`_image_nam e_`	Image name.
`_container _name_`	Container name
`_container _ip_`	Container IP address

Kubernetes

If the cluster is a Kubernetes cluster, the following fields are uploaded by each log by default.

Field	Description
`_image_nam e_`	Image name
`_container _name_`	Container name
`_pod_name_`	Pod name
`_namespace _`	Namespace where a Pod resides

Field	Description
`_pod_uid_`	The unique identifier of a Pod
`_container _ip_`	Pod IP address

3.6.4 Containers-standard output

Logtail supports using the standard output stream of the container as the input source, and uploading the standard output stream together with the container metadata to Log Service.

Features

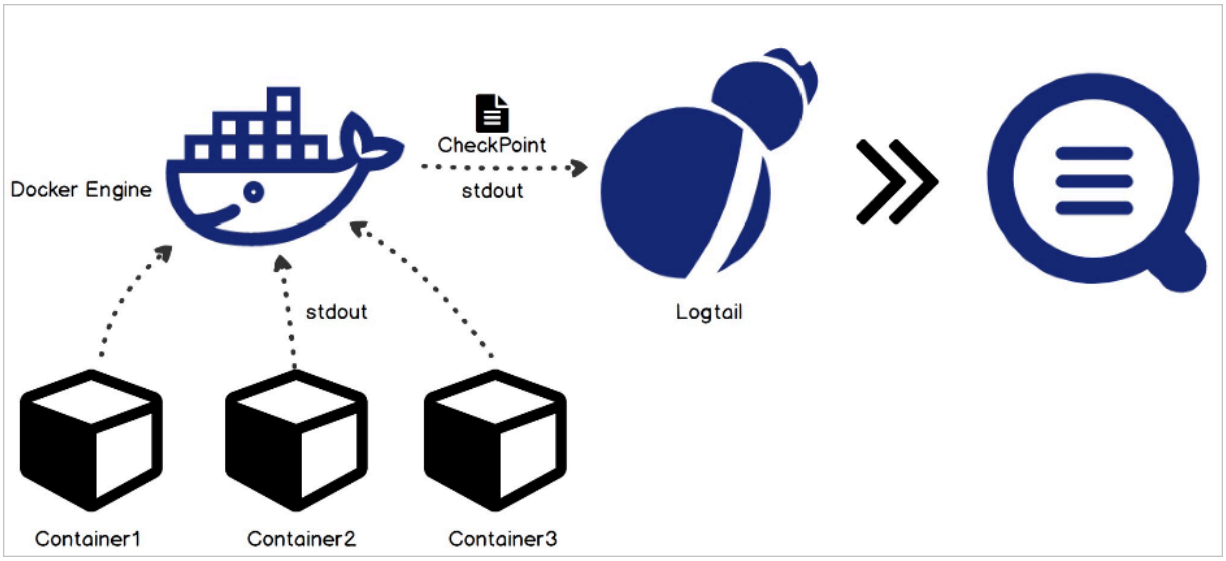
- Supports collection stdout and stderr
- Supports label specified collection containers
- Supports using labels to exclude specific containers
- Supports environments to to specify containers to be collected.
- Supports environments to exclude specific containers.
- Supports multi-line logs (like java stack logs)
- Supports automatic tagging of container data
- Supports automatic tagging for Kubernetes containers

Implementation principle

As shown in the following figure, Logtail communicates with the Domain Socket of Docker to query all of the containers running on Docker and then locate the containers to be collected according to label information. Then, Logtail uses the Docker log command to retrieve the specified container log.

When Logtail collects the standard output of the container, it periodically saves the collected point information in the checkpoint file. If Logtail is restarted after being stopped, the log will be collected from the last saved point.

Figure 3-26: Implementation principle



Limits

- Currently, this feature only supports Linux and depends on Logtail 0.16.0 and later versions. For version check and upgrade, see *Install Logtail in Linux*.
- By default, Logtail uses `/var/run/docker.sock` to access Docker Engine. Make sure that Domain Socket exists and has access permissions.
- Multiline log limit. To ensure that a logmade up of multiple lines is not split up due to output delay, multi-line logs will be cached for a short time by default. The default cache time is three seconds, but can be changed by using the `BeginLineTimeoutMs` parameter. However, this value cannot be less than 1000. Otherwise, the operation may be prone to error.
- Policy of stopping collection. When the container is stopped, Logtail stops collecting the standard output from the container after listening to the container to `die` event. If a collection delay occurs during this time, it is possible to lose parts of the output before the stop.
- Context limit. Each collection is deployed to the same context by default. If you need a different context for each type of container, they must be set individually.
- Data processing. The default field of collected data is `content`, which supports common processing configurations.

- Label. The label is the label information in the Docker inspect, not the label in the Kubernetes configuration.
- Environment The environment is the environment information configured in the container startup.

Configuration process

1. Deploy and configure the Logtail container.
2. Set the collection configuration in Log Service.

1. Logtail deployment and configuration

- Kubernetes
- Other container management methods

2. Collection configuration in Log Service

1. On the Logstore List page, click the Data Import Wizard icon to enter the configuration process.
2. Select the data source.

   Select Docker Stdout under Third-Party Software and then click Next.
3. Configure the data source.

   On the Configure Data Source page, complete your collection configuration. See the following example.

```
{
 " inputs ": [
     {
         " type ": " service_do  cker_stdou  t ",
         " detail ": {
             " Stdout ":  true ,
             " Stderr ":  true ,
             " IncludeLab  el ": {
                 " io . kubernetes . container . name ": " nginx "
             },
             " ExcludeLab  el ": {
                 " io . kubernetes . container . name ": " nginx -
 ingress - controller "
             },
             " IncludeEnv ": {
                 " NGINX_SERV  ICE_PORT ": " 80 "
             },
             " ExcludeEnv ": {
                 " POD_NAMESP  ACE ": " kube - system "
             }
         }
     }
 ]
```

```
}
```

4. Apply the configuration to the machine group.

   Enter the apply to machine group page. select the Logtail machine group to be collected and click Apply to Machine Group to apply the configuration to the selected machine group. If you have not created a machine group, click Create Machine Group to create one.

Description

The input source type is `service_docker_stdout`

Configuration items	Type	Required or not	Description
IncludeLabel	The mapping type, where key and value are both strings.	Required.	Empty by default. When this is empty, all Container data will be collected. When key is not empty and value is empty, all containers with a label containing this key will be collected. Note: 1. Key-value pairs have an OR relationship between each other, that is, a container is collected if its label includes any of the key-value pairs. 2. Here the label is the label information in Docker inspect.

Configuration items	Type	Required or not	Description
ExcludeLabel	The mapping type, where key and value are both strings.	Optional	Empty by default. When empty, no Containers will be excluded. When key is not empty and value is empty, all containers with a label that contains this key will be excluded.  Note: 1. All key-value pairs have an OR relationship. As long as the label for a container includes one of the key-value pairs, it will be excluded. 2. Here the label is the label information in Docker inspect.
IncludeEnv	The mapping type, where key and value are both strings.	Optional	Empty by default. If empty, all containers are collected. If the key is not empty but the value is empty, all the containers whose environment includes this key are collected.  Note: 1. Key-value pairs have an OR relationship between each other, that is, a container is collected if its environment includes any of the key-value pairs. 2. The environment is the environment information configured in the container startup.

Configuration items	Type	Required or not	Description
ExcludeEnv	The mapping type, where key and value are both strings.	Optional	Empty by default. If empty, no containers are excluded. If the key is not empty but the value is empty, all the containers whose environment includes this key are excluded.  Note: 1. Key-value pairs have an OR relationship between each other, that is, a container is excluded if its environment includes any of the key-value pairs. 2. The environment is the environment information configured in the container startup.
Stdout	bool	Optional	True by default. When false, stdout data will not be collected.
Stderr	bool	Optional	True by default. When false, stderr data will not be collected.
BeginLineRegex	String	Optional	Empty by default. When not empty it is the first match to the regular expression in the line . If a line matches the regular expression, then that line will be treated as a new log. Otherwise the line of data will be connected to the previous log.
BeginLineTimeoutMs	int	Optional	Timeout time for matchin a lin, measured in miliseconds, 3000 by default. Every 3 seconds, if a new log has not appeared, the last log will be output.

Configuration items	Type	Required or not	Description
BeginLineCheckLength	int	Optional	The length (in bytes) of the beginning of a row used to match with the regular expression. The default value is 10*1024. If the regular expression can match with the row within the first N bytes, configure this parameter to increase the matching efficiency.
MaxLogSize	int	Optional	The maximum length (in bytes) of a log. The default value is 512*1024. If the log exceeds this setting, it will not continue to be searched, rather it will be directly uploaded.

Default fields

Normal Docker

The following fields are uploaded by each log by default.

Field name	Description:
`_time_`	The data time. For example, `2018-02-02T02:18:41.979147844Z`.
`_source_`	The input source type, either stdout or stderr.
`_image_name_`	The image name.
`_container_name_`	The container name.

Kubernetes

If the cluster is a Kubernetes cluster, the following fields are uploaded by each log by default.

Field name	Description:
`_time_`	The data time. For example, `2018-02-02T02:18:41.979147844Z`.

Field name	Description:
`_source_`	The input source type, either stdout or stderr.
`_image_nam e_`	The image name.
`_container _name_`	The container name.
`_pod_name_`	The pod name.
`_namespace _`	The namespace where the pod resides.
`_pod_uid_`	The unique identifier for the pod.

Configuration example

General configuration

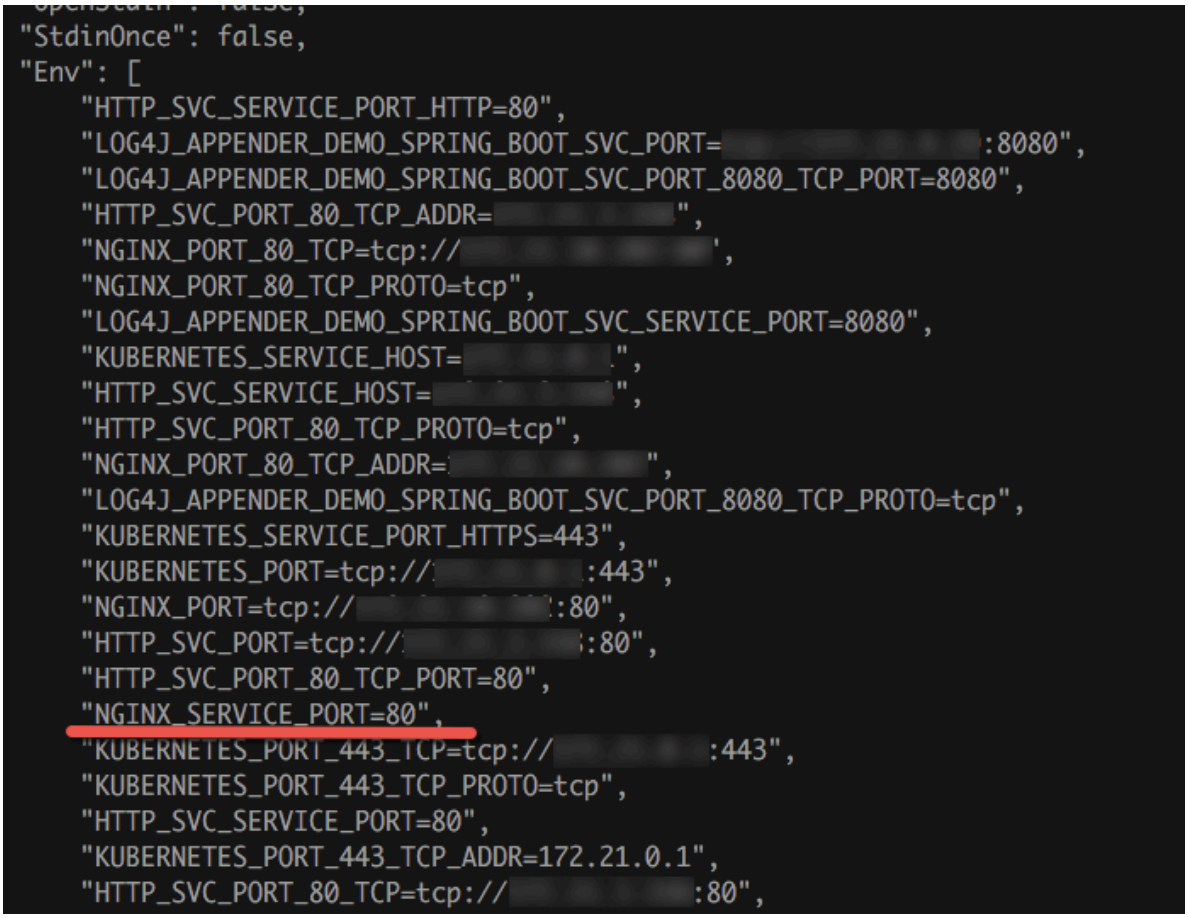
- **Environment configuration configuration**

  Collect the stdout logs and stderr logs of containers whose environment is `NGINX_PORT _80_TCP_PO RT = 80` , and is not `POD_NAMESP ACE = kube - system` :

   **Note:**

The environment is the environment information configured in the container startup.

Figure 3-27: Example of Environment Configuration



Collection configuration

```
{
    " inputs ": [
        {
            " type ": " service_do  cker_stdou  t ",
            " detail ": {
                " Stdout ":  true ,
                " Stderr ":  true ,
                " IncludeEnv ": {
                    " NGINX_PORT  _80_TCP_PO  RT ": " 80 "
                },
                " ExcludeEnv ": {
                    " POD_NAMESP  ACE ": " kube - system "
                }
            }
        }
    ]
```

```
}
```

- Label configuration

  Collect the stdout logs and stderr logs of containers whose label is `io . kubernetes . container . name = nginx` , and is not `type = pre` :

  > **Note:**
  > The label here is Docker not the label in the Kubernetes configuration.

  Figure 3-28: Label configuration example




```
{
    " inputs ": [
        {
            " type ": " service_do  cker_stdou  t ",
            " detail ": {
                " Stdout ":  true ,
                " Stderr ":  true ,
                " IncludeLab  el ": {
                    " io . kubernetes . container . name ": " nginx
 "
                },
                " ExcludeLab  el ": {
                    " type ": " pre "
                }
            }
        }
    ]
}
```

Collection configuration of multiline logs

Multi-line log collection is particularly important for the collection of Java exception stack output. Here we introduce a standard Java standard output log collection configuration.

- Log sample

```
2018 - 02 - 03   14 : 18 : 41 . 968   INFO  [ spring - cloud -
monitor ] [ nio - 8080 - exec - 4 ]  c . g . s . web . controller .
DemoContro  ller  :  service   start
2018 - 02 - 03   14 : 18 : 41 . 969   ERROR  [ spring - cloud -
monitor ] [ nio - 8080 - exec - 4 ]  c . g . s . web . controller .
DemoContro  ller  :  java . lang . NullPointe  rException
at   org . apache . catalina . core . Applicatio  nFilterCha  in .
internalDo  Filter ( Applicatio  nFilterCha  in . java : 193 )
at   org . apache . catalina . core . Applicatio  nFilterCha  in .
doFilter ( Applicatio  nFilterCha  in . java : 166 )
at   org . apache . catalina . core . StandardWr  apperValve .
invoke ( StandardWr  apperValve . java : 199 )
at   org . apache . catalina . core . StandardCo  ntextValve .
invoke ( StandardCo  ntextValve . java : 96 )
...
2018 - 02 - 03   14 : 18 : 41 . 968   INFO  [ spring - cloud -
monitor ] [ nio - 8080 - exec - 4 ]  c . g . s . web . controller .
DemoContro  ller  :  service   start   done
```

- Collection configuration

  Collect input logs of containers whose label is `app = monitor` and the beginning of a row is of the date type (to increase matching efficiency, only the first 10 bytes of the row is used to check for a match with the regular expression).

```
{
" inputs ": [
  {
    " detail ": {
      " BeginLineC  heckLength ":  10 ,
      " BeginLineR  egex ": "\\ d +-\\ d +-\\ d +. *",
      " IncludeLab  el ": {
        " app ": " monitor "
      }
    },
    " type ": " service_do  cker_stdou  t "
  }
]
}
```

Process collected data

Logtail supports common data processing methods for collected Docker standard output. We recommend that you use a regular expression to parse logs into time, module, thread, class, and info based on the multiline log format in the previous section.

- **Collection configuration:**

  Collect input logs of containers whose label is `app = monitor` **and the beginning of a row is of the date type (to increase matching efficiency, only the first 10 bytes of the row is used to check for a match with the regular expression).**

```
{
" inputs ": [
  {
    " detail ": {
      " BeginLineC  heckLength ":  10 ,
      " BeginLineR  egex ": "\\ d +-\\ d +-\\ d +. *",
      " IncludeLab  el ": {
        " app ": " monitor "
      }
    },
    " type ": " service_do  cker_stdou  t "
  }
],
" Processors  ":[
    {
        " type ": " processor_  regex ",
        " detail ": {
            " SourceKey ": " content ",
            " Regex ": "(\\ d +-\\ d +-\\ d + \\ d +:\\ d +:\\ d +\
\.\\ d +)\\ s +(\\ w +)\\ s +\\[([^]]+)]\\ s +\\[([^]]+)]\\ s +:\\ s
 +(. *)",
            " Keys ": [
                " time ",
                " module ",
                " thread ",
                " class ",
                " info "
            ],
            " NoKeyError ":  true ,
            " NoMatchErr  or ":  true ,
            " KeepSource ":  false
        }
    }
]
}
```

- **Sample output:**

  The output after processing the log `2018 - 02 - 03  14 : 18 : 41 . 968  INFO  [ spring - cloud - monitor ] [ nio - 8080 - exec - 4 ]  c . g . s . web . controller . DemoContro  ller  :  service  start  done` **is as follows:**

```
__tag__ : __hostname  __ : logtail - dfgef
_container  _name_ : monitor
_image_nam  e_ : registry . cn - hangzhou . aliyuncs . xxxxxxxxxx
xxxxx
_namespace  _ : default
_pod_name_ : monitor - 6f54bd5d74 - rtzc7
_pod_uid_ : 7f012b72 - 04c7 - 11e8 - 84aa - 00163f00c3  69
_source_ : stdout
```

```
_time_ : 2018 - 02 - 02T14 : 18 : 41 . 979147844Z
Time :  2018 - 02 - 02   02 : 18 : 41 .  968
level : INFO
module : spring - cloud - monitor
Thread :  fig
Class :  c . g . s .  web .  Controller .  demcontrol  ler
message : service   start   done
```

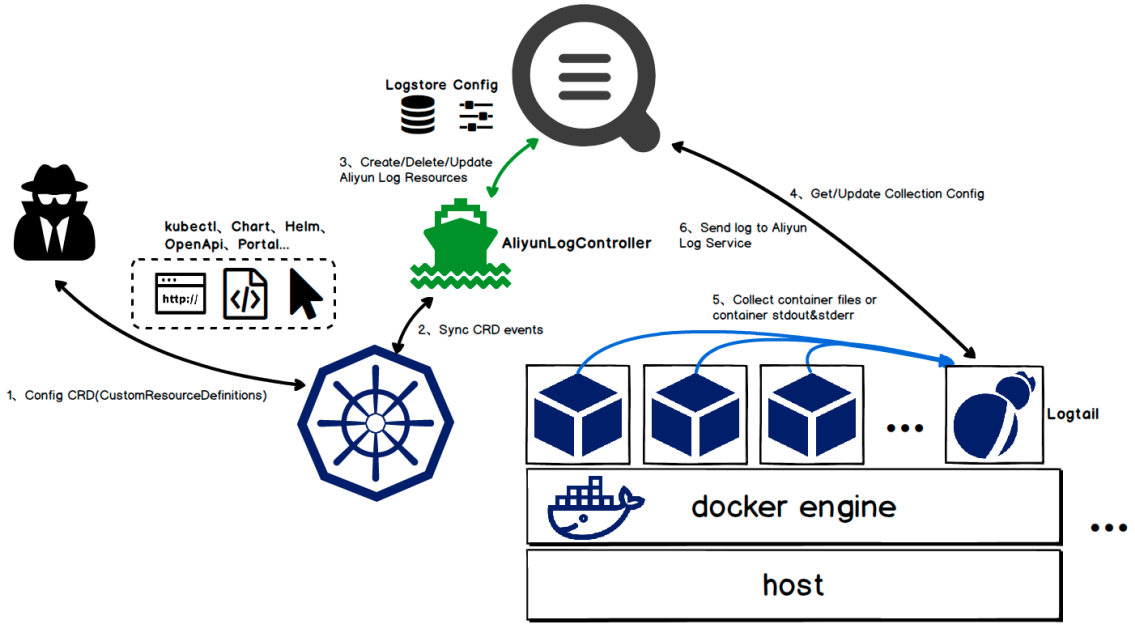
3.6.5 Configure Kubernetes log collection on CRD

Log collection is configured on the console by default. Log Service also provides CRD configuration for log collection for Kubernetes microservice development. This allows you to use kubectl to manage configurations.

We recommend you use the CRD method for collection configuration management, as this method is better integrated with the Kubernetes deployment and publishing process.

Implementation principles

Figure 3-29: Implementation principles



Run the installation command to install the `alibaba - log - controller` Helm package. The Helm package mainly run the following operations:

1. Create aliyunlogconfigs CRD (Custom Resource Definition).
2. Deploy alibaba-log-controller.
3. Deploy Logtail DaemonSet.

The internal workflow of configuration is as follows:

1. Use `kubectl` or other tools to apply the aliyunlogconfigs CRD configuration.
2. alibaba-log-controller detects configuration update.
3. alibaba-log-controller automatically submits requests for Logstore creation, configuration creation, and configuration application to machine groups based on the CRD content and server status.
4. Logtail running in DaemonSet mode periodically sends requests for server configuration, obtains the new or updated configuration, and performs the rapid loading.
5. Logtail collects standard outputs or files from each container (pod) based on the configuration information.
6. Logtail sends processed and aggregated data to the Log Service.

Configuration method

 **Note:**

If you have used the Logtail deployed in DaemonSet mode, you cannot manage configurations in CRD mode. For more information, see Migration process for the DaemonSet deployment mode in this document.

You must define the CRD of AliyunLogConfig to create configurations, and delete the corresponding CRD resource to delete the configuration. The CRD is configured as follows:

```
apiVersion : log . alibabaclo ud . com / v1alpha1 ## Default
value , no need for change
kind : AliyunLogC onfig ## Default value , no need for
change
metadata :
  name : simple - stdout - example ## Resource name , which
must be unique in the cluster
spec :
  logstore : k8s - stdout ## Logstore name , automatica lly
created if no name exists
  shardCount : 2 ## [ Optional ] Number of Logstore shards
. The default value is 2 . The value range is 1
to 10 .
  lifeCycle : 90 ## [ Optional ] Storage period of the
Logstore . The default value is 90 . The value range
  is 1 to 7300 . The value 7300 indicates permanent
storage .
  logtailCon fig : ## Detailed configurat ion
    inputType : plugin ## Input type of collection .
Generally , the value is file or plugin .
    configName : simple - stdout - example ## Collection
configurat ion name . The value must the same as the
  resource name ( metadata . name ).
    inputDetai l : ## Detailed configurat ion informatio n ,
see the example
```

```
      ...
```

After the configuration is completed and applied, alibaba-log-controller is created automatically.

View configuration

You can check the configuration on the Kubernetes CRD or console.

For how to view configuration on the console, see *Create a Logtail configuration*.

> **Note:**
> If you use the CRD method to manage configuration, the configuration changes you have made on the console will be overwritten when you update configuration on the CRD.

- Run `kubectl get aliyunlogconfigs` to view all the configurations.

```
[root@iZbp1dsbiaZ ~]# kubectl get aliyunlogconfigs
NAME AGE
regex-file-example 10s
regex-stdout-example 4h
simple-file-example 5s
```

- Run `kubectl get aliyunlogconfigs ${config_name} -o yaml` to view the detailed configuration and status.

  The `status` field in the configuration shows the configuration execution result. If the configuration is successfully applied, the value of `statusCode` is 200 in the `status` field. If the value of `statusCode` is not 200, applying the configuration failed.

```
[root@iZbp1dsbiaZ ~]# kubectl get aliyunlogconfigs
simple-file-example -o yaml
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
annotations:
  kubectl.kubernetes.io/last-applied-configuration: |

clusterName: ""
creationTimestamp: 2018-05-17T08:44:46Z
generation: 0
name: simple-file-example
namespace: default
resourceVersion: "21790443"
selfLink: /apis/log.alibabacloud.com/v1alpha1/
namespaces/default/aliyunlogconfigs/simple-file-
example
uid: 8d3a09c4-59ae-11e8-851d-00163f008685
spec:
lifeCycle: null
```

```
logstore :  k8s - file
logtailCon  fig :
  configName :  simple - file - example
  inputDetai  l :
    dockerFile :  true
    dockerIncl  udeEnv :
      ALIYUN_LOG  TAIL_USER_  DEFINED_ID : ""
    filePatter  n :  simple . LOG
    logPath : / usr / local / ilogtail
    logType :  common_reg  _log
  inputType :  file
machineGro  ups :  null
project : ""
shardCount :  null
status :
status :  OK
statusCode :  200
```

Configuration example

Container standard output

In the container standard output, set `inputType` to `plugin` and fill the detailed information in the `plugin` field under `inputDetai l` . For more information on the configuration fields, see *Container stdout*.

- Simple collection mode

  Collect standard outputs (stdout and stdeer) of all containers except for those who has environment variable configuration `COLLECT_ST DOUT_FLAG = false` .

```
apiVersion :  log . alibabaclo  ud . com / v1alpha1
kind :  AliyunLogC  onfig
metadata :
 #  your  config  name ,  must  be  unique  in  your  k8s
cluster
  name :  simple - stdout - example
spec :
 #  logstore  name  to  upload  log
  logstore :  k8s - stdout
 #  logtail  config  detail
  logtailCon  fig :
   #  docker  stdout ' s  input  type  is  ' plugin '
    inputType :  plugin
   #  logtail  config  name ,  should  be  same  with  [
metadata . name ]
    configName :  simple - stdout - example
    inputDetai  l :
      plugin :
        inputs :
         -
           #  input  type
            type :  service_do  cker_stdou  t
            detail :
             #  collect  stdout  and  stderr
              Stdout :  true
              Stderr :  true
```

```
              #  collect   all   container ' s   stdout   except
containers   with  " COLLECT_ST  DOUT_FLAG : false "  in   docker
env   config
               ExcludeEnv :
                 COLLECT_ST  DOUT_FLAG : " false "
```

- Custom collection mode

  Collect the access log of Grafana and parse the access log into structured data.

  Grafana container has environment variable configuration `GF_INSTALL _PLUGINS = grafana - piechart -.....`. You can set `IncludeEnv` to `GF_INSTALL _PLUGINS : ''` to enable the Logtail to collect standard outputs from this container only.

  Figure 3-30: Custom collection mode



  The access log of Grafana is in the following format:

```
t = 2018 - 03 - 09T07 : 14 : 03 + 0000   lvl = info   msg ="
Request   Completed "  logger = context   userId = 0   orgId = 0
uname =  method = GET   path =/  status = 302   remote_add  r = 172
. 16 . 64 . 154   time_ms = 0   size = 29   referer =
```

  Parse the access log using a regular expression. The detailed configuration is as follows:

```
apiVersion :  log . alibabaclo  ud . com / v1alpha1
kind :  AliyunLogC  onfig
metadata :
 #  your   config   name ,  must   be   unique   in   your   k8s
cluster
  name :  regex - stdout - example
spec :
 #  logstore   name   to   upload   log
  logstore :  k8s - stdout - regex
 #  logtail   config   detail
  logtailCon  fig :
   #  docker   stdouts   input   type   is   plugin
    inputType :  plugin
   #  logtail   config   name ,  should   be   same   with  [
metadata . name ]
    configName :  regex - stdout - example
```

```
    inputDetai  l :
      plugin :
        inputs :
         -
           #  input   type
            type :  service_do  cker_stdou  t
            detail :
             #  Collect   stdout   outputs   only   and   do   not
  collect   stdeer   outputs .
              Stdout :  true
              Stderr :  false
             #  Collect   only   stdout   outputs   whose   key
is  " GF_INSTALL  _PLUGINS "  in   the   environmen  t   variable
configurat  ion   from   the   container .
              IncludeEnv :
                GF_INSTALL  _PLUGINS : ''
        processors :
         -
           #  Use   a   regular   expression
            type :  processor_  regex
            detail :
             #  The   data   collected   by   the   docker   has
key  " content "  by   default .
              SourceKey :  content
             #  Regular   expression   for   extraction
              Regex : ' t =(\ d +-\ d +-\ w +:\ d +:\ d +\+\ d +)
lvl =(\ w +)  msg ="([^"]+)"  logger =(\ w +)  userId =(\ w +)
orgId =(\ w +)  uname =(\ S *)  method =(\ w +)  path =(\ S +)
status =(\ d +)  remote_add  r =(\ S +)  time_ms =(\ d +)  size =(\
d +)  referer =(\ S *). *'
             #  Extracted   keys
              Keys : [' time ', ' level ', ' message ', ' logger ',
' userId ', ' orgId ', ' uname ', ' method ', ' path ', ' status ',
' remote_add  r ', ' time_ms ', ' size ', ' referer ']
             #  Retain   the   original   fields
              KeepSource :  true
              NoKeyError :  true
              NoMatchErr  or :  true
```

After the configuration is applied, the data collected by Log Service is as follows:

Figure 3-31: Collected log data



Container file

- **Simple file**

  Collect log files from containers whose environment variable configuration contains key `ALIYUN_LOG_TAIL_USER_DEFINED_ID`. The log file path is `/data/logs/app_1` and the file name is `simple.LOG`.

```
apiVersion: log.alibabaclo ud.com/v1alpha1
kind: AliyunLogC onfig
metadata:
 # your config name, must be unique in your k8s
cluster
  name: simple-file-example
spec:
 # logstore name to upload log
  logstore: k8s-file
 # logtail config detail
  logtailCon fig:
   # log file's input type is 'file'
    inputType: file
   # logtail config name, must same with [metadata.
name]
    configName: simple-file-example
    inputDetai l:
     # Set logType to "common_reg _log" for simple
mode logs
      logType: common_reg _log
     # Log file folder
      logPath: /data/logs/app_1
     # File name, which supports wildcards, for
example, log_*.log
      filePatter n: simple.LOG
     # Collect files from the container. dockerFile
flag is set to true
      dockerFile: true
     # Only collect container with "ALIYUN_LOG
TAIL_USER_DEFINED_ID" in docker env config
      dockerIncl udeEnv:
        ALIYUN_LOG_TAIL_USER_DEFINED_ID: ""
```

- **Complete regular expression files**

  The following is an example of a Java program log:

```
[2018-05-11T20:10:16,000][INFO][SessionTra cker]
 [SessionTra ckerImpl.java:148] Expiring sessions
 java.sql.SQLExcepti on: Incorrect string value: '\xF0
 \x9F\x8E\x8F",...' for column 'data' at row 1
 at org.springfram ework.jdbc.support.AbstractFa
 llbackSQLE xceptionTr anslator.translate(AbstractFa
 llbackSQLE xceptionTr anslator.java:84)
 at org.springfram ework.jdbc.support.AbstractFa
 llbackSQLE xception
```

  A log entry may be divided into multiple lines because the log contains error stacking information. Therefore, you must set a regular expression for the

beginning of a line. To extract each field, use a regular expression. The detailed configuration is as follows:

```
apiVersion :  log . alibabaclo  ud . com / v1alpha1
kind :  AliyunLogC  onfig
metadata :
 #  your   config   name ,  must   be   unique   in   your   k8s
cluster
  name :  regex - file - example
spec :
 #  logstore   name   to   upload   log
  logstore :  k8s - file
  logtailCon  fig :
   #  log   file ' s   input   type   is  ' file '
    inputType :  file
   #  logtail   config   name ,  should   be   same   with  [
metadata . name ]
    configName :  regex - file - example
    inputDetai  l :
     #  Set   logType   to  " common_reg  _log "  for   logs   of
the   regular   expression   type .
      logType :  common_reg  _log
     #  Log   file   folder
      logPath : / app / logs
     #  File   name ,  which   supports   wildcards ,  for
example ,  log_ *. log
      filePatter  n :  error . LOG
     #  Regular   expression   for   first   line
      logBeginRe  gex : '\[\ d +-\ d +-\ w +:\ d +:\ d +,\ d +]\ s
\[\ w +]\ s . *'
     #  Parse   the   regular   expression
      regex : '\[([^]]+)]\ s \[(\ w +)]\ s \[(\ w +)]\ s \[([^:]+):
(\ d +)]\ s (. *)'
     #  List   of   extracted   keys
      key  : [" time ", " level ", " method ", " file ", " line ",
" message "]
     #  Logs   in   regular   expression . ` time `  in   the
logs   are   extracted   for   time   parsing   by   default .  If
  time   is   not   required ,  ignore   the   field .
      timeFormat : '% Y -% m -% dT % H :% M :% S '
     #  Collect   files   from   the   container .  dockerFile
flag   is   set   to   true
      dockerFile :  true
     #  Only   collect   container   with  " ALIYUN_LOG
TAIL_USER_  DEFINED_ID "  in   docker   env   config
      dockerIncl  udeEnv :
```

```
          ALIYUN_LOG  TAIL_USER_  DEFINED_ID : ""
```

After the configuration is applied, the data collected by Log Service is as follows:

Figure 3-32: Collected log data



- **Delimiter pattern file**

  Logtail supports log parsing in delimiter mode, an example is as follows:

```
apiVersion :  log . alibabaclo  ud . com / v1alpha1
kind :  AliyunLogC  onfig
metadata :
 #  your   config   name ,  must   be   unique   in   your   k8s
cluster
  name :  delimiter - file - example
spec :
 #  logstore   name   to   upload   log
  logstore :  k8s - file
  logtailCon  fig :
   #  log   file ' s   input   type   is  ' file '
    inputType :  file
    configName :  delimiter - file - example
   #  logtail   config   name ,  should   be   same   with  [
metadata . name ]
    inputDetai  l :
     #  Set   logType   to   delimiter_  log   for   logs   of
the   delimiter   type
      logType :  delimiter_  log
     #  Log   file   folder
      logPath : / usr / local / ilogtail
     #  File   name ,  which   supports   wildcards ,  for
example ,  log_ *. log
      filePatter  n :  delimiter_  log . LOG
     #  Use   a   multi - character   delimiter
      separator : '|&|'
     #  List   of   extracted   keys
      key  : [' time ', ' level ', ' method ', ' file ', ' line ',
' message ']
     #  Keys   for   parsing   time .  Ignore   the   field   if
time   parsing   is   not   required
      timeKey : ' time '
     #  Time   parsing   method .  Ignore   the   field   if   time
  parsing   is   not   required
      timeFormat : '% Y -% m -% dT % H :% M :% S '
     #  Collect   files   from   the   container .  dockerFile
flag   is   set   to   true
      dockerFile :  true
```

```
      #  Only   collect   container   with  " ALIYUN_LOG
TAIL_USER_  DEFINED_ID "  in   docker   env   config
      dockerIncl  udeEnv :
        ALIYUN_LOG  TAIL_USER_  DEFINED_ID : ''
```

- **JSON mode file**

  If each data line in a file is a JSON object, you can use the JSON method for parsing, an example is as follows:

```
apiVersion :  log . alibabaclo  ud . com / v1alpha1
kind :  AliyunLogC  onfig
metadata :
 #  your   config   name ,  must   be   unique   in   you   k8s
cluster
  name :  json - file - example
spec :
 #  logstore   name   to   upload   log
  logstore :  k8s - file
  logtailCon  fig :
   #  log   file ' s   input   type   is  ' file '
    inputType :  file
   #  logtail   config   name ,  should   be   same   with  [
metadata . name ]
    configName :  json - file - example
    inputDetai  l :
     #  Set   logType   to   json_log   for   logs   of   the
delimiter   type
      logType :  json_log
     #  Log   file   folder
      logPath : / usr / local / ilogtail
     #  File   name ,  which   supports   wildcards ,  for
example ,  log_ *. log
      filePatter  n :  json_log . LOG
     #  Keys   for   parsing   time .  Ignore   the   field   if
time   parsing   is   not   required
      timeKey : ' time '
     #  Time   parsing   method .  Ignore   the   field   if   time
  parsing   is   not   required
      timeFormat : '% Y -% m -% dT % H :% M :% S '
     #  Collect   files   from   the   container .  dockerFile
flag   is   set   to   true
      dockerFile :  true
     #  Only   collect   container   with  " ALIYUN_LOG
TAIL_USER_  DEFINED_ID "  in   docker   env   config
      dockerIncl  udeEnv :
        ALIYUN_LOG  TAIL_USER_  DEFINED_ID : ""
```

3.6.6 Kubernetes-Sidecar log collection mode

Logtail can collect logs in Sidecar mode from Kubernetes and create a Sidecar container for each service container requiring log collection, thereby facilitating multi-tenant isolation and improving collection performance.

Currently, the default log component installed in Kubernetes clusters is DaemonSet , which simplifies O&M operations, occupies a few resources, supports collection of container stdout and container files, and can be flexibly configured.

However, in DaemonSet mode, Logtail needs to collect logs from all container on a node. This leads to a bottleneck in performance and does not allow for total isolation among service logs. To resolve the preceding issue, Logtail now provides the Sidecar mode, which enables Logtail to create a Sidecar container for each service container requiring log collection. This mode greatly strengthens multi-tenant isolation and improves the collection performance. We recommend that you use the Sidecar mode for large-scale Kubernetes clusters and for clusters that function as a PaaS platform to serve multiple services.

Features

- The Sidecar mode can be applied to Container Service for Kubernetes, on-premises ECS Kubernetes, and on-premises Kubernetes in IDCs.
- In Sidecar mode, Logtail can collect Pod metadata, including the Pod name, Pod IP address, Pod namespace, and name and IP address of the node to which the Pod belongs.
- In Sidecar mode, Logtail can automatically create Log Service resources through CustomResourceDefinition (CRD), including projects, Logstores, indexes, Logtail Configs, and machine groups.
- The Sidecar mode supports dynamic scaling. You can adjust the number of replicas at any time, and the changes take effect immediately.

Concepts

In Sidecar mode, log collection requires that Logtail share the log directory with the service container. Briefly, the service container writes logs into the log directory, and Logtail monitors changes on log files in the log directory and collects logs. For more information, see the following official documents:

1. *Introduction to the Sidecar log collection mode*
2. *Example of the Sidecar mode*

Prerequisites

1. You have activated Log Service.

   If you have not activated Log Service, *activate it*.
2. You have installed *Kubernetes log collection process* for CRD-based settings.

Limits

1. Logtail must share the log directory with the service container.
2. Sidecar mode does not support collection of container stdout.

Sidecar configuration

The Sidecar configuration involves:

1. *Setting basic operation parameters*
2. *Setting the mount path*

The following is an example:

```
apiVersion :  batch / v1
kind :  Job
metadata :
  name :  nginx - log - sidecar - demo
  namespace :  default
spec :
  template :
    metadata :
      name :  nginx - log - sidecar - demo
    spec :
      restartPol  icy :  Never
      containers :
     -  name :  nginx - log - demo
        image :  registry . cn - hangzhou . aliyuncs . com / log -
service / docker - log - test : latest
        command : ["/ bin / mock_log "]
        args : ["-- log - type = nginx ", "-- stdout = false ", "--
stderr = true ", "-- path =/ var / log / nginx / access . log ", "--
total - count = 1000000000 ", "-- logs - per - sec = 100 "]
        volumeMoun  ts :
       -  name :  nginx - log
          mountPath : / var / log / nginx
     #####  logtail   sidecar   container
     -  name :  logtail
       #  more   info :  https :// cr . console . aliyun . com /
repository / cn - hangzhou / log - service / logtail / detail
       #  this   images   is   released   for   every   region
        image :  registry . cn - hangzhou . aliyuncs . com / log -
service / logtail : latest
        livenessPr  obe :
          exec :
            command :
           - / etc / init . d / ilogtaild
           -  status
          initialDel  aySeconds :  30
          periodSeco  nds :  30
        resources :
          limits :
            memory :  512Mi
          requests :
            cpu :  10m
            memory :  30Mi
        env :
         #####  base   config
         #  user   id
```

```
          -  name : " ALIYUN_LOG  TAIL_USER_  ID "
             value : "${ your_aliyu  n_user_id }"
          #  user   defined   id
          -  name : " ALIYUN_LOG  TAIL_USER_  DEFINED_ID "
             value : "${ your_machi  ne_group_u  ser_define  d_id }"
          #  config   file   path   in   logtail ' s   container
          -  name : " ALIYUN_LOG  TAIL_CONFI  G "
             value : "/ etc / ilogtail / conf /${ your_regio  n_config
 }/ ilogtail_c  onfig . json "
          #####  env   tags   config
          -  name : " ALIYUN_LOG  _ENV_TAGS "
             value : " _pod_name_ | _pod_ip_ | _namespace  _ |
 _node_name  _ | _node_ip_ "
          -  name : " _pod_name_ "
             valueFrom :
               fieldRef :
                 fieldPath :  metadata . name
          -  name : " _pod_ip_ "
             valueFrom :
               fieldRef :
                 fieldPath :  status . podIP
          -  name : " _namespace  _ "
             valueFrom :
               fieldRef :
                 fieldPath :  metadata . namespace
          -  name : " _node_name  _ "
             valueFrom :
               fieldRef :
                 fieldPath :  spec . nodeName
          -  name : " _node_ip_ "
             valueFrom :
               fieldRef :
                 fieldPath :  status . hostIP
        volumeMoun  ts :
         -  name :  nginx - log
            mountPath : / var / log / nginx
      #####  share   this   volume
       volumes :
      -  name :  nginx - log
         emptyDir : {}
```

Configuration 1: Set basic operation parameters.

The following shows major parameters and their settings:

```
#####  base   config
          #  user   id
          -  name : " ALIYUN_LOG  TAIL_USER_  ID "
             value : "${ your_aliyu  n_user_id }"
          #  user   defined   id
          -  name : " ALIYUN_LOG  TAIL_USER_  DEFINED_ID "
             value : "${ your_machi  ne_group_u  ser_define  d_id }"
          #  config   file   path   in   logtail ' s   container
          -  name : " ALIYUN_LOG  TAIL_CONFI  G "
```

```
            value : "/ etc / ilogtail / conf /${ your_regio  n_config
 }/ ilogtail_c  onfig . json "
```

Parameter	Description
`${ your_regio n_config }`	This parameter is determined by the region and network type of the project. Set the parameter to an appropriate value according to the network type. Valid values: · For the Internet: `region - internet`. For example, the value for the China (Hangzhou) region is `cn - hangzhou - internet`. · For Alibaba Cloud intranet: `region`. For example, the value for the China (Hangzhou) region is `cn - hangzhou`. In this parameter, region is a *Table 1*. Set it to the region to which the project belongs.
`${ your_aliyu n_user_id }`	This parameter specifies the user ID, which must be replaced with your Alibaba Cloud account ID in string format. For more information about how to query your ID, see section 2.1 in *user ID configuration*. Note: This parameter value must be your Alibaba Cloud account ID. RAM user IDs do not take effect.
`${ your_machi ne_group_u ser_define d_id }`	This parameter specifies the custom ID of a machine group in your cluster. The ID must be unique within the region where Log Service is deployed. For more information, see *Create an ID to identify a machine group*.

Configuration 2: Set the mount path.

1. Logtail and the service container must be mounted to the same directory.
2. The `emptyDir` mount method is recommended.

The mount path example is shown in the preceding configuration example.

Log collection settings

Log collection can be set through CRD or the Log Service console. CRD-based settings support automatic creation of projects, Logstores, indexes, machine groups, and Logtail Configs, and can be easily integrated with Kubernetes. Therefore, CRD-based settings are recommended. Console-based settings are easier for users who debug or use Kubernetes log collection for the first time.

CRD-based settings

For more information, see *Configure Kubernetes log collection on CRD*. Compared with the DaemonSet collection mode, CRD-based settings are subjected to the following limits:

1. You must specify the name of the project requiring log collection. Otherwise, logs are collected and sent to the project where the log component is installed by default.
2. You must specify the machine group for the settings to take effect. Otherwise, the settings are applied to the machine group to which the DaemonSet belongs by default.
3. The Sidecar mode supports only file collection, during which, `dockerFile` must be set to false.

For more information, see the corresponding example.

Console-based settings

1. Configure the machine group.

   In the Log Service console, create a Logtail machine group with the identification set to a custom ID to dynamically adapt to changes of the Pod IP address. To do so, perform the following steps:

   a. Activate Log Service and create a project and a Logstore as needed. For more information, see *Preparation*.

   b. On the machine group list page, click Create Machine Group.

   c. Set the identification to the custom ID `ALIYUN_LOG_TAIL_USER_DEFINED_ID`.



2. Set the collection mode.

   Set collection details for the target file. Currently, various modes are supported, such as the simple mode, Nginx access mode, delimiter mode, JSON mode, and regular mode. For more information, see *Collect text logs*.

   The settings of this example is shown in the following figure.

    Note:

Docker File must be disabled.


* 配置名称:
nginx-log-sidecar
* 日志路径:
/var/log/nginx
/**/
access
指定文件夹下所有符合文件名称的文件都会被监控到(包含所有层次的目
符模式匹配。Linux文件路径只支持/开头，例：/apsara/nuwa/.../app.Lo
如：C:\Program Files\Intel\...*.Log
是否为Docker文件:
如果是Docker容器内部文件，可以直接配置内部路径与容器Tag，Logta
进行过滤采集指定容器的日志，具体说明参考文档链接
模式:
分隔符模式
如何设置Delimiter类型配置
日志样例:
2018-09-26T03:16:53.033307075Z 10.200.98.220 - - "POST /PutData
Category=YunOsAccountOpLog&AccessKeyId=Uxxxx45A&Date=Fri%
A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bm
18204 200 37 "-" "aliyun-sdk-java" 1
请贴入需要解析的日志样例(支持多条) 常见样例>>
* 分隔符:
空格
引用符:
双引号
双引号（"）作为Quote时，内部包含分隔符的字段需要被一对Quote包
包含空格、制表符等字符，请修改格式。

Examples

Scenario:

1. **The Kubernetes cluster is an on-premises cluster in an IDC, and the region where Log Service is deployed is China (Hangzhou). Logs are collected from the Internet.**
2. **In the following examples, the mount object is `nginx - log`, and the mount type is `emptyDir`. They are mounted to the `/ var / log / nginx` directory in the nginx-log-demo and logtail containers, respectively.**
3. **The access log is `/ var / log / nginx / access . log`, and the destination Logstore is `nginx - access`.**
4. **The error log is `/ var / log / nginx / error . log`, and the destination Logstore is `nginx - error`.**

· **Sidecar settings:**

```
apiVersion :  batch / v1
kind :  Job
metadata :
  name :  nginx - log - sidecar - demo
  namespace :  default
spec :
  template :
    metadata :
      name :  nginx - log - sidecar - demo
    spec :
      restartPol  icy :  Never
      containers :
     -  name :  nginx - log - demo
        image :  registry . cn - hangzhou . aliyuncs . com / log -
service / docker - log - test : latest
        command : ["/ bin / mock_log "]
        args : ["-- log - type = nginx ", "-- stdout = false ", "--
stderr = true ", "-- path =/ var / log / nginx / access . log ",
"-- total - count = 1000000000 ", "-- logs - per - sec = 100 "]
        volumeMoun  ts :
       -  name :  nginx - log
          mountPath : / var / log / nginx
     #####  logtail   sidecar   container
     -  name :  logtail
       #  more   info :  https :// cr . console . aliyun . com /
repository / cn - hangzhou / log - service / logtail / detail
       #  this   images   is   released   for   every   region
        image :  registry . cn - hangzhou . aliyuncs . com / log -
service / logtail : latest
        livenessPr  obe :
          exec :
            command :
           - / etc / init . d / ilogtaild
           -  status
          initialDel  aySeconds :  30
          periodSeco  nds :  30
        env :
         #####  base   config
```

```
          #  user   id
          -  name : " ALIYUN_LOG  TAIL_USER_  ID "
             value : " xxxxxxxxxx "
          #  user   defined   id
          -  name : " ALIYUN_LOG  TAIL_USER_  DEFINED_ID "
             value : " nginx - log - sidecar "
          #  config   file   path   in   logtail ' s   container
          -  name : " ALIYUN_LOG  TAIL_CONFI  G "
             value : "/ etc / ilogtail / conf / cn - hangzhou -
 internet / ilogtail_c  onfig . json "
          #####  env   tags   config
          -  name : " ALIYUN_LOG  _ENV_TAGS "
             value : " _pod_name_ | _pod_ip_ | _namespace  _ |
 _node_name  _ | _node_ip_ "
          -  name : " _pod_name_ "
             valueFrom :
               fieldRef :
                 fieldPath :  metadata . name
          -  name : " _pod_ip_ "
             valueFrom :
               fieldRef :
                 fieldPath :  status . podIP
          -  name : " _namespace  _ "
             valueFrom :
               fieldRef :
                 fieldPath :  metadata . namespace
          -  name : " _node_name  _ "
             valueFrom :
               fieldRef :
                 fieldPath :  spec . nodeName
          -  name : " _node_ip_ "
             valueFrom :
               fieldRef :
                 fieldPath :  status . hostIP
        volumeMoun  ts :
        -  name :  nginx - log
           mountPath : / var / log / nginx
      #####  share   this   volume
       volumes :
      -  name :  nginx - log
         emptyDir : {}
```

- **CRD settings:**

```
#  config   for   access   log
 apiVersion :  log . alibabaclo  ud . com / v1alpha1
 kind :  AliyunLogC  onfig
 metadata :
  #  your   config   name ,  must   be   unique   in   you   k8s
 cluster
   name :  nginx - log - access - example
 spec :
  #  project   name   to   upload   log
   project :  k8s - nginx - sidecar - demo
  #  logstore   name   to   upload   log
   logstore :  nginx - access
  #  machine   group   list   to   apply   config ,  should   be
 same   with   your   sidecar ' [ ALIYUN_LOG  TAIL_USER_  DEFINED_ID
 ]
   machineGro  ups :
  -  nginx - log - sidecar
  #  logtail   config   detail
   logtailCon  fig :
```

```
    #  log   file ' s   input   type   is  ' file '
     inputType :  file
    #  logtail   config   name ,  should   be   same   with  [
 metadata . name ]
     configName :  nginx - log - access - example
     inputDetai  l :
      #  Simple   logs   with   logType   set   to   common_reg
 _log
       logType :  common_reg  _log
      #  Log   folder
       logPath : / var / log / nginx
      #  File   name   with   wildcards   supported ,  for   example
 ,  log_ *. log
       filePatter  n :  access . log
      #  Sidecar   mode   with   dockerFile   set   to   false
       dockerFile :  false
      #  Line   start   regular   expression ,  which   is   set
 to  .*  is   the   log   contains   only   a   line
       logBeginRe  gex : '. *'
      #  Regular   expression   for   parsing
       regex : '(\ S +)\ s (\ S +)\ s \ S +\ s \ S +\ s "(\ S +)\ s
 (\ S +)\ s +([^"]+)"\ s +(\ S +)\ s (\ S +)\ s (\ d +)\ s (\ d +)\
 s (\ S +)\ s "([^"]+)"\ s . *'
      #  List   of   the   extracted   keys
       key  : [" time ", " ip ", " method ", " url ", " protocol ",
 " latency ", " payload ", " status ", " response - size ", ser -
 agent "]
#  config   for   error   log
```

```
#  config   for   error   log
 apiVersion :  log . alibabaclo  ud . com / v1alpha1
 kind :  AliyunLogC  onfig
 metadata :
  #  your   config   name ,  must   be   unique   in   you   k8s
 cluster
   name :  nginx - log - error - example
 spec :
  #  project   name   to   upload   log
   project :  k8s - nginx - sidecar - demo
  #  logstore   name   to   upload   log
   logstore :  nginx - error
  #  machine   group   list   to   apply   config ,  should   be
 same   with   your   sidecar ' [ ALIYUN_LOG  TAIL_USER_  DEFINED_ID
 ]
   machineGro  ups :
  -  nginx - log - sidecar
  #  logtail   config   detail
   logtailCon  fig :
    #  log   file ' s   input   type   is  ' file '
     inputType :  file
    #  logtail   config   name ,  should   be   same   with  [
 metadata . name ]
     configName :  nginx - log - error - example
     inputDetai  l :
      #  Simple   logs   with   logType   set   to   common_reg
 _log
       logType :  common_reg  _log
      #  Log   folder
       logPath : / var / log / nginx
      #  File   name   with   wildcards   supported ,  for   example
 ,  log_ *. log
       filePatter  n :  error . log
      #  Sidecar   mode   with   dockerFile   set   to   false
```

```
        dockerFile :  false
```

- View log collection results

  After the preceding settings are applied to the Kubernetes cluster, the Logtail container automatically creates the corresponding project, Logstore, machine group, and Logtail Config, and automatically sends the collected logs to Log Service. You can log on to the Log Service console to view details.

3.7 Limits

Table 3-4: Limits on file collection

Item	Capabilities and limits
File encoding	Log files encoded in UTF-8 and GBK are supported. Log files encoded in other formats result in undefined behaviors such as gibberish and data loss. We recommend that you use UTF-8 encoding for better processing performance.
Log file size	Unlimited.
Log file rotation	Both `. log *` and `. log` are supported.
Log collection behavior upon log parsing block	When block occurs in log parsing, Logtail keeps the open status of the log file FD. If log file rotation occurs multiple times during the block, Logtail attempts to keep the log parsing sequence of each rotation. If the number of unparsed log rotations is more than 20, Logtail does not process subsequent log files. Soft link support More information, see here.
Single log size	Monitored directories can be soft links.

Item	Capabilities and limits
Single log size	The size of a single log cannot exceed 512 KB. If multiple-line logs are divided by a regular expression, the maximum size of each log is still 512 KB. If the log size exceeds 512 KB, the log is forced to be divided into multiple parts for collection. For example, a log is 1025 KB. The first 512 KB is processed for the first time, the subsequent 512 KB is processed for the second time, and the last 1 KB is processed for the third time.
Regular expression type	Use regular expressions that are compatible with Perl.
Multiple collection configurations for the same file	Not supported. We recommend that you collect log files to a Logstore and configure multiple subscriptions. If this function is required, configure a soft link for the log file to bypass this limit.
File opening behavior	Logtail keeps a file to be collected in the open status. Logtail closes the file if the file does not have any modification within five minutes.
First log collection behavior	Logtail only collects incremental log files. If modifications are found in a file for the first time and the file size exceeds 1 MB, Logtail collects the logs from the last 1 MB. Otherwise, Logtail collects logs from the beginning. If a log file is not modified after the configuration is issued, Logtail does not collect this file.
Non-standard text log	For a row containing ‘\0’ in the log. The log is truncated to the first ‘\0’.

Table 3-5: Checkpoint management

Item	Capabilities and limits
Checkpoint timeout period	If the file has not been modified for more than 30 days, the Checkpoint is deleted.

Item	Capabilities and limits
Checkpoint storage policy	Regular save every 15 minutes, automatically saved when the program exits.
Checkpoint save path	The default save path is `/ tmp / logtail_ch eckpoint` , you can modify the parameters according to *Configure startup parameters*.

Table 3-6: Limits on configuration

Item	Capabilities and limits
Configuration update	Your updated configuration takes effect with a delay of about 30 seconds.
Dynamic configuration loading	Supported. The configuration update does not affect other collections.
Number of configurations	Theoretically unlimited. We recommend that the number of collection configurations for a server is no more than 100.
Multi-tenant isolation	The isolation between collection configurations.

Table 3-7: Limits on resources and performance

Item	Capabilities and limits
Log processing throughput	The default limit to raw log traffic is 2 MB/s. Data is uploaded after being encoded and compressed, generally with a compression ratio of 5–10 times. Logs may be lost if the log traffic exceeds the limit. To adjust the parameter, see *Configure startup parameters* Configure startup parameters.
Maximum performance	In case of single core, the maximum processing capability is 100 MB/s for logs in simple mode, 20 MB/s by default for logs in full mode (depending on the complexity of the regular expression), 40 MB/s for logs in delimiter mode, and 30 MB/s for logs in JSON mode. Enabling multiple log processing threads improves the performance by 1.5–3 times.

Item	Capabilities and limits
Number of monitored directories	Logtail actively limits the depth of monitored directories to conserve your resources. If the upper limit is reached, Logtail stops monitoring more directories and log files. Logtail monitors at most 3,000 directories (including subdirectories).
Default resource limit	By default, Logtail occupies up to 40% of CPU usage and 256 MB of memory usage. If logs are generated at a high speed, you can adjust the parameter by following the *Configure startup parameters* Configure startup parameters.
Processing policy for resource limit exceeding	If the resources occupied by Logtail in 3 minutes exceed the upper limit, Logtail is forced to restart, which may cause loss or duplication of data.

Table 3-8: Limits on error handling

Item	Capabilities and limits
Network error handling	If the network connection is abnormal, Logtail actively retries and automatically adjusts the retry interval.
Handling of resource quota exceeding	If the data transmission rate exceeds the maximum quota of Logstore, Logtail blocks log collection and automatically retries.
Maximum retry period for timeout	If data transmission fails for more than 6 successive hours, Logtail discards the data.
Status self-check	Logtail automatically restarts in the case of an exception, for example, abnormal exit of a program or resource limit exceeding.

Table 3-9: Other limits

Item	Capabilities and limits
Log collection delay	Except for block status, the delay in log collection by Logtail does not exceed one second after logs are flushed to a disk.

Item	Capabilities and limits
Log uploading policy	Logtail automatically aggregates logs in the same file before uploading them. Log uploading is triggered in the condition that more than 2,000 logs are generated, the log file exceeds 2 MB, or the log collection exceeds 3 seconds.

4 Cloud product collection

4.1 Cloud product logs

Log Service can collect logs from various cloud products, such as ECS, OSS, and SLB. The logs record product information including operation information, operation statuses, and service statuses.

The following table lists the cloud products from which Log Service can collect logs:

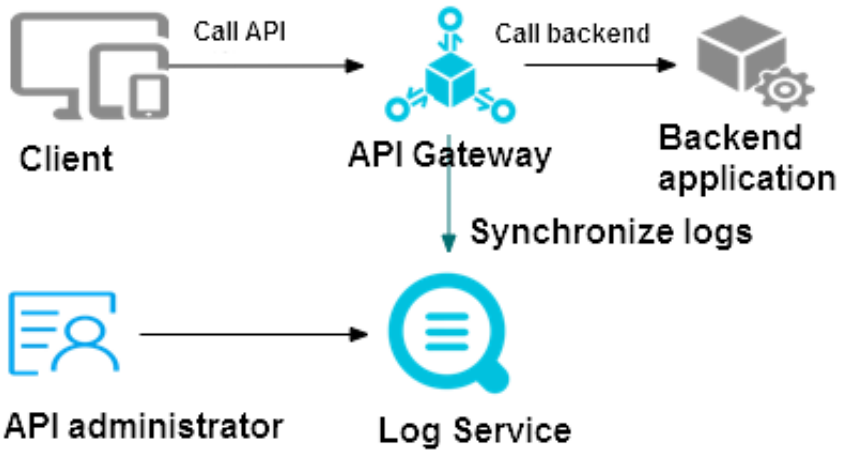
Type	Cloud product name	Activation method	Details
Elastic computing	Elastic Compute Service (ECS)	Through Logtail installation	*Logtail introduction*
	Container Service/ Container Service for Kubernetes	Through the Container Service console	*Text logs* and *stdout*
Storage	Object Storage Service (OSS)	Through the OSS console	*OSS access logs*
Network	Server Load Balancer (SLB)	Through the SLB console	*Access logs of Layer-7 SLB*
	Virtual Private Cloud (VPC)	Through the VPC console	*Flow logs*
	API Gateway	Through the API Gateway console	*API Gateway access logs*
Security	ActionTrail	Through the ActionTrail console	*ActionTrail overview*
	DDoS Protection	Through the DDoS Protection console	*DDOS Protection overview*
	Threat Detection Service	Purchase Threat Detection Service Enterprise Edition and activate the service in the Threat Detection Service console.	*Log retrieval*

Type	Cloud product name	Activation method	Details
	Anti-Bot Service	Through the Anti-Bot Service console	*Anti-Bot Service logs*
Application	Log Service (LOG)	Through the Log Service console	*Log Service overview*

4.2 API Gateway Access Log

Alibaba Cloud API Gateway provides API hosting service to facilitate micro-service aggregation, frontend/backend isolation, and system integration. An access log is a log generated by Web services. Each API request corresponds to an access record, containing caller IP, requested URL, response latency, returned status code, number of bytes for each request and response, and other information. With the preceding information, you can understand the operation status of your Web services.

Figure 4-1: API gateway



With Log Service, you can collect access logs of the API Gateway by using Data Import Wizard.

Data migration

1. Online log query: You can perform a rapid accurate or fuzzy search using any keyword in the log. This feature can be used to locate a problem or count queries.
2. Detailed call logs: You can search for details of API call logs.
3. Customized analysis chart: You can customize any log item into a statistical chart according to the statistical requirement to meet your business needs.

4. Preset analysis report: In the API Gateway, some global statistical charts are predefined, including request volume, success rate, failure rate, latency, the number of applications that call APIs, failure statistics, Top grouping, Top API, and Top latency.

Field Description

Log Fields	Description
apiGroupUid	The API group ID.
apiGroupName	API group name
apiUid	The API ID.
apiName	The API name.
apiStageUid	The API stage ID.
apiStageName	The API stage name.
httpMethod	The called HTTP method.
path	The requested path.
domain	The called domain name.
statusCode	The HTTP status code.
ErrorMessage	Error message
appId	The application ID of the caller.
appName	The application name of the caller.
clientIp	The client IP of the caller.
Exception	The specific error message returned by backend.
providerAliUid	The account ID of the API provider.
region	such as cn-hangzhou
requestHandleTime	The request time (GMT).
RequestId	The request ID, which is globally unique.
requestSize	The size of the returned data (in bytes).
Responsesize	The size of the returned data (in bytes).
Servicelatency	The backend latency (in milliseconds).

Procedure

1. Create a project and a Logstore.

   For how to create a project and a Logstore, see *Preparation*.

   Skip this step if a Logstore already exists.
2. Enter the data access wizard.

   After creating a Logstore, click the Data Import Wizard icon on theLogstore List page .
3. Select a data source.

   Click API Gateway in Cloud Services, and then click Next to go to the Configure Data Source step.
4. Configure the data source.

   In the Configure Data Source step, check whether you have completed the following configurations:

   a. Activate the API Gateway service.

      API Gateway provides a complete API hosting service, helping you open capabilities, services, and data to your partners in the form of API.

      If you have not activated the API Gateway service, activate it as instructed on the relevant page.

   b. Complete Resource Access Management (RAM) authorization.

      Authorize Log Service by using RAM before establishing a dispatch rule, so that Log Service can collect your API Gateway logs.

      Click Authorize in the upper-right corner for quick authorization.

   c. Establish a dispatch rule.

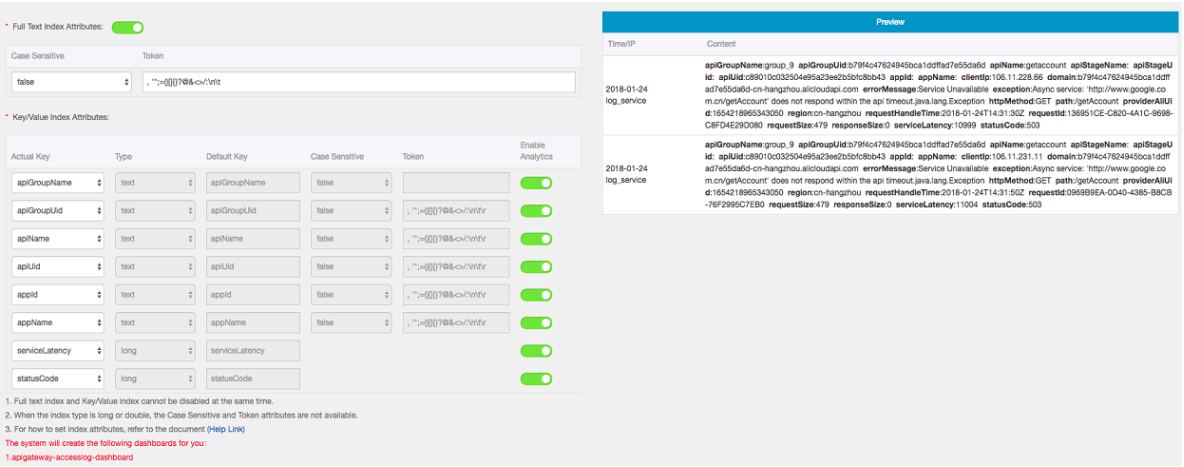
      If you do this for the first time, the system automatically imports API Gateway logs and establishes a dispatch rule. If you have configured API Gateway log collection before, a message indicating the log dispatch rule already exists is displayed. You can select to delete the existing dispatch rule.

   Click Next to enter the Search, Analysis, and Visualization page.

5. Configure Search, Analysis, and Visualization.

   Configure indexes as shown in the following figure. The configuration of the indexes is related to your log search and analysis efficiency.You will also use this configuration in Dashboard, so proceed with caution when modifying this configuration.

   Figure 4-2: Configure Indexes



   Click Next to complete the configuration. Log shipper can be configured separately when necessary.

You have finished the data import wizard initialization. You can select the configured Logstore api-gateway-access-log to query and analyze logs, or go to Dashboard to view reports.

4.3 Access logs of Layer-7 Server Load Balancer

Alibaba Cloud Server Load Balancer can distribute traffic for multiple Elastic Compute Service (ECS) instances, and support Layer-4 Server Load Balancer based on TCP and Layer-7 Server Load Balancer based on HTTP/HTTPS. By using Server Load Balancer, the impact on the business is reduced when a single ECS instance has an exception so that the system availability is enhanced. Working with the dynamic expansion and contraction of Auto Scaling, backend servers can respond to the changes of business traffic quickly.

Each access request to Server Load Balancer records the access logs. The access logs collect the details of all the requests sent to Server Load Balancer, including request time, client IP address, latency, request path, and server response. As an Internet

access point, Server Load Balancer hosts a large number of access requests. By using the access logs, you can analyze the user behavior on the client, the geographical distribution of the client users, and troubleshoot the issues.

Use Log Service to collect the Server Load Balancer access logs. You can monitor, probe, diagnose, and report the Layer-7 access logs of HTTP/HTTPS continuously and understand Server Load Balancer instances more comprehensively.



Note:

Only Layer-7 Server Load Balancer supports the access logs function. The access logs function is available in all regions. For more information, see *Configure access logs*.

Function advantages

- Simple. Free developers and maintenance staff from tedious and time-consuming log processing so that they can concentrate on business development and technical research.
- Massive. Access logs are proportional to request PVs of Server Load Balancer instances. The data size is usually large. Therefore, the performance and cost issues must be considered when processing access logs. Log Service can analyze 100 million logs in a second and has obvious cost advantages compared with the open-source solutions.
- Real-time. Scenarios such as DevOps, monitoring, and alerting require real-time log data. Traditional data storage and analysis tools cannot meet this requirement. For example, it takes long time to ETL data to Hive at which a lot of work is spent on data integration. Powered by its powerful computing capability, Log Service can process and analyze access logs in seconds.
- Flexible. You can enable or disable the access log function at the level of Server Load Balancer instance. You can enable or disable the access log function at the level of Server Load Balancer instance. Additionally, you can set the storage period (1–365 days) and the Logstore capacity of logs is dynamically scalable to meet business growth requirements.

Configure Log Service to collect Layer-7 Server Load Balancer access logs

Prerequisites

1. You have activated Server Load Balancer and Log Service. The created *Create an SLB instance*, Log Service project, and Logstore are in the same region.

   > Note:
   > Only Layer-7 Server Load Balancer supports the function of access logs. For the available regions, see Access logs.

2. If you are a RAM user, you must be authorized to use the SLB access logging. For more information, see *Authorize a RAM user to configure access logs*.

Procedure

1. Log on to the Log Service console.
2. After project and Logstore are created, follow the page prompts to enter the data import wizard. You can also click the Data Import Wizard icon on the Logstore List page to enter the configuration process.
3. Select a data source.

   Click Server Load Balancer in Cloud Services and then click Next.
4. RAM authorization.

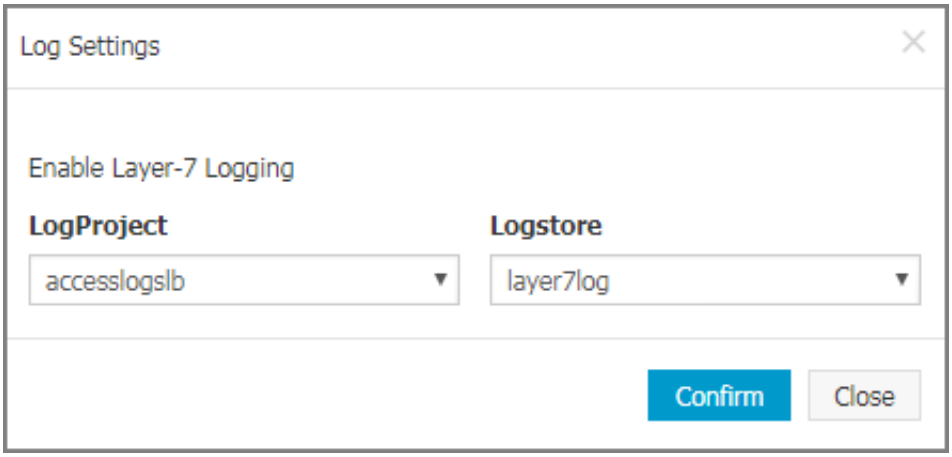
   Click Authorize as instructed on the page. Then, click Confirm Authorization Policy to authorize Server Load Balancer to access Log Service.
5. Set dispatch rule. Click Dispatch configuration to go to the Server Load Balancer console.

   a. ClickLogs > Access Login the left-side navigation pane.
   b. Click Configure at the right of the Server Load Balancer instance.

   > Note:

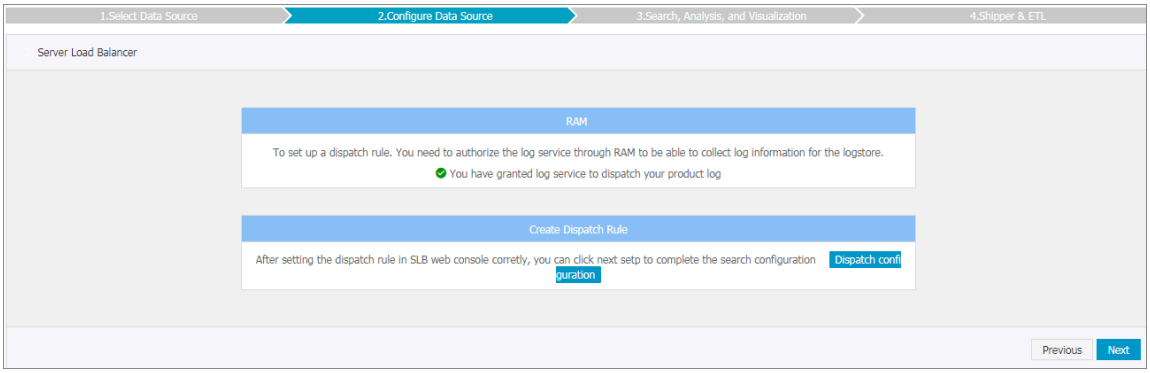
Make sure the Log Service project and the SLB instance are in the same region.

Figure 4-3: Log Settings



c. Select the project and Logstore of Log Service. Then, click Confirm.
d. After the configuration is complete, close the dialog box. Return to the data import wizard and click Next.

Figure 4-4: Configure Data Source



6. Search, analysis, and visualization.

   Log Service presets the query indexes required by Server Load Balancer. For the field descriptions, see Field description in this document. Click Next.

   Note:
   The dashboard {LOGSTORE}-slb_layer7_access_center and {LOGSTORE}-slb_layer7_operation_center are created by default. After the configuration is complete, you can view it on the Dashboard page.

7. Click Confirm to complete the data access.

Subsequent operations

- Query logs in real time

  You can perform a rapid accurate or fuzzy *query* by using any keyword in the log. This feature can be used for problem location or statistical query.

- Preset analysis reports

  Server Load Balancer predefines some global statistics graphs, including Top client access, distribution of request status codes, Top URI access, traffic trend of request messages, and statistics of RealServer response time.

- Customize analysis charts

  You can perform an ad-hoc query for any log item according to the statistical requirement and save the results as a chart to meet your daily business requirements.

- Set log monitoring alarms

  You can perform customized analysis on Server Load Balancer request logs and save the results as a quick query. Set the quick query as an alarm. When the computing results of real-time logs exceed the defined threshold, the system sends an alarm notification.

Field descriptions

Field	Description
body_bytes_sent	The size of the HTTP body (in bytes) sent to the client.
client_ip	The request client IP.
host	The host is obtained from the request parameters first. If no value is obtained, obtain the host from the host header. If the value is still not obtained, use the backend server IP of the processing request as the host.
http_host	The host header contents in the request message.
http_referer	The HTTP referer header contents in the request message received by the proxy.

Field	Description
http_user_agent	The HTTP user-agent header contents in the request message received by the proxy.
http_x_forwarded_for	The x-forwarded-for contents in the request message received by the proxy.
http_x_real_ip	The real client IP.
read_request_time	The time (in milliseconds) for the proxy to read request.
request_length	The length of the request message, including startline, HTTP header, and HTTP body.
request_method	The method of the request message.
Request_time	The interval (in seconds) between the time when proxy receives the first request message and the time when proxy returns the response.
request_uri	The URI of the request message received by the proxy.
scheme	The request schema (http or https).
server_protocol	The HTTP protocol version received by the proxy. For example, HTTP/1.0 or HTTP/1.1.
slb_vport	The listening port of Server Load Balancer.
slbid	The Server Load Balancer instance ID.
ssl_cipher	The used cipher, such as ECDHE-RSA-AES128-GCM-SHA256/.
ssl_protocol	The protocol used to establish the SSL connection, such as TLSv1.2.
status	The status of proxy responding to the message.
tcpinfo_rtt	The tcp rtt time (in microseconds) on the client.
time	The log recorded time.

Field	Description
Upstream_addr	The IP address and port of the backend server.
upstream_response_time	The time (in seconds) during which Server Load Balancer establishes a connection on the backend, receives the data, and closes the connection.
upstream_status	The response status code of the backend server received by the proxy.
vip_addr	The vip address.
write_response_time	The time (in milliseconds) for the proxy to write responses.

4.4 DDoS log collection

4.4.1 Overview

Alibaba Cloud Anti-DDoS Pro is a paid service for Internet servers (including non-Alibaba Cloud hosts). To avoid the risk of service unavailability after large traffic DDoS attack, paid service can be applied. Configure Anti-DDoS Pro, and drain the attack traffic for high IP protection to ensure that the source is stable and reliable.

Background information

The security of the Internet community has been constantly facing challenges. Network threats represented by DDoS attacks have a serious impact on the network security.

DDoS attacks are moving towards large-scale, mobile and global development. According to recent survey reports, the frequency of DDoS attacks is on the rise. The hacker attacks are concealed, and can control a large number of cloud service providers with poor security measures, IDC, and even massive cameras to launch attacks. The attacks have formed a mature black industry chain, which getting more organized. At the same time, the attack mode develops toward polarization, and the proportion of slow attacks, mixed attacks, especially CC attacks increases, which makes the detection of the defense more difficult. The peak of attacks exceeding 1Tbps are common , and the number of 100 GB attacks has doubled. However, application layer attacks are also increasing significantly.

According to *Kaspersky 2018Q1 DDoS Risk Report*, China remains the main source of DDoS attacks and targets. The main industries that have being attacked are Internet, games, software, and finance companies. More than 80% of DDoS attacks mix HTTP and CC attacks, and have a high level of concealment. Therefore, it is especially important to analyze the access and attack behavior by using logs, and apply a protection strategy.

Log Service supports real-time collection of *Alibaba Cloud Anti-DDoS Pro* website access logs, CC attack logs, and supports real-time query and analysis of collected log data. The results of the query are displayed in the form of dashboards.

Functional advantages

- Simple configuration: Easily configure to capture real-time protected logs.
- Real-time analysis: Relying on Log Service, it provides real-time log analysis and out-of-box report center, that gives information about CC attack status and customer access details.
- Real-time alarms: Supports custom monitoring and alarms based on specific indicators in real time to provide timely response to critical business exceptions.
- Ecosystem: Supports the docking of other ecosystems, such as stream computing, cloud storage, and visualization solutions for the further data value exploration.
- FreeTier quota: Provides a free data import quota, and three days free log storage, query and real-time analysis. You can freely expand your storage time for compliance management, tracing, and filing. Support unlimited storage time, and the storage cost is 0.35 USD/GB per month.

Limits and instructions

- Exclusive Logstores do not support writing additional data.

  Exclusive Logstore is used to store Anti-DDoS Pro website logs, so writing other data is not supported. There are no restrictions on other functions such as query, statistics, alarms, and streaming consumption.
- Pay-As-You-Go billing method If DDoS log collection protection is not enabled for any website, no charge appears.

  DDOS log collection function is billed according to the charge item of Log Service. If DDoS log collection function is not enabled for any website, no charge appears. Log Service supports Pay-As-You-Go billing method, and provides FreeTier quota. For more information, see *Billing method*.

Scenarios

- **Troubleshoot website access exceptions**

  Log Service has been configured to collect DDoS logs, you can query and analyze the collected logs in real time. Using SQL statement to analyze the DDoS access log, you can quickly check and analyze the website access exceptions, and view information such as read and write delays and operator distribution.

  For example, view the DDoS access log by using the following statement:

```
__topic__ : ddos_acces s_log
```

- **Track CC attack source**

  The distribution and source of CC attacks are recorded in the DDoS access log. By performing real-time query and analysis on the DDoS access log, you can conduct source tracking, trace CC attacks, and provide a reference for response strategy.

  For example, analyze the CC attack country distribution recorded in the DDoS access log by the following statement:

```
__topic__ : ddos_acces s_log and cc_blocks > 0 | SELECT
  ip_to_coun try ( if ( real_clien t_ip ='-', remote_add r ,
real_clien t_ip )) as country , count ( 1 ) as " number of
  attacks " group by country
```

- **For example, view the PV access by the following statement:**

```
__topic__ : ddos_acces s_log | select count ( 1 ) as PV
```

- **Website operation analysis**

  DDoS access log records the website access data in real time. You can perform SQL query analysis of the collected access log data to obtain real-time access status, such as determining the website popularity, the source and channel of the access, the client distribution, and assist in website operation analysis.

  For example, view the visitor traffic distribution from different network clouds:

```
__topic__ : ddos_acces s_log | select ip_to_prov ider ( if (
real_clien t_ip ='-', remote_add r , real_clien t_ip )) as
provider , round ( sum ( request_le ngth )/ 1024 . 0 / 1024 . 0
, 3 ) as mb_in group by provider having ip_to_prov
```

```
ider ( if ( real_clien  t_ip ='-',  remote_add  r ,  real_clien
t_ip )) <> ''  order   by   mb_in   desc   limit   10
```

4.4.2 Collection procedure

In the Anti-DDoS Pro console, you can enable DDos log collection function for the website.

Prerequisites

1. Enable Anti-DDoS Pro function, purchase Anti-DDoS Pro instances, and *Online configuration*.
2. Enable Anti-DDoS Pro function, purchase Anti-DDoS Pro instances.
3. Activate Log Service.

Context

Log Service supports real-time collection of Alibaba Cloud Anti-DDoS Pro website access logs, CC attack logs, and supports real-time query and analysis of collected log data. The results of the query are displayed in the form of dashboards, and logs are used to analyze the access and attack behavior in real time, and assist the security department to formulate a protection strategy.

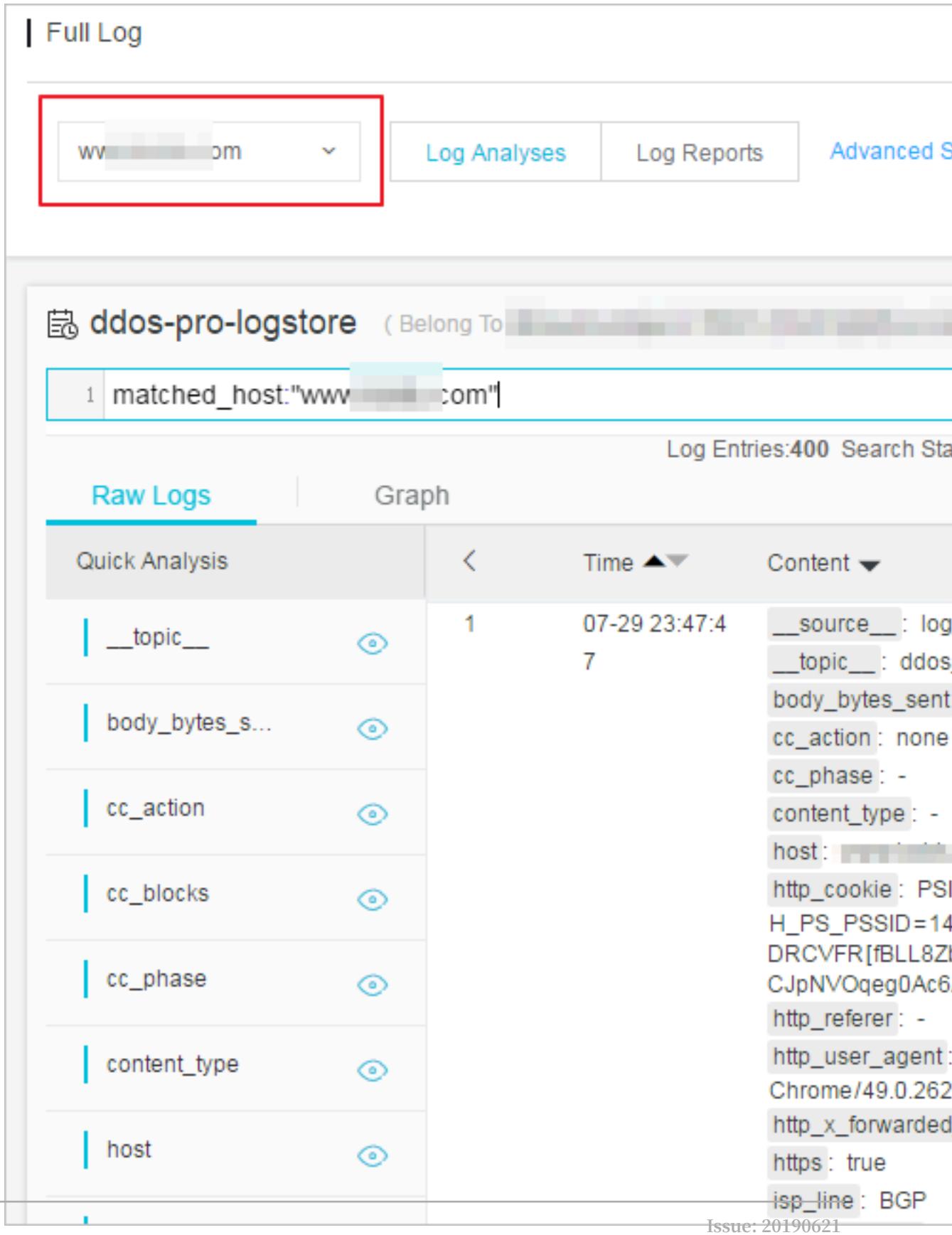
Procedure

1. Log on to the Anti-DDoS Pro console and select Log > Full Log in the left-side navigation pane. Enter the Full Log page.
2. If you are configuring DDoS log collection for the first time, follow the instructions on the page.

   DDoS has permission to distribute DDoS logs to your Logstore after authorization.

3. Select the website for which you want to enable DDoS log collection function and make sure the Status is on.

Figure 4-5: Enable the function



At this point, you have successfully enabled DDoS log collection for the current website. Log Service automatically creates a Logstore under your account. DDoS imports all the logs of the website that have this feature enabled into this Logstore. For Logstore default configurations, see *Default configuration*.

Table 4-1: Default configuration

Default configuration item	Configuration content
Project	By default, `ddos - pro - logstore` project is created.
Logstore	By default, Logstore is created. Logstore name is determined by the domain of the DDoS you purchased. · DDoS instances in mainland China: `ddos-pro-project-Alibaba Cloud Account ID-cn-hangzhou`. · Other DDoS instances: `ddos-pro-project-Alibaba Cloud Account ID-ap-southeast-1` All logs generated by the DDoS log collection function are saved in this Logstore.
Region	· If the DDoS region is in mainland China, the default project is saved in China East 1. · If the DDoS region is outside mainland China, the default project is saved in Asia Pacific SE 1.
Shard	By default, two shards are created and the *Auto split shard* feature is turned on.
Log storage time	The default storage time is three days, within the free quota. After three days logs are automatically deleted. For longer storage time, you can customize the configurations. For more information, see the How to modify the storage time of the website log section in *Billing method*.

Default configuration item	Configuration content
Dashboard	By default, two dashboards are created: · `ddos - pro - logstore_ ddos_operation_center`: Operation center · `ddos - pro - logstore_ ddos_access_center`: Access center For more information about dashboards, see *Log Report*.

You can query and analyze the collected logs in real time on the currentFull Log page. See the following figure for a log field description. In addition, Log Service creates two DDoS Operation center and Access center dashboards. You can also customize the dashboard configurations.

Field	Description	Example
__topic__	The topic of the log is fixed to ddos_access_log.	-
body_bytes_sent	Request to send the size of the Body. The unit is byte .	2
content_type	Content type.	application/x-www-form-urlencoded
host	Source website.	api.zhihu.com
http_cookie	Request cookie.	k1=v1;k2=v2
http_referer	Request referer. If none, the - is displayed.	http://xyz.com
http_user_agent	User agent request.	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build/HUAWEIEDISON-AL10)
http_x_forwarded_for	The upstream user IP that is redirected by the proxy .	-

Field	Description	Example
https	Whether the request is an HTTPS request, wherein: · true: the request is an HTTPS request. · false: the request is an HTTP request.	true
matched_host	The source website of the matching configuration may be a pan-domain name. If not matching, the - is displayed.	*.zhihu.com
real_client_ip	Access the customer real IP. If not available, the - is displayed.	1.2.3.4
isp_line	Line information, such as BGP, telecommunication , Unicom.	Telecommunication
remote_addr	Request client IP connection.	1.2.3.4
remote_port	Request client port connection.	23713
request_length	The length of the request. The unit is byte.	123
request_method	The HTTP request method .	GET
request_time_msec	Request time. The unit is microsecond.	44
request_uri	Request path.	/answers/377971214/ banner
server_name	The matching host name. If not matching, the `default` is displayed.	api.abc.com
status	HTTP status code.	200
time	Time.	2018-05-02T16:03:59+08: 00

Field	Description	Example
cc_action	CC protection policy, such as none, challenge, pass, close, captcha, wait, logon, n.	close
cc_blocks	Indicates whether CC protection is blocked, wherein: · 1: Blocked. · Other codes: Passed.	1
cc_phase	CC protection policy, including seccookie, server_ip_blacklist, static_whitelist, server_header_blacklist, server_cookie_blacklist, server_args_blacklist, qps_overmax.	server_ip_blacklist
ua_browser	Browser.	ie9
ua_browser_family	Browser series.	Internet explorer
ua_browser_type	Browser type.	web_browser
ua_browser_version	Browser version.	9.0
ua_device_type	Client device type.	computer
ua_os	Client operating system.	windows_7
ua_os_family	Client operating system series.	windows
upstream_addr	Return source address list, the format is `IP : Port`. Multiple addresses are separated by commas.	1.2.3.4:443
upstream_ip	The actual return source address IP.	1.2.3.4
upstream_response_time	The response time of the source. The unit is second.	0.044

Field	Description	Example
upstream_status	Return source request HTTP status.	200
user_id	Alibaba Cloud user ID.	12345678

What's next

- Click Log Analysis, *Query Analysis* on the collected log data.
- Click Log Report to view the built-in *dashboard*.
- Click Advanced Management to go to Log Service console to query and collect statistics, stream consumption, and set alarms for the collected log data.

4.4.3 Log analysis

Anti-DDoS Pro is embedded in the Full log page of Log Service in the Log analysis and Log report. After you have enabled the DDoS log protection function for a specific website, you can query and analyze the collected log data in real time on the current page, view or edit the dashboard, and set monitoring alarms.

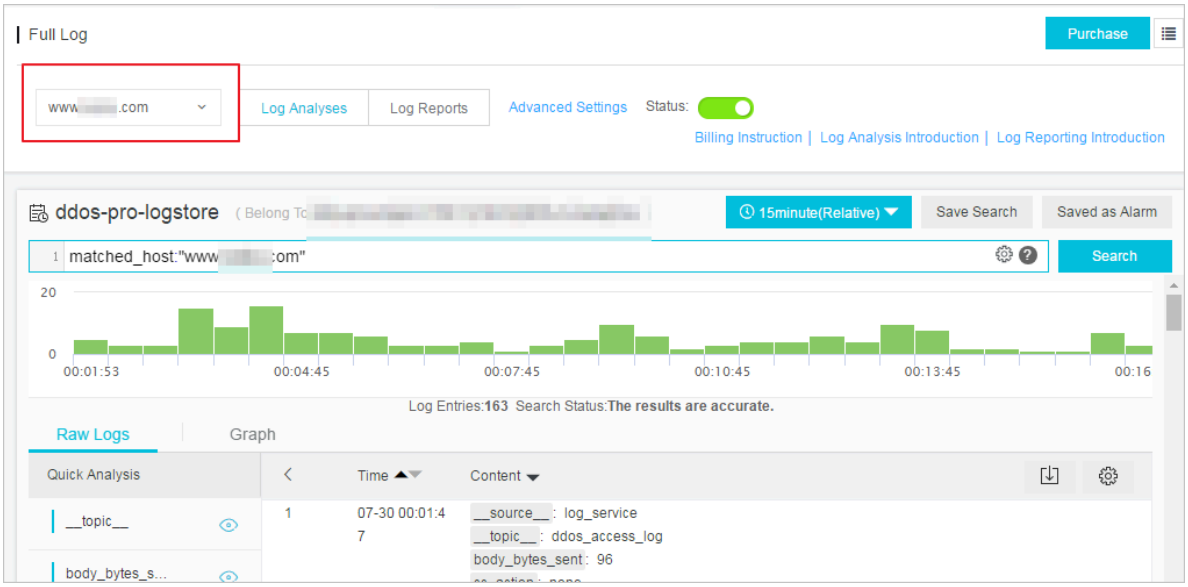
Procedure

1. Log on to the Anti-DDoS Pro console, and select Log > Full log in the left-side navigation pane.
2. Select the website for which you want enable DDoS log collection protection, then confirm the Status is on.

3. Click Log analysis.

   The current page is embedded in the Query analysis page of Log Service, and the system automatically enters the query statement for you, such as `matched_ho st : www . aliyun . com`, to view the log data based on the selected website.

   Figure 4-6: Log analysis

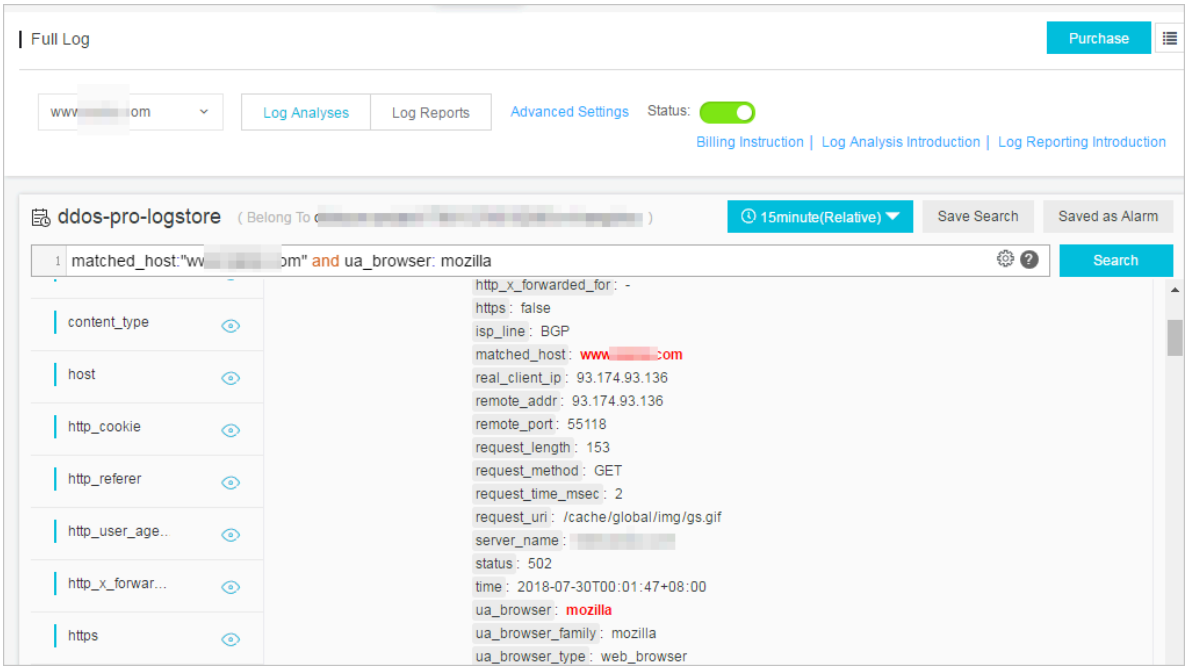


4. Enter query analysis statement, select the log time range and click Query.

   Note:

The default storage time of DDoS logs is three days. After three days, the log data is deleted. By default, you can only query log data for the past three days. To modify the log storage time, see *Modify log storage time*.

Figure 4-7: Log query



On the Query and Analysis page, you can also perform the following operations.

- Custom query and analysis

  Log Service provides different query and analysis syntaxes to support log queries in various complex scenarios. For more information, see *Custom query and analysis*.

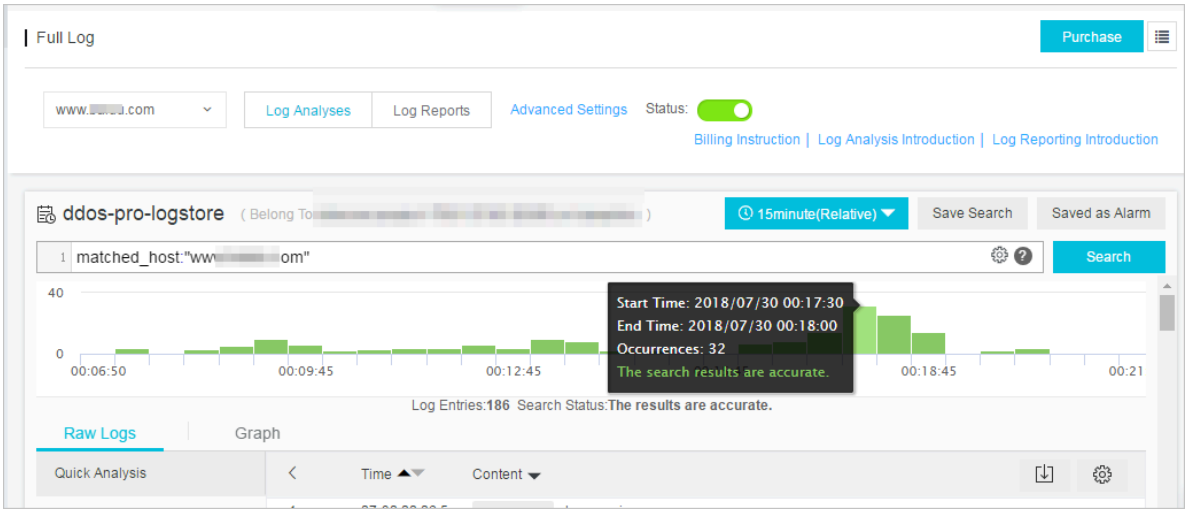
- View the log time distribution

  Under the search box, the time distribution of the log matching the query time and the query statement is displayed. Time distribution is displayed in the form of a histogram with the horizontal and vertical axis. The total number of queried logs is displayed.

  Note:

You can slide the histogram to select a smaller range of time zones, and the `time picker` automatically updates the selected time range and refresh the results.

Figure 4-8: Log time distribution



- View the raw logs

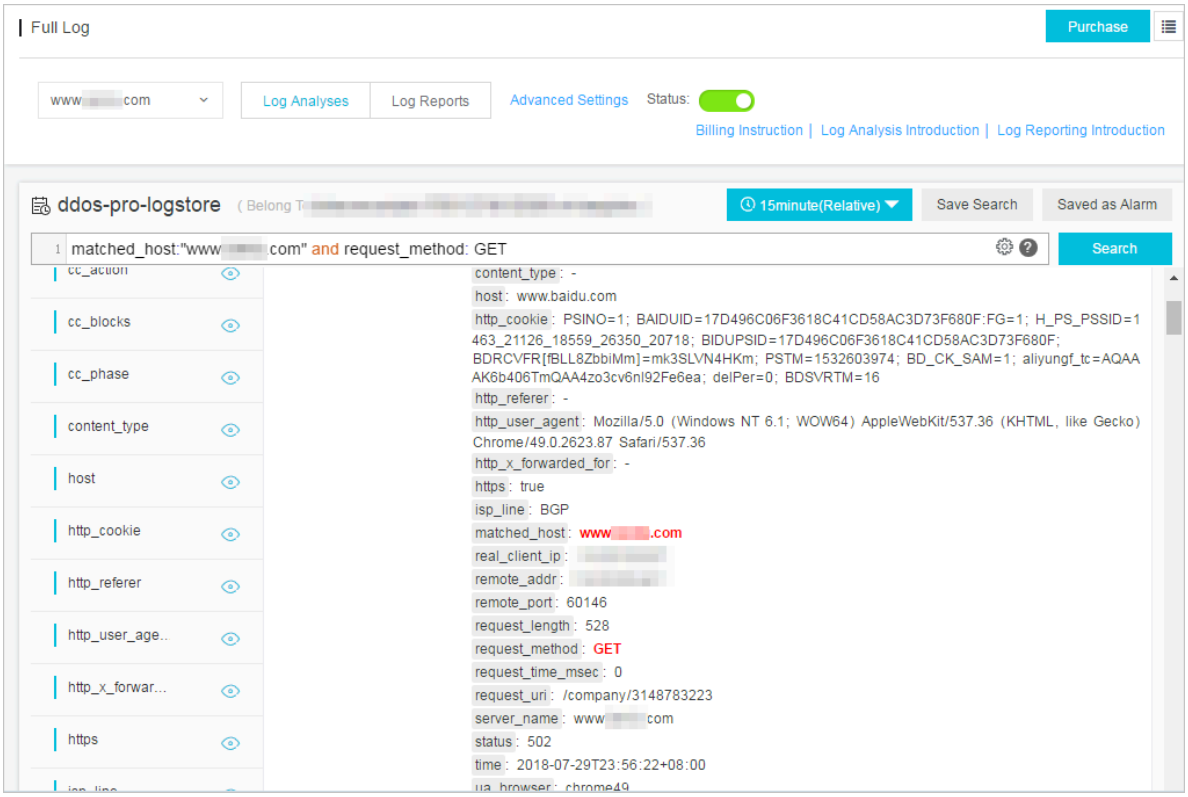
  In the Raw log, the details of each log are displayed in pagination, including time and content of these fields. You can sort the columns, download the current query results, or click the gear to select specific fields to be displayed.

  Click on the value or part of the corresponding field in the page to automatically enter the appropriate search criteria in the search box. For example, click

the value `GET` in `request_me thod : GET`, the following statement is automatically added to the search box:

```
Raw search statement and request_me thod : GET
```

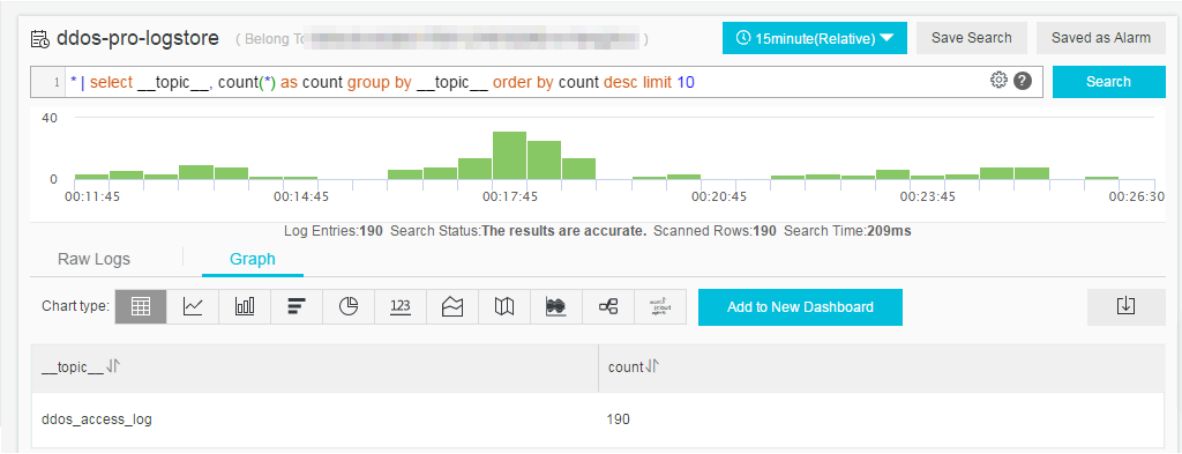
Figure 4-9: Raw logs



- View analysis charts

  Log Service supports graphical presentation of the analysis results, you can select different chart types on the Statistics Chart page. For more information, see *Analysis charts*.

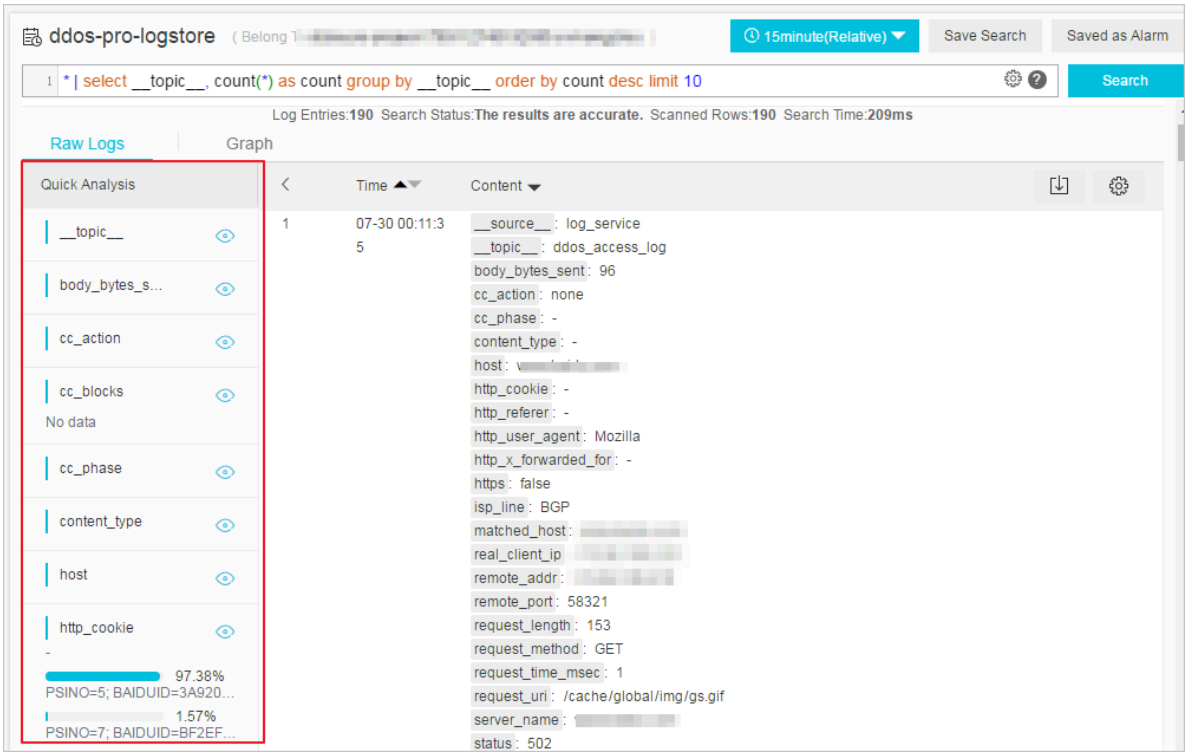
  Figure 4-10: Statistic chart



- Quick analysis

  Quick analysis feature provides one-click interactive query that helps you quickly analyze the distribution of a field over a period of time and reduce the time cost of indexing critical data. For more information, see *Quick analysis*.

  Figure 4-11: Quick analysis



Custom query analysis

Log query statement consists of two parts: query syntax (Search) and analysis syntax (Analytics), which are divided by `|`:

```
$ Search  | $ Analytics
```

Type	Description
Query (Search)	The query conditions can be generated by keywords, fuzzy, numerical values, interval range and combination conditions. If left empty or `*`, all data is displayed.
Analysis (Analytics)	Calculate and count the query results or the full amount of data.

Note:

Both Search and Analytics are optional. If Search is empty, all the data in the specified period is not filtered and the results are counted directly. If Analytics is empty, the query results are returned and no statistics are collected.

Query syntax

Log Service query syntax supports Full-text query and Field query. Query box supports line break display, syntax highlighting, and other functions.

- Full-text query

  You do not need to specify a field to enter the keyword query directly. You can wrap a keyword in double quotation marks (""), separated by a space or by `and` between multiple keywords.

  Example

  - Multiple keywords query

    Search for logs containing `www . aliyun . com` and `error` . For example:

    ```
    www . aliyun . com   error
    ```

    or

    ```
    www . aliyun . com   and   error
    ```

  - Conditional query

    Search for logs containing `www . aliyun . com` and including `error` or `404` . For example:

    ```
    www . aliyun . com   and  ( error   or   404 )
    ```

  - Prefix query

    Search for all keywords that contain `www . aliyun . com` and start with `failed_` . For example:

    ```
    www . aliyun . com   and   failed_ *
    ```

    Note:
    Query only supports suffix plus `*`, does not support prefix `*`, such as `* _error` .

- **Field query**

  Log Service supports more accurate queries based on fields.

  A comparison of numeric type fields can be implemented in the format `field : value` or `field >= value`, using `and`, `or`. It can also be combined with full-text search, also by using the combination of `and` and `or`.

  DDoS website access log and attack log can also base on field query. For the meaning, type, format and other information of each field, see *DDoS log field*.

  Example

  - Multiple fields query

    Search for logs containing `www . aliyun . com` attacked by CC:

    ```
    matched_ho  st :  www . aliyun . com  and  cc_blocks :  1
    ```

    Search the access logs containing the error 404 of a client `1 . 2 . 3 . 4` on the website `www . aliyun . com`:

    ```
    real_clien  t_ip :  1 . 2 . 3 . 4  and  matched_ho  st :  www .
    aliyun . com  and  status :  404
    ```

     **Note:**

Fields used in the examples `matched_ho st` , `cc_blocks` , `real_clien t_ip` , and `status` are fields of DDoS access and attack logs. For more information about fields, see *DDoS log fields*.

- Numeric field query

  Search for all slow request logs with a response time of more than 5 seconds:

  ```
  request_ti me_msec > 5000
  ```

  Interval queries are also supported, querying logs with a response time greater than 5 seconds and less than or equal to 10 seconds:

  ```
  request_ti me_msec in ( 5000 10000 ]
  ```

  The query can also be performed by the following statement:

  ```
  request_ti me_msec > 5000 and request_ti me_msec <=
  10000
  ```

- Check whether Japanese characters are used.

  Query for the presence of specific fields:

  ■ Query logs in the `ua_browser` field: `ua_browser : *`.

  ■ Query logs that do not belong to the `ua_browser` field: `not ua_browser : *`

For more information about query syntax, see *Index and query*.

Analysis syntax

You can use the SQL/92 syntax for log data analysis and statistics. For more information about the syntax and functions supported by Log Service, see *Syntax description*.



Note:

- The `from table name` statement in the SQL standard syntax can be omitted from the analysis statement, that is, `from log` .
- Log data returns the first 100 entries by default, and you can modify the return range by *LIMIT syntax*.

Time-based log query analysis

Each DDoS log has a `time` field, in the format `year - month - day T hour : minute : second + time zone`. For example, `2018 - 05 - 31T20 : 11 : 58 + 08 : 00`, where the time zone is `UTC + 8`, that is Beijing time. At the same time, each log has a built-in field: `__time__`, which also indicates the time of this log, so that time-based calculations can be performed in statistics. The format is *Unix timestamp*. The essence is a cumulative number of seconds since the 1970- 1 0:0:0 UTC time. Therefore, in actual use, after calculation, time must be formatted before it can be displayed.

- Select and show time

  Over a specific period of time, select the latest 10 logs of the website `www . aliyun . com` attacked by CC, show the time, source IP and access client, using the `time` field directly:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| select time , real_clien t_ip , http_user_ agent
    order by time desc
    limit 10
```

- Calculation time

  To query the number of days after the CC attack, use `__time__` to calculate:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| select time ,
          round (( to_unixtim e ( now ()) - __time__ )/ 86400 ,
1 ) as " days_passe d ", real_clien t_ip , http_user_ agent
      order by time desc
      limit 10
```



Note:

Use `round (( to_unixtim e ( now ()) - __time__ )/ 86400 , 1 )`, first part `to_unixtim e`, the time obtained by `now ()`, is converted to a Unix timestamp, and subtracted from the built-in time field `__time__` to get the number of seconds that have passed. Finally, divide by `86400`, which is the total number of seconds in a day, and then round it to the decimal with the function `round ( data , 1 )`. One-digit value indicates that each attack log has passed a few days.

- Group statistics based on specific time

  If you want to know how a website is being attacked by CC every day for a specific time frame, use the following SQL:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| select date_trunc (' day ', __time__ ) as dt ,
        count ( 1 ) as PV
    group by dt
    order by dt
```

  Note:
  This example uses the built-in time field `__time__` to pass the function `date_trunc (' day ', ..)` to the time alignment. Each log is grouped into the partition of the day it belongs to for the total number of statistics (count(1)) and sorted by partition time block. The first argument of the function `date_trunc` provides alignment for other units, including `second` , `miniute` , `hour` , `week` , `month` , `year` . For more information about function, see *Date and time functions*.

- Time-based group statistics

  For more flexible grouping time rules, for example, to know the trend of a website being attacked by CC every five minutes the math calculations are required. Run the following SQL:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| select from_unixt ime ( __time__ - __time__ % 300 ) as
dt ,
        count ( 1 ) as PV
    group by dt
    order by dt
    limit 1000
```

  Note:
  Use the built-in time field to calculate `__time__ - __time__ % 300` and format it using the function `from_unixt ime` . Each log is grouped into a 5 minute (300 seconds) partition for the total number of statistics (count(1)), and sorted by partition time block to obtain the first 1000 logs, which is equivalent to the first 83 hours of data in the selection time.

More time-resolved functions, such as converting a time format, require using `date_parse` and `date_forma t` . For more information, see *Date and time functions*.

Client IP-based query analysis

DDoS log has a field `real_clien t_ip`. However, if the user cannot obtain the real IP by the proxy and the IP address in the header is incorrect, you can use the `remote_add r` field to directly connected to the client IP.

- Country attack distribution

  Distribution of source countries of CC attacks on a website:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_coun try ( if ( real_clien t_ip ='-',
remote_add r , real_clien t_ip )) as country ,
        count ( 1 ) as " number of attacks "
        group by country
```

  Note:
  Use the function `if ( condition , option1 , option2 )` to select the field `real_clien t_ip` or `real_clien t_ip` (when `real_clien t_ip` is -). Pass the obtained IP to the function `ip_to_coun try` to get the country information corresponding to this IP.

- Access distribution

  To get more detailed province-based distribution, use the `ip_to_prov ince` function, for example:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_prov ince ( if ( real_clien t_ip ='-',
remote_add r , real_clien t_ip )) as province ,
        count ( 1 ) as " number of attacks "
        group by province
```

  Note:
  Another IP function `ip_to_prov ince` to get a province of IP. If IP address is outside of China, system still tries to convert to the province (state), .

- Attackers heat distribution

  To get an attackers heat map, use the `ip_to_geo` function, for example:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_geo ( if ( real_clien t_ip ='-', remote_add r
, real_clien t_ip )) as geo ,
        count ( 1 ) as " number of attacks "
        group by geo
```

```
            limit   10000
```

> **Note:**
> Use another IP function `ip_to_geo` to get the latitude and longitude of an IP and get the first 10,000.

More IP-based parsing functions, such as obtaining the IP operator `ip_to_prov ider` , determining whether the IP is Internet or Intranet `ip_to_doma in` , see *IP functions*.

4.4.4 Log Report

Log Reports page is embedded in the dashboard of the Log Service. This page displays the default dashboard. You can view dashboard data under various filter conditions by modifying the time range and adding filters.

View reports

1. Log on to the Anti-DDoS Pro console and select Log > Full Log in the left-side navigation pane. Enter the Full Log page.
2. Select the website for which you want to enable DDoS log collection function and make sure the Status is on.
3. Click Log Reports.

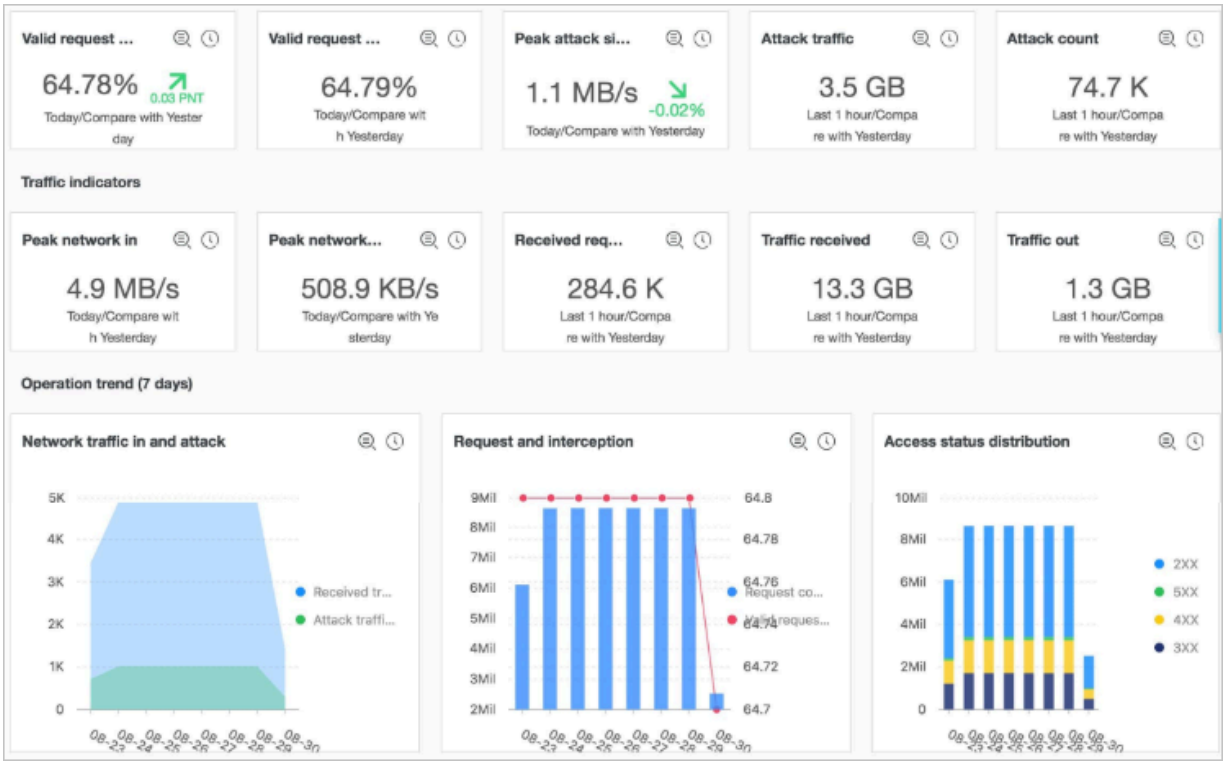
   Dashboard page of Log Service is embedded in the current page, and the filter condition is automatically added. For example, use `matched_ho st : www . aliyun . com` to view log reports based on selected website.

Figure 4-12: View reports

After the DDoS log collection function is enabled for the website, Log Service automatically creates two default instruments for reporting: operation center and access center. For more information about the default dashboard, see *Default dashboard*.

Dashboard	Dashboard name	Description
ddos-pro-logstore_ddos_operation_center	DDoS operation center	Displays the current overall operational status of DDoS protected websites, including valid request status, traffic, trends, attack distributions, and traffic volumes and peaks attacked by CC.
ddos-pro-logstore_ddos_access_center	DDoS access center	Displays the current overall operational status of DDoS protected websites, including PV/UV trends and bandwidth peaks, visitors, traffic, client type, request, and visited websites distribution.

Figure 4-13: Default dashboard



Besides viewing the report, the following operations can be performed:

- Select *time range*
- Add or edit *filter condition*

- View *charts*

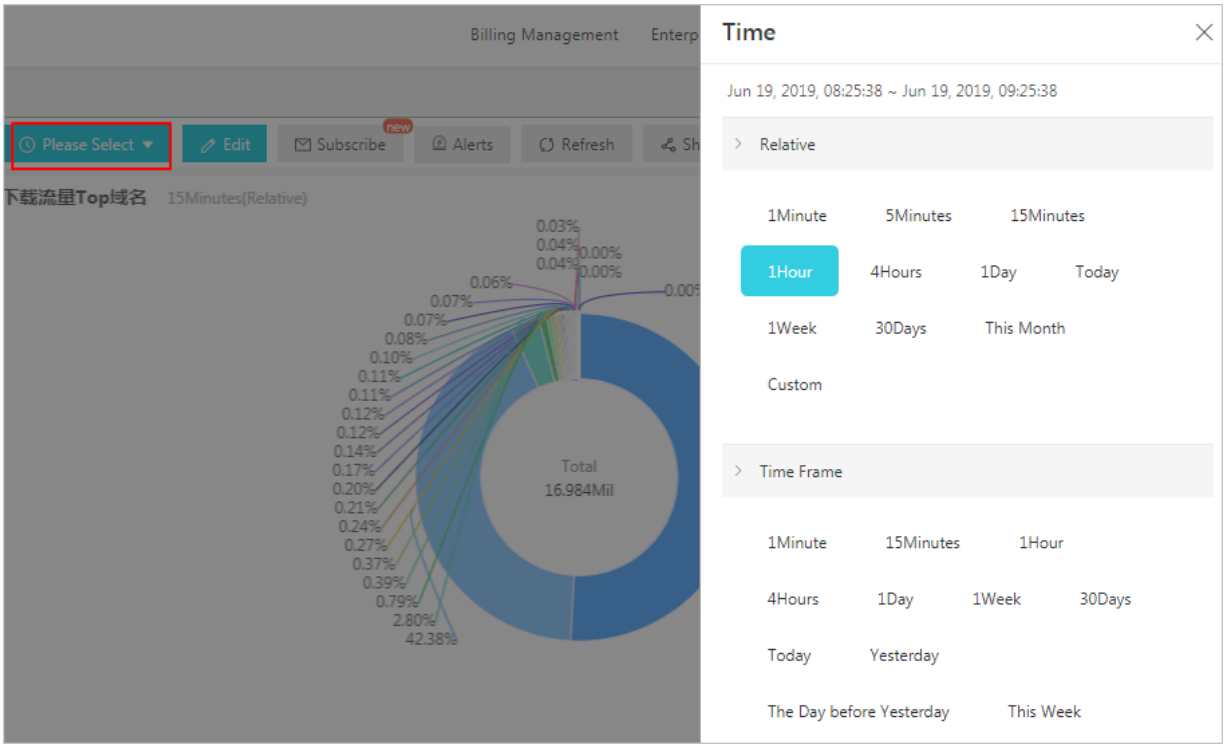
Time picker

All charts on the dashboard page are based on statistical results for different time periods. For example, the default time range for visits is one day and the access trend is 30 days. To set all charts on the current page to be displayed in the same time range, you can configure the time picker.

1. Click Select.
2. Configure the settings in the dialog box. You can select relative time, entire point time, or set a custom time.

> **Note:**
>
> - When the time range is modified, the time of all charts is changed to this time range.
> - Time picker only provides a temporary view of the chart on the current page, and the system does not save the setting. The next time you view the report, the system will display the default time range.

Figure 4-14: Set the time range



Filter conditions

Select the website and click Log Reports to enter the dashboard page. System automatically adds filter condition, such as `matched_ho st : www . aliyun . com` to view log reports based on selected website.

You can modify the data display range of the report by setting filter condition.

- View overall reports for all websites

  Clear the filter condition to display the overall reports library `ddos - pro - logstore` .

- Add more filter conditions

  You can filter the report data by setting key and value. `AND` relationship between multiple filters is supported.

  For example, view the overall situation of access requests by telecommunications lines.

  Figure 4-15: Add filter conditions



Note:
The `isp_line` is the field of the DDoS log, indicating the operator network connecting to the port. For more information about fields, see *DDoS log fields*.

Chart type

The report display area shows multiple reports according to a predefined layout, including the following types. For more information about chart types, see *Graph description*.

Chart type	Description
Number	Displays important indicators, such as effective request rate, and attack peaks.
Line/area map	Displays trend graphs for certain important indicators within a specific time period, such as inbound bandwidth trends and attack interception rates.

Chart type	Description
Map	Displays the geographical distribution of visitors and attackers, such as CC attack country, access hotspot.
Pie chart	Displays the distribution of the information, such as the top 10 of the websites being attacked, client type distribution.
Table	Displays information such as the list of attackers, typically divided into multiple columns.
Maps	Displays the geographical distribution of the data.

Default dashboards

- Operation center

  Operations center displays the current overall operational status of DDoS protected websites, including valid request status, traffic, trends, attacker distributions, and traffic volumes and peaks attacked by CC.

Chart	Type	Default time range	Description	Example
Valid request package rate	Single value	1 hour (relative)	A valid request, that is, the number of non-CC attacks or 400 error requests in the total number of all requests.	95%
Valid request flow rate	Single value	1 hour (relative)	Valid request percentage of the total flow of all requests.	95%
Received traffic	Single value	1 hour (relative)	The sum of valid request inflows. The unit is MB.	300 MB

Chart	Type	Default time range	Description	Example
Attack traffic	Single value	1 hour (relative)	The sum of inbound traffic of CC attacks. The unit is MB.	30 MB
Outbound traffic	Single value	1 hour (relative)	The sum of valid request outbound traffic. The unit is MB.	300 MB
Network in bandwidth peak.	Single value	1 hour (relative)	The highest peak of incoming traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Network out bandwidth peak.	Single value	1 hour (relative)	The highest peak of outbound traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Received data packets	Single value	1 hour (relative)	The number of incoming requests for valid requests (non-CC attacks), measured in units.	30, 000
Attack data packets	Single value	1 hour (relative)	The sum of the number of requests for the CC attack, measured in units.	100

Chart	Type	Default time range	Description	Example
Attack peak	Single value	1 hour (relative)	The highest peak of CC attack. The unit is number per minute.	100 per minute
Inbound bandwidth and attack trends	Two-line diagram	1 hour (entire point)	Trend chart of valid requests per minute and traffic bandwidth for attack requests. The unit is KB/s.	-
Request and interception trends	Two-line diagram	1 hour (entire point)	Trend chart of the total number of requests and intercepted CC attack requests per minute. The unit is number per minute.	-
Valid request rate trend	Two-line diagram	1 hour (entire point)	Trend chart of the number of valid requests per minute (non-CC attacks or 400 error requests) in the total number of all requests.	-

Chart	Type	Default time range	Description	Example
Access status distribution trend	Flow chart	1 hour (entire point)	Trend chart of various request processing statuses (400, 304, 20) per minute. The unit is number per minute.	-
CC attacks distribution	World map	1 hour (relative)	The sum of the number of CC attacks in the source country.	-
CC attack distribution	Map of China	1 hour (relative)	The sum of the number of CC attacks in the source province (China).	-
List of attacks	Table	1 hour (relative)	The attacker information of the first 100 attacks, including IP, city, network, number of attacks, and total traffic.	-
Attack access line distribution	Pie chart	1 hour (relative)	CC attack source access DDoS protection line distribution, such as telecommunications, Unicom, and BGP.	-

Chart	Type	Default time range	Description	Example
Top 10 attacked websites	Donut chart	1 hour (relative)	Top 10 attacked websites	-

- Access center

  Access center displays the current overall operational status of DDoS protected websites, including PV/UV trends and bandwidth peaks, visitors, traffic, client type, request, and visited websites distribution.

Chart	Type	Default time range	Description	Example
Page view	Single value	1 hour (relative)	The total number of requests.	100,000
Unique visitors	Single value	1 hour (relative)	Total number of independent access clients.	100,000
Inbound traffic	Single value	1 hour (relative)	The sum of inbound traffic of the website. The unit is MB.	300 MB
Network in bandwidth peak.	Single value	1 hour (relative)	The highest peak of inbound traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Network out bandwidth peak.	Single value	1 hour (relative)	The highest peak of inbound traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s

Chart	Type	Default time range	Description	Example
Traffic bandwidth trend	Two-line diagram	1 hour (entire point)	Trend chart of website inbound and outbound traffic per minute. The unit is KB/s.	-
Request and interception trends	Two-line diagram	1 hour (entire point)	Trend chart of the total number of requests and intercepted CC attack requests per minute . The unit is number per minute.	-
PV/UV access trends	Two-line diagram	1 hour (entire point)	Trend chart of PV and UV per minute. Measured in units.	-
Visitor distribution	World map	1 hour (relative)	The distributi on of visitors PV (page view) in the source country.	-
Visitor heat map	Amap	1 hour (relative)	Visitor geographic access heat map.	-
Inbound traffic distribution	World map	1 hour (relative)	Sum of inbound traffic distribution in the source country. The Unit is MB.	-

Chart	Type	Default time range	Description	Example
Inbound traffic distribution	Map of China	1 hour (relative)	Sum of inbound traffic distribution in the source province. The Unit is MB.	-
Access line distribution	Donut chart	1 hour (relative)	Source-based access DDoS protection line distribution, such as telecommunications, Unicom, and BGP.	-
Inbound traffic network provider distribution	Donut chart	1 hour (relative)	The distribution of inbound traffic that visitors access by network operators. For example, telecommunications, Unicom, mobile connections, education network. The Unit is MB.	-

Chart	Type	Default time range	Description	Example
Most visited clients	Table	1 hour (relative)	The top 100 most visited clients, including IP, city, network, request method distribution, incoming traffic, number of incorrect accesses, number of intercepted CC attacks.	-
Access domain name	Donut chart	1 hour (relative)	The top 20 most visited domain names.	-
Referer	Table	1 hour (relative)	The top 100 most redirected referer URLs, hosts, and frequency.	-
Client type distribution	Donut chart	1 hour (relative)	The top 20 most visited user agents, such as iPhone, iPad, Windows IE, Chrome.	-
Request content type distribution	Donut chart	1 hour (relative)	The top 20 most requested content types, such as HTML, Form, JSON, streaming data.	-

4.4.5 Billing method

DDoS log collection function is charged according to the charge items of the Log Service. If no log data is generated, no billing is made. Log Service is billed by resource usage and provides the FreeTier quota for DDoS Logstore.

DDoS log collection function provides functions such as log collection, storage, real-time query and analysis, and dashboards. The real-time query and analysis of log data relies on Log Service. Therefore, this feature is charged according to Log Service billing method. Log Service is billed by the resource usage and provides the FreeTier quota for DDoS Logstore. The specific fee depends on the amount of your log data. If you have Log Service enabled, but you have not turned on logging function for any website, no charge appears.

Deduction and outstanding payment

Log Service is billed by the resource usage, and the billing cycle is one day. For more information about deduction and outstanding payment, see *Deduction and outstanding payment*.

Billing item

Billing item	Description
Read and write traffic	· The read and write traffic is calculated by the traffic for transmitting compressed logs. DDoS logs are generally compressed by 5 to 10 times. · Read and write traffic also includes a loss of consumption interface that generates read traffic, generally, by using API/SDK and consumer group SDK. According to the compressed transmission traffic calculation, logs can be compressed in the API/SDK mode. Note: In the Log Service console, Preview function under Log Consumption also can generate micro-flow traffic consumption. · The data generated by the index-based query and analysis is free of read and write traffic charges. For example, the log query analysis, dashboards, and alarms in the console are not charged.
Storage space	The storage space is the sum of data size after compression and the indexed data size.

Billing item	Description
Indexing traffic	· The indexing traffic is calculated by actual index fields. Storage fee is collected in full during writing. DDoS logs enable full indexing by default. · The traffic of fields having both FullText and KeyValue indexes is calculated only once. · Indexes occupy the storage space and thus the storage space fee is collected.
Active shard rent	Only shards currently in readwrite status are counted. Rent of merged/split shards is not collected. Note: By default, Log Service creates two shards, and enables the *Auto Split Shard* feature. Typically, each shard can proceed 430 GB of write data volume per day.
Read/write count	The write count of logs written into Log Service is a subject to the log generation speed. The background realization mechanism minimizes the read/write count.
Internet read traffic	The data traffic generated when Internet programs read log data collected by Log Service.

FreeTier quota

Log Service is not charged in the following cases:

- Log Service is activated, and DDoS logging function has not been enabled for any website.
- The amount of website logs that enable DDoS logging is within the free quota.
- Index-based query analysis, reports, and alarms are not charged.

Log Service provides the free quota for your DDoS Logstore. If the data volume is less than the free-quota limit, no charges appears.

Billing item	FreeTier quota
Read and write traffic	30 GB/day
Storage space	3 days
Indexing traffic	100 GB/day
Active shard rent	4 days/month
Read/write count	1 million times/day

Billing item	FreeTier quota
Internet read traffic	0
Read traffic consumption	0
Read count consumption	0

Note:
Log data storage time is set to 3 days by default, and when you modify for more than 3 days, extra charges can appear.

Billing method

When the log volume of the website that enables the log analysis function exceeds the free quota, Log Service charges the excess of the quota amount.

Billing item	Extra payment
Read/write traffic (USD/GB)	0.045
Storage space (USD/GB/day)	0.002875
Indexing traffic (USD/GB)	0.0875
Active shard rent (USD/day)	0.01
Read/write count (USD/million times)	0.03
Internet read traffic (USD/GB)	0.2

Billing example

- FreeTier quota: The average log is about 1600 bytes, about 60 million logs are generated per day, and the storage period is 3 days. The total log volume is about 96 GB per day, not exceeding the quota.
- Index: The log volume is 150 GB per day, and the 50 GB is charged (150 GB - 100 GB), which is 0.0875 x 50 = 17.5 USD per day.
- Write transmitting: The log volume is 300 GB per day, logs are compressed in six times. The actual compression size is about 50GB, and the 20GB is charged (50GB - 30GB), which is 0.045 x 20 = 0.9 USD per day.
- Storage space size:
  - 10 GB of data per day, 2 GB after compression, and 10 GB of indexing traffic. The storage period is 30 days, and the maximum storage capacity after 30 days is 30

× (10+2) = 360 GB, with a 3-day free quota, it is 27 × (10+2) = 324 GB, and the maximum charge for one day storage is 0.002875 × 324 = 0.9315 USD.

- 1 GB of data per day, 200 MB after compression, and 1 GB indexing traffic. The cumulative maximum storage capacity after 30 days is 30 × (1000 + 200) ≈ 36 GB, with a 3-day free quota, it is 27 × (1000 + 200) ≈ 32.4 GB, and the maximum charge for one day storage is 0.002875 × 32.4 = 0.09315 USD.

- Active shard rent: Currently, there are 10 shards, 7 read/write shards, and 3 read-only shards. DDoS Logstores are only charged per day. The rental fee for 3 (7 - 4) shards is 0.03 USD per day.
- Read/write count: The number of website logs is 10 billions per day, and the write count is about 500,000 (on average, 2,000 per time), free of charge.
- Internet traffic: 2 GB of Log Service data was delivered to non-Alibaba Cloud products, resulting in an external network read traffic of 0.4 USD.

Billing FAQ

- How can I modify the storage time of website logs?

  1. Log on to the Log Service console, click the Project name to enter the Logstore list. The default Project for DDoS log is `ddos-pro-project-`*`Alibaba Cloud Account ID`*.
  2. Click Modify in the Action column.
  3. On the Data Storage Time page, click Modify.

- How can I view the current log volume and estimate the cost?
  - To view the cost measurement data on day basis go to Alibaba Cloud *Expense Management Center*.
  - 1. Log on to the DDoS IP protection console and click Full Log on the left.
    2. Select the website which log volume you want to view, and click Log Analysis on the right.
    3. Enter the following query statement in the query box, the time range is `Yesterday ( entire point time ):`

```
__topic__ : ddos_acces s_log | select count ( 1 ) as
PV
```

    4. Click Query and select Statistics Chart with the chart type Table .

    You can get data volume of the previous day, and estimate the cost according to your current log storage time.
  - How can I configure Log Service to trigger an alarm when a large number of logs is generated?

    When a large number of DDoS logs is collected, the free quota of Log Service may be exceeded, and the certain charge can appear. If you want to receive an alarm notification when there is such a risk, you can configure Log Service to trigger an alarm when a large number of logs is generated.

    1. Log on to the DDoS IP protection console and click Full Log on the left.
    2. Select the website which log volume you want to view, and click Log Analysis on the right.
    3. Enter the following query statement in the query box, and click Query:

```
*| select count ( 1 ) as PV
```

    4. Click Save as Quick Query in the upper-right corner of the query page to enter the information about the query, such as `ddos - metering - pv` . Then click OK.
    5. Click Save as Alarm and create an alarm configuration, see the following figure. Check the log volume of the past 1 hour every 5 minutes, and trigger an alarm if more than 5.6 million logs are generated.

     **Note:**

To ensure that the daily log volume is less than 100 GB free quota, the average hourly import volume is estimated to be: 100 GB ÷ 1600 bytes ÷ 24 hours ≈ 2.8 million. The example is two times of the hourly log volume, which is 5.6 millions, and can be adjusted according to the actual situation and needs.

4.5 Logs of BGP-line Anti-DDoS Pro

4.5.1 Overview

Alibaba Cloud Anti-DDoS Pro provides BGP bandwidth resources that are exclusively available in Chinese Mainland to mitigate volumetric DDoS attacks. The service can scrub Terabits of attack traffic per second based on eight ISPs. Compared with earlier versions, Anti-DDoS Pro supports more reliable networks with less latency and provides quicker disaster recovery.

Background

Security is always a challenge facing the Internet. Network threats represented by DDoS attacks have a serious impact on network security.

DDoS attacks are becoming more large-scale, mobile, and global. According to recent survey reports, the frequency of DDoS attacks is on the rise. Attackers are difficult to detect and can manipulate a large number of cloud service providers with poor security measures, IDCs, and even cameras to launch attacks. The attackers have formed a mature black industry chain and become increasingly organized. At the same time, the attack methods develop towards polarization. The proportion of slow and hybrid attacks, especially HTTP flood attacks, is increasing, which makes detection and defense more difficult. On the one hand, attacks peaking at 1 Tbit/s or higher become common, and the number of 100 GB attacks is multiplied. On the other hand, application layer attacks are also doubled.

According to *Kaspersky Lab DDoS Q1 2018 Intelligence Report*, China remains the main source and target of DDoS attacks. The main industries under attack include the Internet, games, software companies, and financial companies. More than 80% of DDoS attacks mix with HTTP attacks, and HTTP flood attacks are especially difficult to detect. Therefore, it is particularly important to analyze and study access and attack activities through logs and set protection policies accordingly.

Log Service can collect website access logs and HTTP flood attack logs of *Alibaba Cloud Anti-DDoS Pro* in real time. Log Service also supports real-time retrieval and analysis of the collected log data and displays the query results in the form of dashboards.

Benefits

- Simple configuration: Real-time anti-DDoS logs can be collected with simple configuration. Log collection is automatically enabled for new websites after they are added.
- Real-time analysis: Relying on Log Service, this function provides real-time log analysis and an out-of-the-box report center. This helps you understand the HTTP attack status and customer access details.
- Real-time alerts: Supports real-time monitoring and alerts based on customized metrics to ensure timely response to critical business exceptions.
- Collaboration: This function can be integrated with real-time computing, cloud storage, visualization, and other data solutions to discover more data value.
- Free quota: Provides a free data import quota of 3 TB and also allows you to use the log storage, query, and real-time analysis functions for 30 days for free.

Restrictions and guidelines

- Additional data cannot be written to the exclusive Logstore.

  The exclusive Logstore is used to store Anti-DDoS Pro website logs. Writing other data is not supported. Other functions such as query, statistics, alerts, and streaming consumption are not restricted.
- The data TTL and total storage capacity of the exclusive Logstore cannot be modified.

  Purchase the storage capacity of the exclusive Logstore based on your business requirements. Up to 1,000 TB of storage capacity is supported, and logs can be stored for up to 180 days.

Scenarios

- Troubleshoot website access exceptions

  After you configure Log Service to collect Anti-DDoS Pro logs, you can query and analyze the collected logs in real time. By using SQL statements to analyze the Anti-DDoS Pro access logs, you can quickly troubleshoot and analyze the website

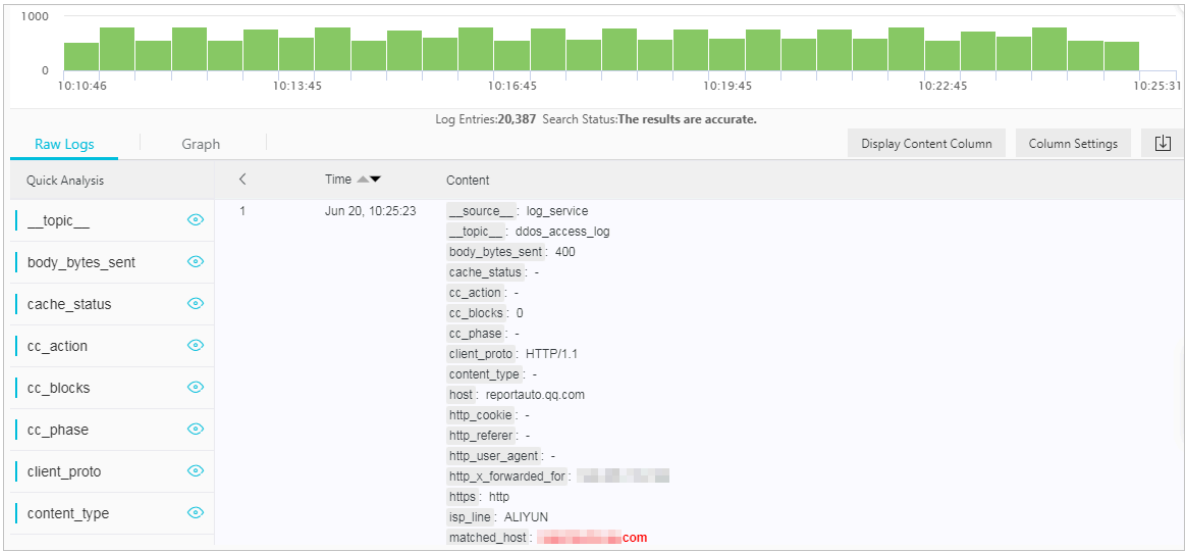
access exceptions. You can also query information such as read and write delays and exception distribution by ISP.

For example, use the following statement to query the access log entries of Anti-DDoS Pro:

```
__topic__ : DDoS_acces s_log
```

The query results are displayed, as shown in the following figure.

Figure 4-16: Access log entries of Anti-DDoS Pro



- Track HTTP flood attack sources

  Anti-DDoS Pro access logs record information about the sources and distribution of HTTP flood attacks. You can query and analyze access logs in real time to identify the attackers, and use this information to select the most effective protection policy.

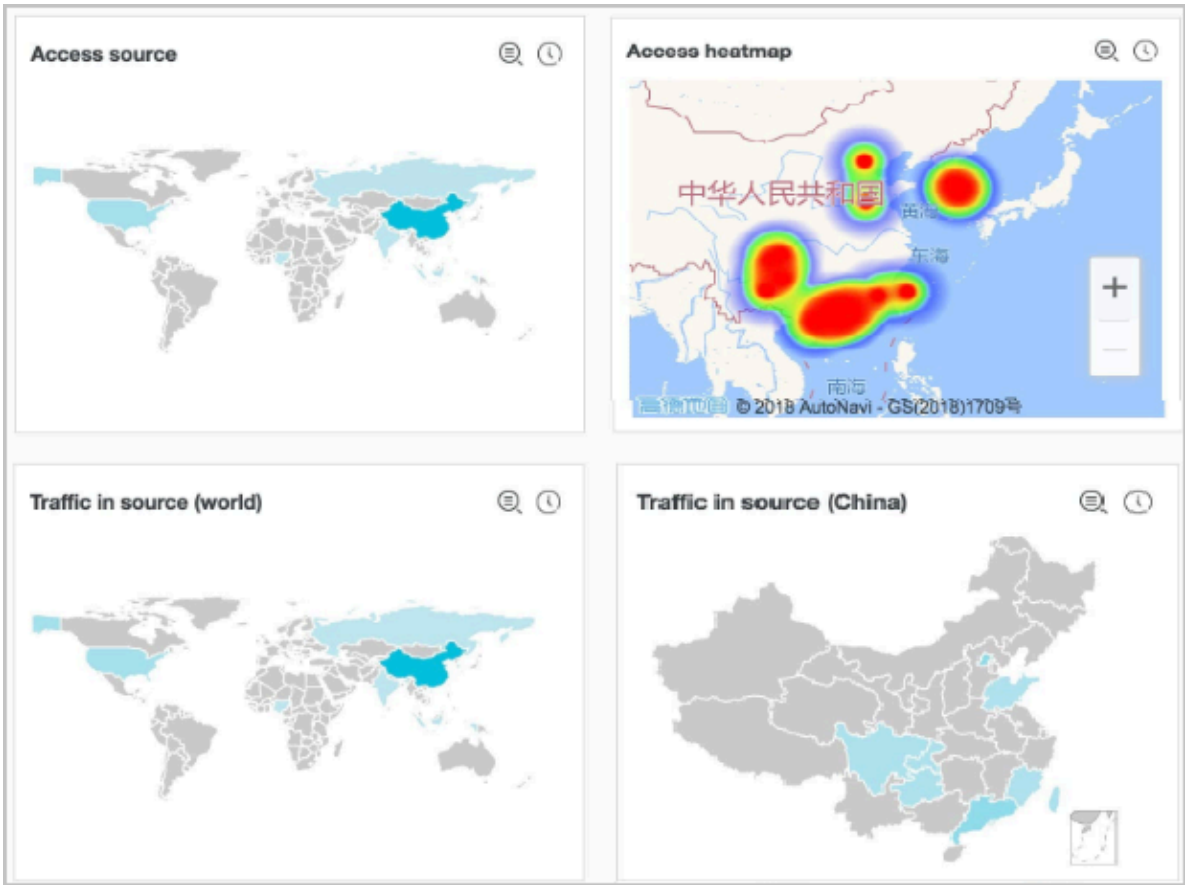
  For example, use the following statement to analyze the geographical distribution of HTTP flood attacks:

```
__topic__ : DDoS_acces s_log and cc_blocks > 0 | SELECT
  ip_to_coun try ( if ( real_clien t_ip ='-', remote_add r ,
```

```
real_clien  t_ip ))  as  country ,  count ( 1 )  as  " number "
group  by  country
```

The analysis results are displayed in a dashboard as follows:

Figure 4-17: HTTP flood attacks

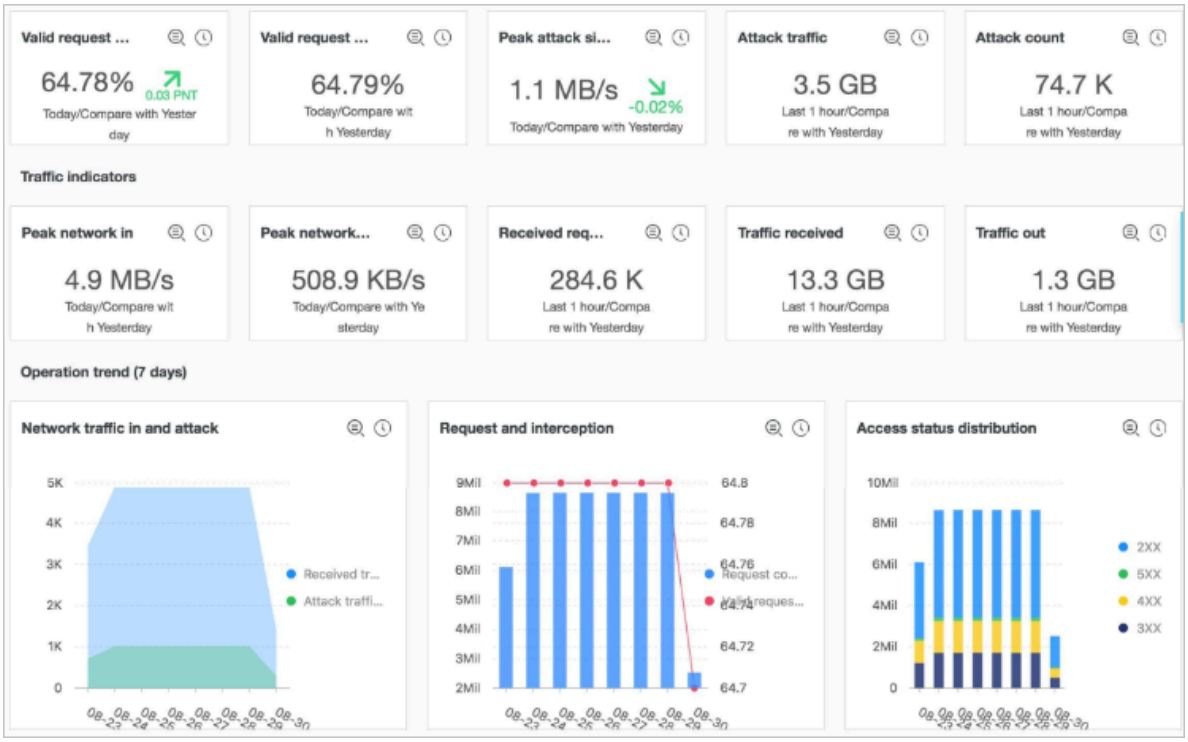


- For example, use the following statement to view PVs:

```
__topic__ : DDoS_acces s_log | select count ( 1 ) as PV
```

The analysis results are displayed in a dashboard, as shown in the following figure.

Figure 4-18: PV access



- Analyze website operations

Anti-DDoS Pro access logs record information about website traffic in real time. You can use SQL queries to analyze log data to better understand your visitors and analyze website operations. For example, you can identify the most visited Web pages, the browsers that initiated the requests, and the clients, source IP addresses, and ISPs of the requests.

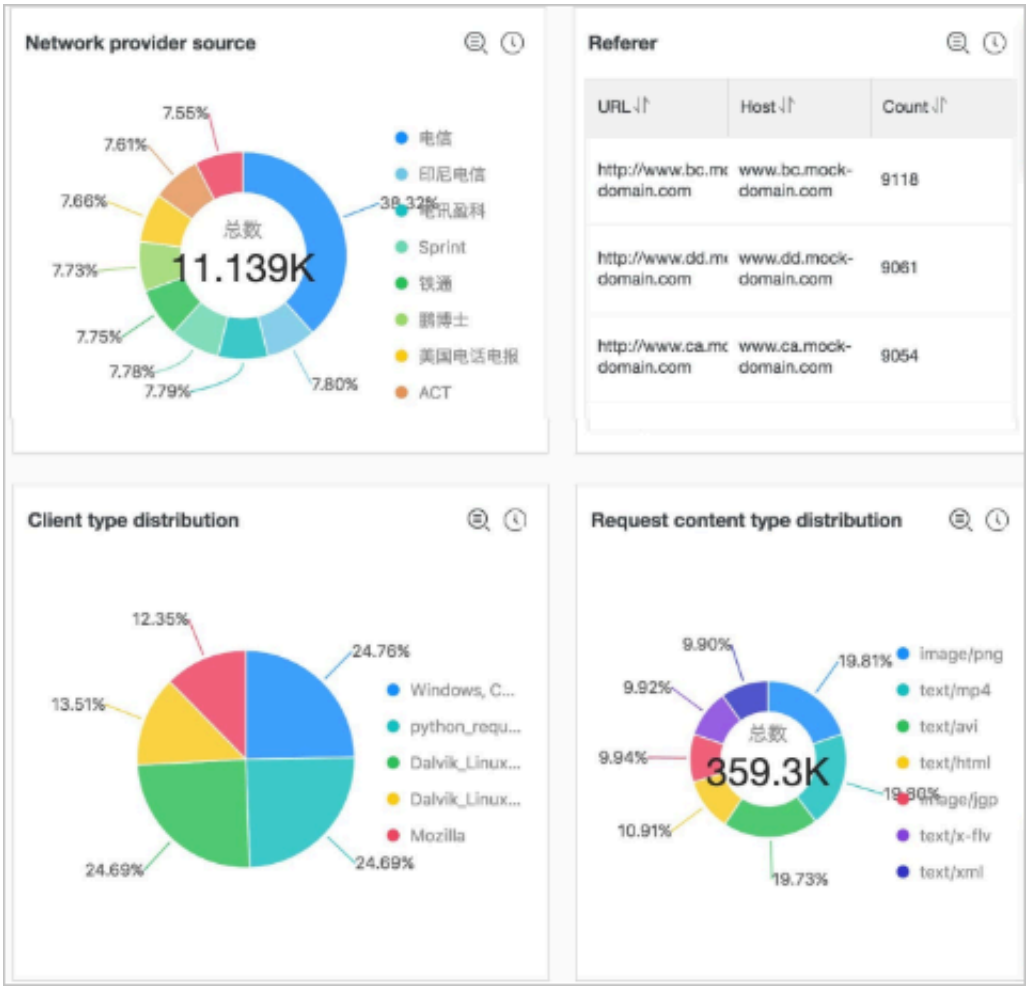
For example, use the following statement to view the visitor distribution by ISP:

```
__topic__ : ddos_acces s_log | select ip_to_prov ider ( if ( real_clien t_ip ='-', remote_add r , real_clien t_ip )) as provider , round ( sum ( request_le ngth )/ 1024 . 0 / 1024 . 0 , 3 ) as mb_in group by provider having ip_to_prov
```

```
ider ( if ( real_clien  t_ip ='-',  remote_add  r ,  real_clien
t_ip )) <> ''  order   by   mb_in   desc   limit   10
```

The analysis results are displayed in a dashboard, as shown in the following figure.

Figure 4-19: Visitor distribution



4.5.2 Enable or disable log collection

When you purchase the Anti-DDoS Pro service, the log collection function is automatically enabled. You can disable or re-enable the log collection function of the specified website.

Context

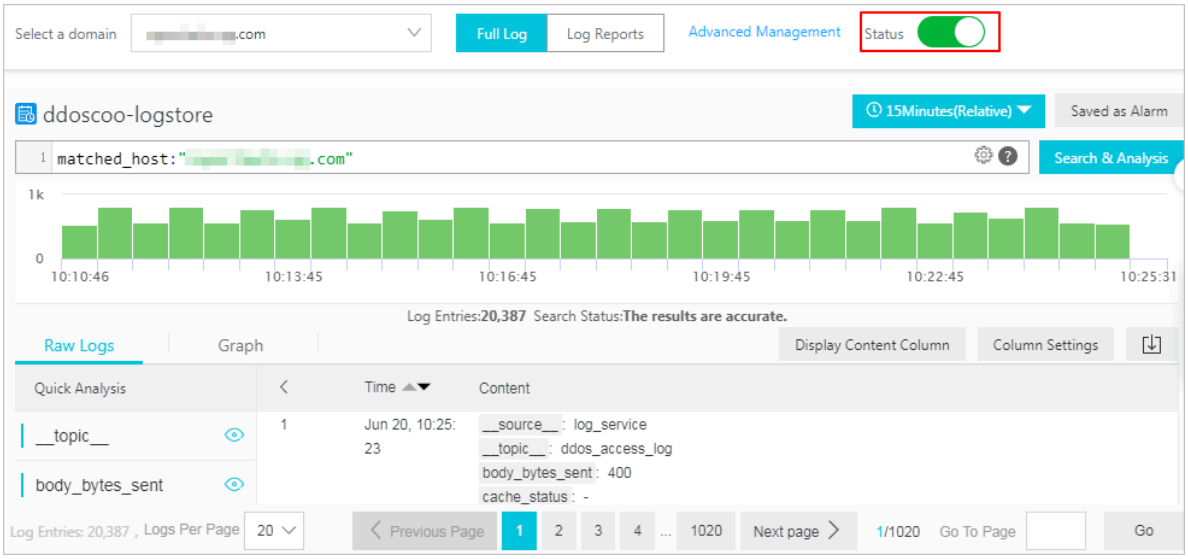
Log Service can collect website access logs and HTTP flood attack logs of Alibaba Cloud Anti-DDoS Pro in real time. Log Service also supports real-time retrieval and analysis of the collected log data and displays the query results in the form of dashboards. By analyzing the access and attack activities in real time through logs, Log Service helps the security department set protection policies.

Procedure

1. Log on to the Anti-DDoS Pro console, and choose Statistics > Full Log in the left-side navigation pane.
2. Select the website for which you want to enable log collection, turn on or off the Status switch.

   By default, log collection is enabled when you purchase Anti-DDoS Pro. This function is also enabled for newly added websites.

   Figure 4-20: Enable or disable the log collection function



Log Service automatically creates an exclusive project and an exclusive Logstore under your account. The Anti-DDoS Pro logs of all websites that have log collection enabled are imported to this exclusive Logstore. For more information about the default configurations of the exclusive project and Logstore, see the following table .

Table 4-2: Default configuration

Default configuration item	Description
Project	By default, a project is created. The name of the project is `ddoscoo-project-`*`Alibaba Cloud account ID`*`-cn-hangzhou`.

Default configuration item	Description
Logstore	By default, a Logstore is created. The name of the Logstore is `ddoscoo - logstore`. All log entries generated by the Anti-DDoS Pro log collection function are stored in this Logstore.
Region	By default, the project is created under the China (Hangzhou) region.
Shard	By default, two shards are created and the *automatic shard splitting* function is enabled.
Log TTL	A log entry can be stored for 30 days. After 30 days, the log entry is automatically deleted.
Log storage capacity	You can purchase the storage capacity of the exclusive Logstore based on your business requirements. The maximum storage capacity is 1,000 TB, and log entries can be stored for up to 180 days.
Dashboard	By default, the following dashboards are created: · Access center: displays website access metrics, client distribution, traffic, and performance data. · Operations and maintenance center: displays attack status and operational metrics such as PV, UV, and success rate. For more information about dashboards, see *Log reports of BGP-line Anti-DDoS Pro*.

You can query and analyze collected log entries in real time on the Log Service page. For more information about log fields, see *Log fields*. In addition, apart from the operations and maintenance center and access center that are created by Log Service, you can also customize a dashboard.

What's next

- Click Search & Analysis to *query and analyze* the collected log data.
- Click Log Reports to view built-in *Log reports of BGP-line Anti-DDoS Pro*.
- Click Advanced Management to go to the Log Service console. You can query log data, collect statistics, consume streaming data, and set alerts for the collected log data.

4.5.3 Manage log storage space

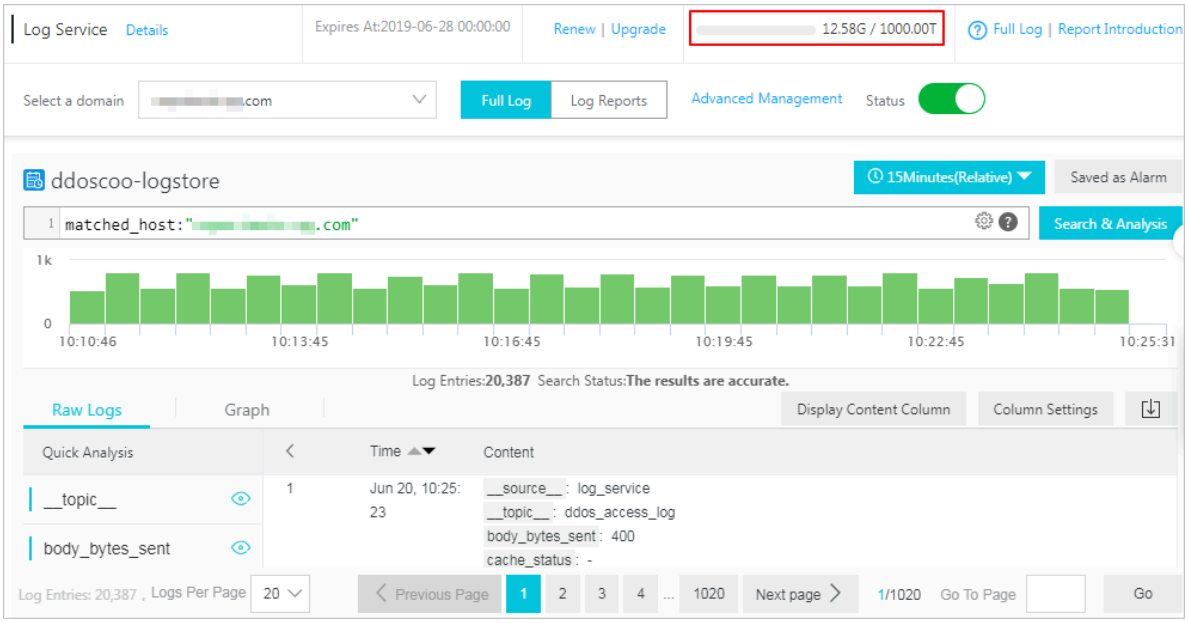
After you enable the Anti-DDoS Pro log collection function, log data is pushed to the specified Logstore in real time. You can view the usage of the log storage space in the Anti-DDoS Pro console.

View log storage usage

You can view the usage of the log storage for Anti-DDoS Pro Log Service at any time.

> **Note:**
> The log storage information in the console is not updated in real time. It takes up to two hours to synchronize the actual storage information to the console. Therefore, we recommend that you expand the log storage space before it is exhausted.

1. Log on to the Anti-DDoS Pro console.
2. In the left-side navigation pane, choose Statistics > Full Log.
3. In the upper-right corner of the page, view the log storage usage.



Clear log storage

You can clear all log entries stored in the log storage space. For example, you can clear the log entries generated during the testing phase and use the log storage space to store log entries that are generated during the production phase only.

Go to the Log Service page, click Clear in the upper-right corner to clear all the log entries in your log storage space.

Notice:
Log entries that are cleared cannot be restored. Clear log entries with caution.

Note:
You can clear the log storage for only a limited number of times.

4.5.4 Log fields

This topic describes the supported fields of Anti-DDoS Pro log entries.

You can go to the Log Service page to query and analyze collected logs in real time.

For more information about log fields, see the following figure.

Field	Description	Example
__topic__	The topic of the log entry. The value of this field is fixed to ddos_access_log.	N/A
body_bytes_sent	The size of the body in the access request, in bytes.	2
content_type	The content type.	application/x-www-form-urlencoded
host	The source website.	api.zhihu.com
http_cookie	The request cookie.	k1=v1;k2=v2
http_referer	The request referer. If no referer exists, a hyphen (-) is displayed.	http://xyz.com
http_user_agent	The User-Agent of the request.	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build/HUAWEIEDISON-AL10)
http_x_forwarded_for	The IP address of the upstream user redirected by proxy.	N/A

Field	Description	Example
https	Indicates whether the request is an HTTPS request. · true: The request is an HTTPS request. · false: The request is an HTTP request.	true
matched_host	The matching source site, which may be a wildcard domain name. If no match is found, a hyphen (-) is displayed.	*.zhihu.com
real_client_ip	The real IP of the visitor. If no real IP is returned, a hyphen (-) is returned.	1.2.3.4
isp_line	Line information, such as BGP, China Telecom, and China Unicom.	China Telecom
remote_addr	The IP address of the client that initiates the connection request.	1.2.3.4
remote_port	The port number of the client that initiates the connection request.	23713
request_length	The size of the request in bytes.	123
request_method	The HTTP method of the request.	GET
request_time_msec	The request time in ms.	44
request_uri	The request URI.	/answers/377971214/banner
server_name	The name of the matching host. If no match is found, the value is `default`.	api.abc.com
status	The HTTP status code.	200

Field	Description	Example
time	The time when the log entry was generated.	2018-05-02T16:03:59+08:00
cc_action	The HTTP flood protection action. Valid values include none, challenge, pass, close, captcha, wait, and login.	close
cc_blocks	Indicates whether HTTP flood attacks are blocked. Valid values: · 1: block · Other values: pass	1
cc_phase	The HTTP flood protection policy. Valid values include seccookie, server_ip_blacklist, static_whitelist, server_header_blacklist, server_cookie_blacklist, server_args_blacklist, and qps_overmax.	server_ip_blacklist
ua_browser	The browser.	ie9
ua_browser_family	The browser series.	internet explorer
ua_browser_type	The browser type.	web_browser
ua_browser_version	The browser version.	9.0
ua_device_type	The type of the client device.	computer
ua_os	The operating system of the client.	windows_7
ua_os_family	The operating system series of the client.	windows
upstream_addr	The list of origin addresses that are separated with commas (,). Each address is in the format of `IP : Port`.	1.2.3.4:443
upstream_ip	The real origin IP address.	1.2.3.4

Field	Description	Example
upstream_response_time	The response time in seconds for the back-to-origin process.	0.044
upstream_status	The HTTP status of the back-to-origin request.	200
user_id	The user ID of the Alibaba Cloud account.	12345678
querystring	The request string.	token=bbcd&abc=123

4.5.5 Analyze the logs of BGP-line Anti-DDoS Pro

On the Full Log page of the BGP-line Anti-DDoS Pro console, the Log Analyses and Log Reports pages of Log Service are embedded. After you activate the Full Log service for BGP-line Anti-DDoS Pro, you can process collected logs in real time on the Full Log page. Specifically, you can search and analyze logs in real time, view or edit a dashboard, configure an alarm, and perform other operations.

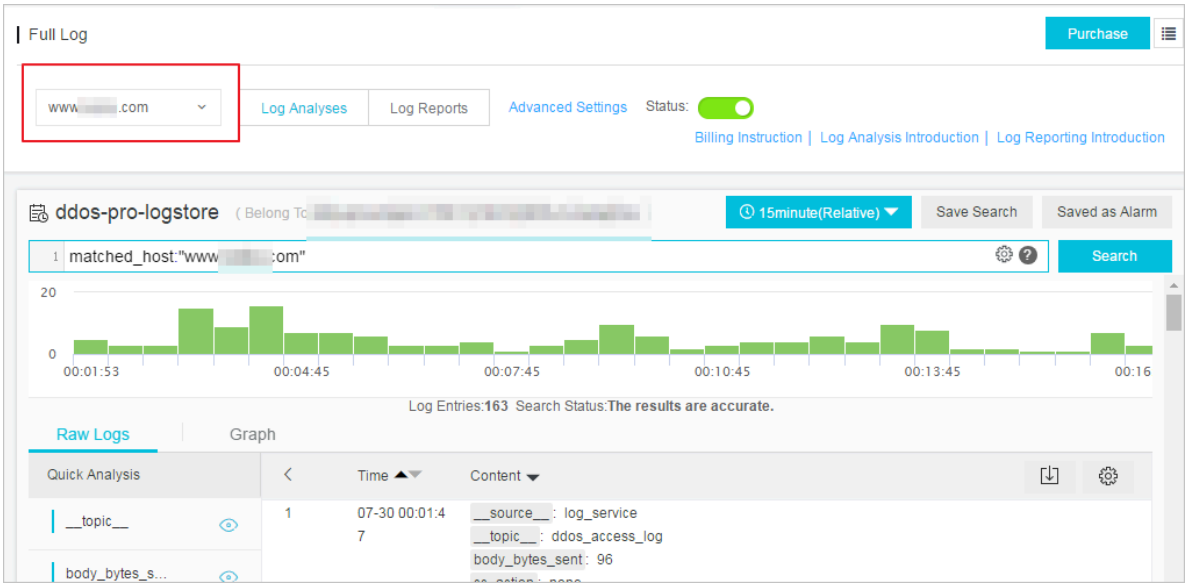
Procedure

1. Log on to the BGP-line Anti-DDoS Pro console.
2. In the left-side navigation pane, choose System > Full Log.
3. Select the domain for which you want to analyze logs, and confirm that the Status switch is enabled.

4. Click Full Log.

   On this page, the Search & Analysis page of Log Service is embedded. The system automatically executes a search statement to view the logs of the target website. For example, `matched_ho st :" 0523 . yuanya . aliyun . com "`.

   Figure 4-21: Analyze logs

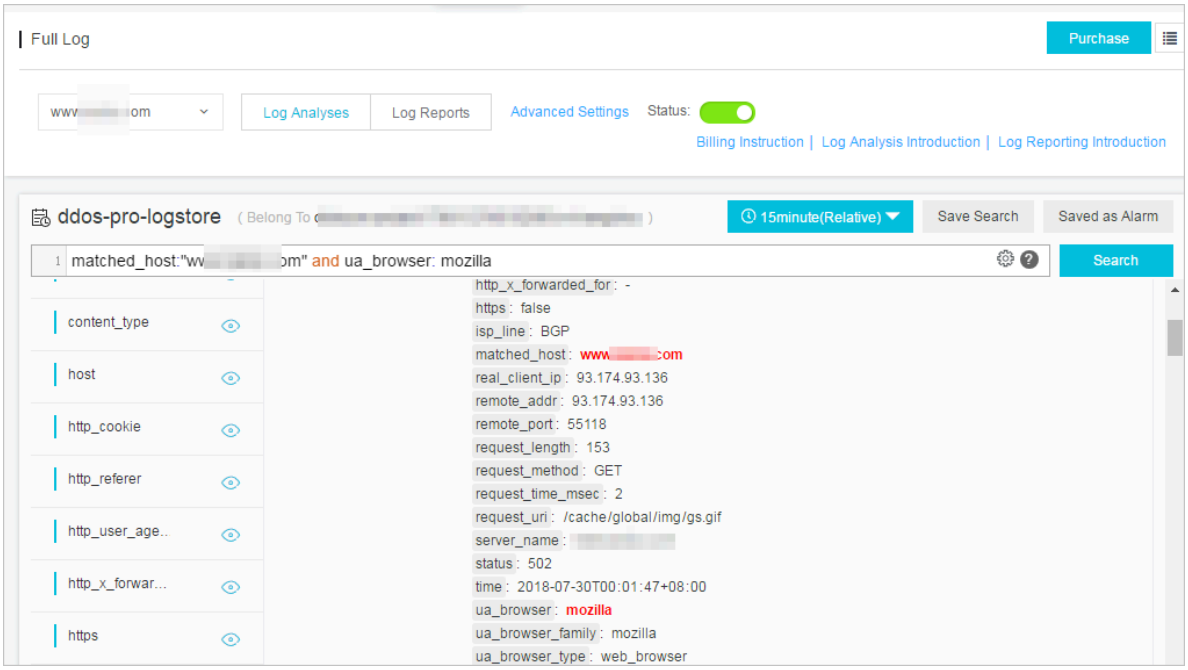


5. Enter a search statement, select a time range, and then click Search & Analysis.

   Note:

Logs of BGP-line Anti-DDoS Pro can be retained for 180 days.

Figure 4-22: Log query



For the logs obtained on the Search & Analysis page, you can perform the following operations:

- Customize search and analysis

  Log Service provides the syntax used to search and analyze logs for different scenarios. For more information, see*Query syntax*.

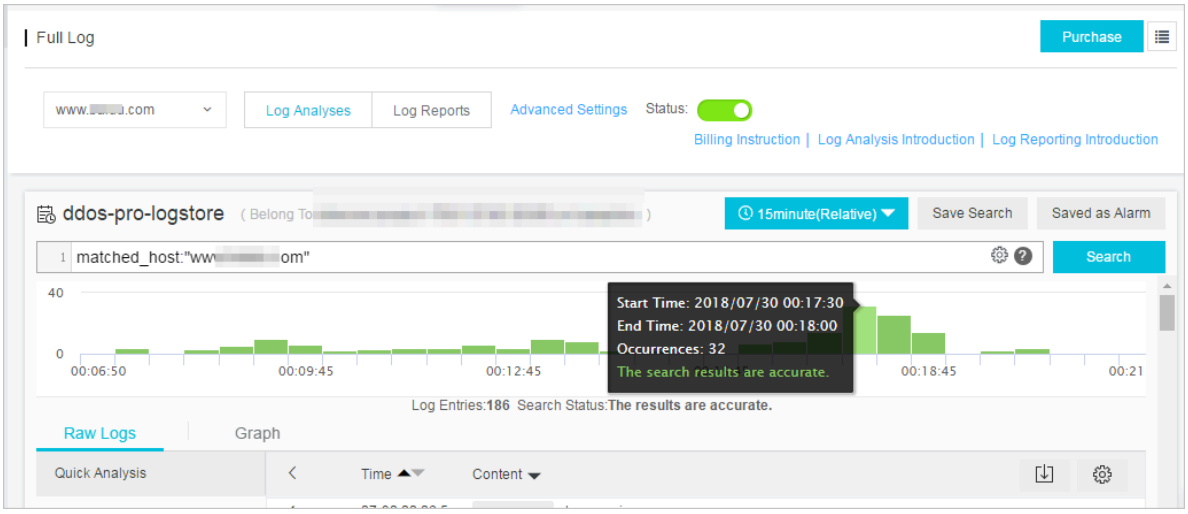
- View log time distribution

  The area under the search box displays the time distribution of the logs that meet the requirements of the query time and the search statement. Time distribution is displayed in the form of a histogram with the horizontal and vertical axis. The total number of queried logs is also displayed.

Note:

You can slide the histogram to select a smaller time range. Then, the `time selector` automatically updates the selected time range and refresh the result.

Figure 4-23: Log time distribution



- View raw logs

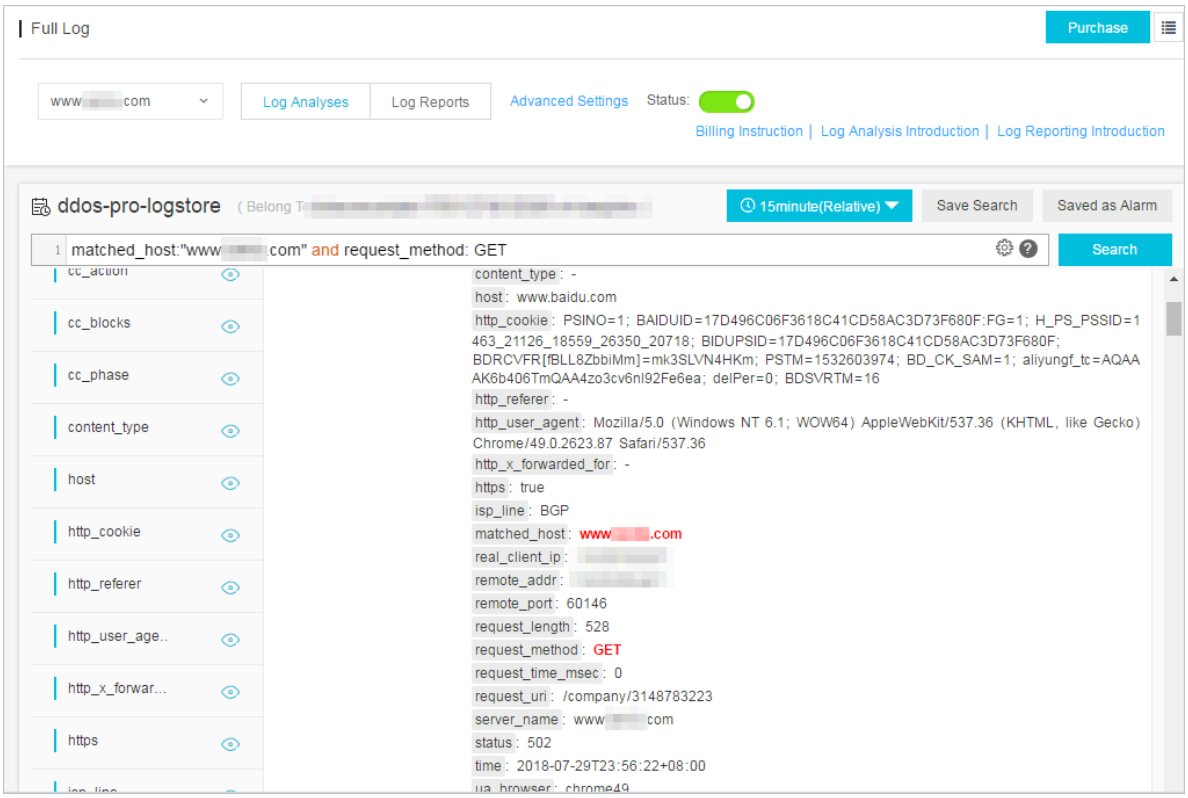
  The Raw Logs tab page displays the details of each log, including the time and content of each log. The log content contains all the fields of a log. You can sort the columns, download the current query results, or click the gear icon to display specific fields.

  You can click the value of a filed contained in the log content. Then, the system will automatically add the corresponding query condition to the original statement.

For example, if you click `GET` in `request_me thod : GET`, the following statement is automatically added to the search box:

```
The original search statement and request_me thod : GET
```

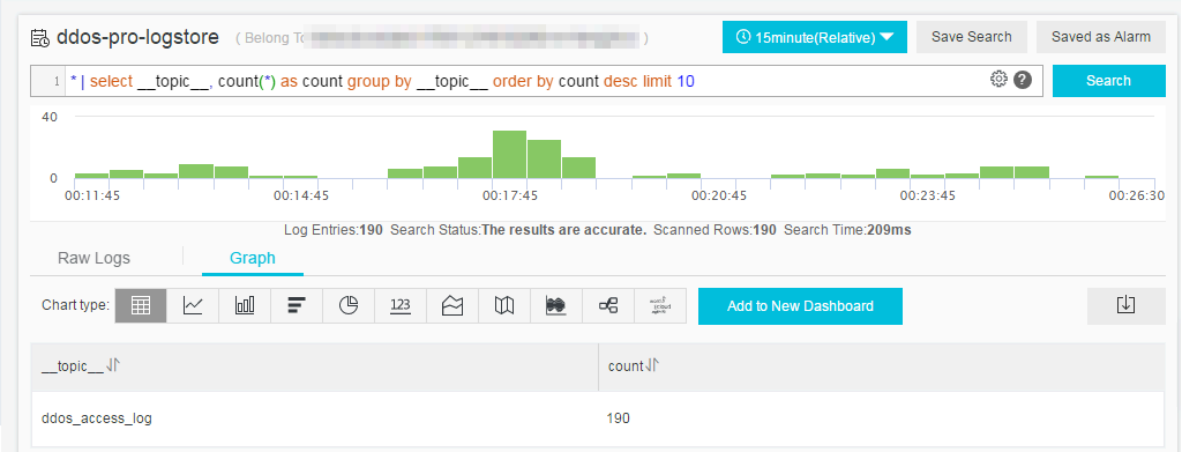
Figure 4-24: Raw logs



- View charts

  Log Service supports charts to display analysis results of logs. You can select different chart types on the Graph page. For more information, see *Query syntax*.

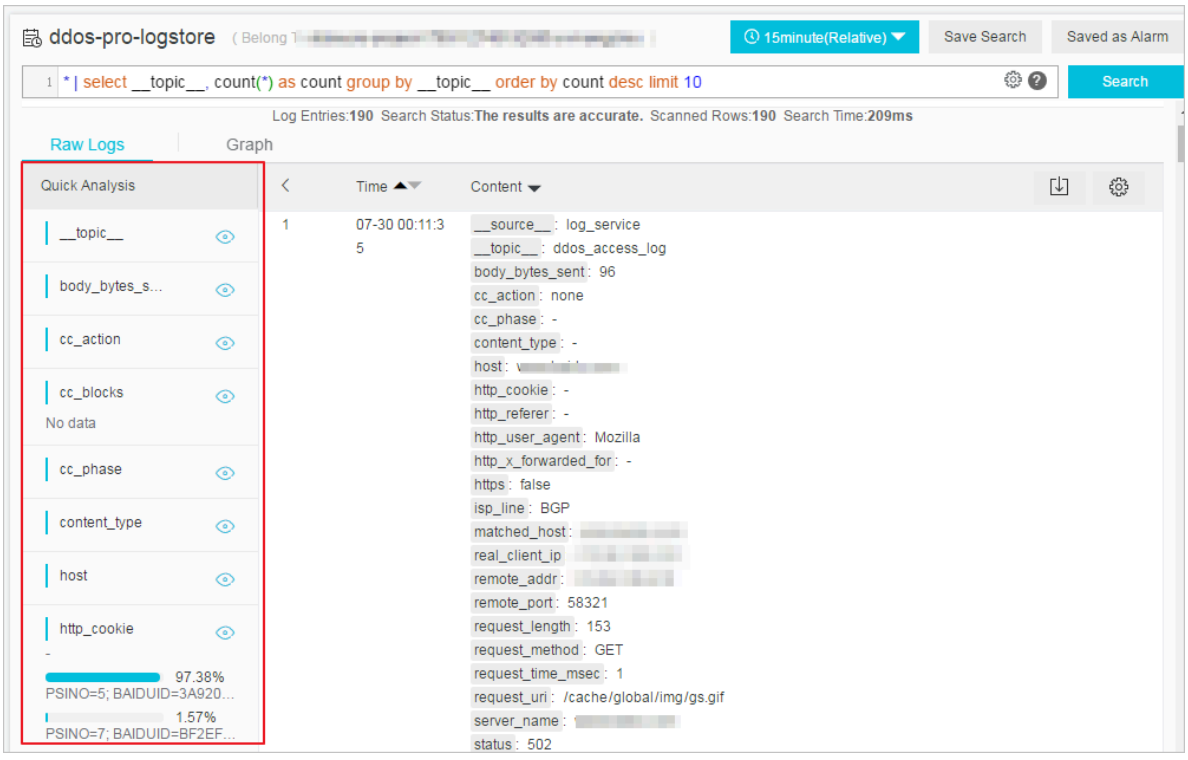
  Figure 4-25: Chart



- Quick analysis

  The quick analysis feature provides one-click query that helps you quickly analyze the distribution of a field over a period of time and reduce the time cost of indexing critical data. For more information, see *Quick analysis*.

  Figure 4-26: Quick analysis



Custom search and analysis

A query statement consists of two parts: query syntax (`Search`) and analysis syntax (`Analytics`), which are separated by `|`.

```
$ Search  | $ Analytics
```

Type	Description
Query (`Search`)	The query conditions can be generated by keywords, fuzzy, numerical values, interval range and combination conditions. If this field is empty or set to `*`, all data is displayed.
Analysis (`Analytics`)	Calculate and count the query results or all of the log data.



Note:

Both `Search` and `Analytics` are optional. If `Search` is set to empty, all the data in the specified period is not filtered and the results are counted directly. If `Analytics` is set to empty, the query results are returned and no statistics are collected.

Query syntax

The Log Service query syntax supports Full-text query and Field query. Query box supports line break display, syntax highlighting, and other functions.

- Full-text query

  No field is required. You can directly enter one or more keywords included in quotation marks (""). You can use a space or `and` to separate two keywords.

  Example:

  - Multiple keywords query

    Search for logs containing `www . aliyun . com` and `error`. For example:

    ```
    www . aliyun . com   error
    ```

    or

    ```
    www . aliyun . com   and   error
    ```

  - Conditional query

    Search for logs that contain `www . aliyun . com`, and `error` or `404`. For example:

    ```
    www . aliyun . com   and  ( error   or   404 )
    ```

- Prefix query

  Search for logs that contain `www . aliyun . com`, and keywords starting with `failed_`. For example:

  ```
  www . aliyun . com   and   failed_ *
  ```

   Note:
  You can only add an asterisk (*) after a keyword. No asterisk (*) can be added before a keyword (for example, `* _error`).

- **Field query**

  Log Service supports more accurate queries based on fields.

  A comparison of numeric type fields can be implemented in the format `field : value` or `field >= value`, using `and`, `or`. It can also be combined with full-text search, also by using the combination of `and` and `or`.

  BGP-line Anti-DDoS Pro website access logs and attack logs can also be queried based on field query.

  Example

  - Multiple fields query

    Search for logs containing `www . aliyun . com` attacked by CC:

    ```
    matched_ho st : www . aliyun . com and cc_blocks : 1
    ```

    Search the access logs containing the error 404 of a client `1 . 2 . 3 . 4` on the website `www . aliyun . com`:

    ```
    real_clien t_ip : 1 . 2 . 3 . 4 and matched_ho st : www .
    aliyun . com and status : 404
    ```

     **Note:**

Fields used in the examples `matched_ho st` , `cc_blocks` , `real_clien t_ip` , and `status` are fields of access and attack logs.

- Numeric field query

  Search for all slow request logs with a response time of more than 5 seconds:

  ```
  request_ti me_msec > 5000
  ```

  Interval queries are also supported. You can query logs with a response time greater than 5 seconds and less than or equal to 10 seconds:

  ```
  request_ti me_msec in ( 5000 10000 ]
  ```

  The query can also be performed by the following statement:

  ```
  request_ti me_msec > 5000 and request_ti me_msec <=
  10000
  ```

- Check whether whether a specific log exists

  ■ Query logs that contain the `ua_browser` field: `ua_browser : *`.

  ■ Query logs that do not contain the `ua_browser` field: `not ua_browser : *`

For more information, see *Index and query overview*.

Analysis syntax

You can use the SQL/92 syntax for log data analysis and statistics. For more information about the syntax and functions supported by Log Service, see *Syntax description*.



Note:

- The `from table name` statement in the SQL standard syntax can be omitted from the analysis statement, that is, `from log` .
- Log data returns the first 100 entries by default, and you can modify the return range by *LIMIT syntax*.

Time-based log query analysis

Each BGP-line Anti-DDoS Pro log has a `time` field, in the format `year - month - day T hour : minute : second + time zone` . For example, `2018 - 05 - 31T20 : 11 : 58 + 08 : 00` , where the time zone is `UTC + 8` , that is

Beijing time. At the same time, each log has a built-in field: `__time__`, which also indicates the time of this log, so that time-based calculations can be performed in statistics. The format is *Unix timestamp*. The essence is a cumulative number of seconds since the 1970- 1 0:0:0 UTC time. Therefore, in actual use, after calculation, time must be formatted before it can be displayed.

- Select and show time

  Over a specific period of time, select the latest 10 logs of the website `www . aliyun . com` attacked by CC, show the time, source IP and access client, using the `time` field directly:

```
matched_ho  st :  www . aliyun . com   and   cc_blocks :  1
|   select   time ,  real_clien  t_ip ,  http_user_  agent
     order   by   time   desc
     limit   10
```

- Calculate time

  To query the number of days after the CC attack, use `__time__` to calculate:

```
matched_ho  st :  www . aliyun . com    and   cc_blocks :  1
|   select   time ,
          round (( to_unixtim  e ( now ()) -  __time__ )/ 86400
,  1 )  as  " days_passe  d ",              real_clien  t_ip ,
http_user_  agent
      order   by   time   desc
      limit   10
```

  Note:

  Use `round (( to_unixtim  e ( now ()) -  __time__ )/ 86400 ,  1 )`, first part `to_unixtim  e` , the time obtained by `now ()`, is converted to a Unix timestamp, and subtracted from the built-in time field `__time__` to get the number of seconds that have passed. Finally, divide by `86400` , which is the total number of seconds in a day, and then round it to the decimal with the function `round ( data ,  1 )`. One-digit value indicates that each attack log has passed a few days.

- Group statistics based on specific time

  If you want to know how a website is being attacked by CC every day for a specific time frame, use the following SQL:

```
matched_ho  st :  www . aliyun . com    and   cc_blocks :  1
|  select   date_trunc (' day ',  __time__ )  as   dt ,
         count ( 1 )  as   PV
```

```
        group   by   dt
        order   by   dt
```

> **Note:**
> This example uses the built-in time field `__time__` to pass the function `date_trunc (' day ', ..)` to the time alignment. Each log is grouped into the partition of the day it belongs to for the total number of statistics (count(1)) and sorted by partition time block. The first argument of the function `date_trunc` provides alignment for other units, including `second` , `miniute` , `hour` , `week` , `month` , `year` . For more information about function, see *Date and time functions*.

- Time-based group statistics

  For more flexible grouping time rules, for example, to know the trend of a website being attacked by CC every five minutes the math calculations are required. Run the following SQL:

```
 matched_ho  st :  www . aliyun . com    and   cc_blocks :  1
|  select   from_unixt  ime ( __time__  -  __time__ %  300 )  as
 dt ,
          count ( 1 )  as   PV
       group   by   dt
       order   by   dt
       limit   1000
```

> **Note:**
> Use the built-in time field to calculate `__time__ - __time__ % 300` and format it using the function `from_unixt ime` . Each log is grouped into a 5 minute (300 seconds) partition for the total number of statistics (count(1)), and sorted by partition time block to obtain the first 1000 logs, which is equivalent to the first 83 hours of data in the selection time.

More time-resolved functions, such as converting a time format, require using `date_parse` and `date_forma t` . For more information, see *Date and time functions*.

Client IP-based query analysis

Logs of BGP-line Anti-DDoS Pro contain a field `real_clien t_ip` . However, if the user cannot obtain the real IP through the proxy and the IP address in the header is incorrect, you can use the `remote_add r` field to directly connect to the client IP.

- Country distribution of attackers

  Distribution of source countries of CC attacks on a website:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_coun try ( if ( real_clien t_ip ='-',
remote_add r , real_clien t_ip )) as country ,
        count ( 1 ) as " number of attacks "
        group by country
```

  Note:

  Use the function `if ( condition , option1 , option2 )` to select the field `real_clien t_ip` or `real_clien t_ip` (when `real_clien t_ip` is -). Pass the obtained IP to the function `ip_to_coun try` to get the country information corresponding to this IP.

- Province distribution of users who access the website

  To get more detailed province-based distribution, use the `ip_to_prov ince` function, for example:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_prov ince ( if ( real_clien t_ip ='-',
remote_add r , real_clien t_ip )) as province ,
        count ( 1 ) as " number of attacks "
        group by province
```

  Note:

  Another IP function `ip_to_prov ince` to get a province of IP. If IP address is outside of China, system still tries to convert to the province (state).

- Attackers heat distribution

  To get an attackers heat map, use the `ip_to_geo` function, for example:

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_geo ( if ( real_clien t_ip ='-', remote_add r
, real_clien t_ip )) as geo ,
        count ( 1 ) as " number of attacks "
        group by geo
        limit 10000
```

  Note:

  Use another IP function `ip_to_geo` to get the latitude and longitude of an IP and get the first 10,000.

More IP-based parsing functions, such as obtaining the IP operator `ip_to_provider`, determining whether the IP is Internet or Intranet `ip_to_domain`, see *IP functions*.

4.5.6 Log reports of BGP-line Anti-DDoS Pro

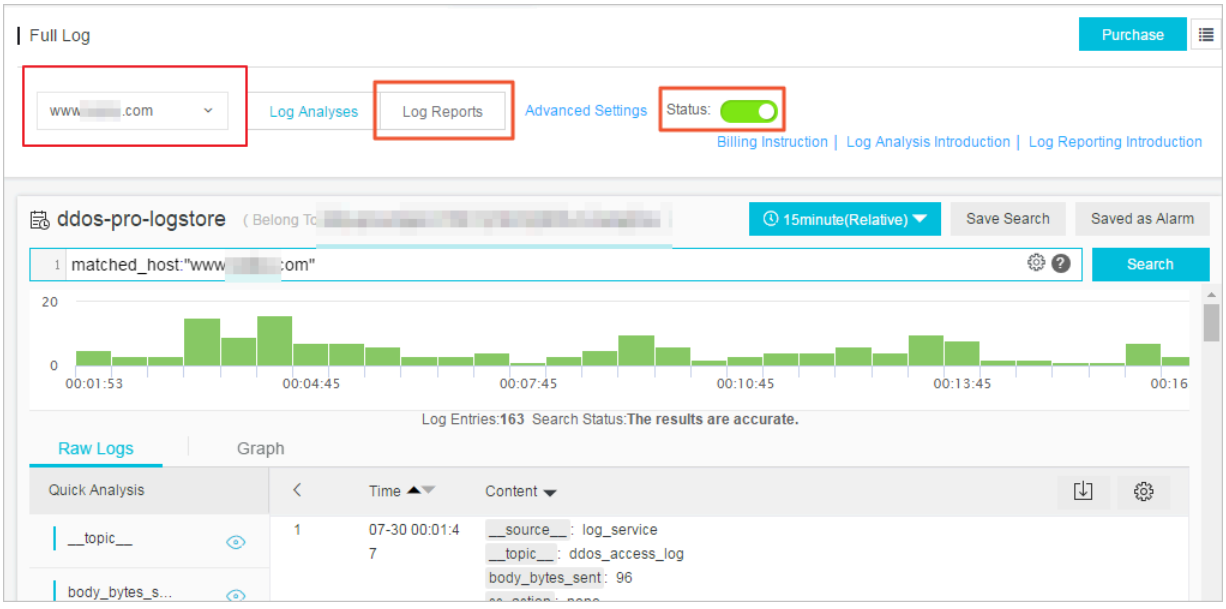
The Dashboard page of Log Service is embedded in the Log Reports page. This page displays the default dashboard. You can view log data in the dashboard by modifying the time range and adding conditions to filter logs.

View log reports

1. Log on to the BGP-line Anti-DDoS Pro console.
2. In the left-side navigation pane, choose System > Full Log.
3. Select the domain for which you want to view the log report, and confirm that the Status switch is enabled.
4. Click Log Reports.

   On this page, the Dashboard page of Log Service is embedded. The system automatically executes a search statement to view the logs of the target website. For example, `matched_host :" 0523 . yuanya . aliyun . com "`.

Figure 4-27: View log reports



After the BGP-line Anti-DDoS Pro log collection function is enabled for the website, Log Service automatically creates two default dashboards: operation center and access center.

Dashboard	Description
Operation center	Displays the current overall operational status of websites protected by BGP-line Anti-DDoS Pro, including valid request status, traffic, trends, attack distributions, and traffic volumes and peaks attacked by CC.
Access center	Displays the current overall operational status of websites protected by BGP-line Anti-DDoS Pro, including PV/UV trends and bandwidth peaks, visitors , traffic, client type, request, and visited websites distribution.

In addition to viewing log reports, you can also perform the following operations:

- Select *time range*
- Add or edit *filter condition*
- View *charts*

Time selector

All charts on the dashboard page are based on statistical results for different time periods. For example, the default time range for visits is one day and the access trend is 30 days. To set all charts on the current page to be displayed in the same time range, you can configure the time selector.

1. Click Select.
2. Configure the settings in the dialog box. You can select relative time, entire point time, or set a custom time.



Note:

- When the time range is modified, the time of all charts is changed to this time range.
- Time picker only provides a temporary view of the chart on the current page, and the system does not save the setting. The next time you view the report, the system will display the default time range.

Conditions to filter logs

Select a website and click Log Reports to enter the dashboard page. System automatically adds filter condition, such as `matched_ho st :" 0523 . yuanya . aliyun . com "` to view log reports based on selected website.

You can modify the data display range of the report by setting filter condition.

- View overall reports for all websites

  Clear the filter condition to display the overall reports for the Logstore.
- Add more filter conditions

  You can filter the report data by setting key and value. `AND` relationship between multiple filters is supported.

  For example, view the overall situation of access requests by telecommunications lines.

   Note:
  The `isp_line` is the field of the logs of BGP-line Anti-DDoS Pro, indicating the operator network that connects to the port.

Chart

The report display area shows multiple reports according to a predefined layout, including the following types. For more information about chart types, see *Graph description*.

Type	Description
Number	Displays important indicators, such as effective request rate and attack peaks.
Line or area map	Displays trend graphs for certain important indicators within a specific time period, such as inbound bandwidth trends and attack interception rates.
Map	Displays the geographical distribution of visitors and attackers, such as CC attack country and access hotspot .
Pie chart	Displays the distribution of the information, such as the top 10 of the websites being attacked and client type distribution.

Type	Description
Table	Displays information such as the list of attackers, typically divided into multiple columns.
Map	Displays the geographical distribution of data.

Default dashboards

- Operation center

  This dashboard displays the current overall operational status of the websites protected by BGP-line Anti-DDoS Pro, including valid request status, traffic, trends, attacker distributions, and traffic volumes and peaks attacked by CC.

Chart	Type	Default time range	Description	Example
Valid request package rate	Single value	1 hour (relative)	A valid request, that is, the number of non-CC attacks or 400 error requests in the total number of all requests.	95%
Valid request flow rate	Single value	1 hour (relative)	The percentage of valid requests in the all requests.	95%
Received traffic	Single value	1 hour (relative)	The sum of valid request inflows, in MB.	300 MB
Attack traffic	Single value	1 hour (relative)	The sum of inbound traffic of CC attacks. The unit is MB.	30 MB
Outbound traffic	Single value	1 hour (relative)	The sum of valid request outbound traffic. The unit is MB.	300 MB

Chart	Type	Default time range	Description	Example
Network incoming bandwidth peak.	Single value	1 hour (relative)	The highest peak of incoming traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Network outgoing bandwidth peak.	Single value	1 hour (relative)	The highest peak of outbound traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Received data packets	Single value	1 hour (relative)	The number of incoming requests for valid requests (non-CC attacks), measured in units.	30, 000
Attack data packets	Single value	1 hour (relative)	The sum of the number of requests for the CC attack.	100
Attack peak	Single value	1 hour (relative)	The highest peak of CC attack. The unit is number per minute.	100 per minute

Chart	Type	Default time range	Description	Example
Inbound bandwidth and attack trends	Two-line diagram	1 hour (entire point)	Trend chart of valid requests per minute and traffic bandwidth for attack requests. The unit is KB/s.	-
Request and interception trends	Two-line diagram	1 hour (entire point)	Trend chart of the total number of requests and intercepted CC attack requests per minute. The unit is number per minute.	-
Valid request rate trend	Two-line diagram	1 hour (entire point)	Trend chart of the number of valid requests per minute (non-CC attacks or 400 error requests) in the total number of all requests.	-
Access status distribution trend	Flow chart	1 hour (entire point)	Trend chart of various request processing statuses (400, 304, 20) per minute. The unit is number per minute.	-

Chart	Type	Default time range	Description	Example
CC attacks distribution	World map	1 hour (relative)	The sum of the number of CC attacks in the source country.	-
CC attack distribution	Map of China	1 hour (relative)	The sum of the number of CC attacks in the source province (China).	-
List of attacks	Table	1 hour (relative)	The attacker information of the first 100 attacks, including IP, city, network, number of attacks, and total traffic.	-
Attack access line distribution	Pie chart	1 hour (relative)	CC attack source access BGP-line Anti-DDoS Pro distribution, such as telecommunications, Unicom, and BGP.	-
Top 10 attacked websites	Donut chart	1 hour (relative)	Top 10 attacked websites	-

- Access center

  This dashboard displays the current overall operational status of the websites protected by BGP-line Anti-DDoS Pro, including PV/UV trends and bandwidth peaks, visitors, traffic, client type, request, and visited websites distribution.

Chart	Type	Default time range	Description	Example
PV	Single value	1 hour (relative)	The total number of requests.	100,000
UV	Single value	1 hour (relative)	Total number of independent access clients.	100,000
Inbound traffic	Single value	1 hour (relative)	The sum of inbound traffic of the website. The unit is MB.	300 MB
Network in bandwidth peak.	Single value	1 hour (relative)	The highest peak of inbound traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Network out bandwidth peak.	Single value	1 hour (relative)	The highest peak of inbound traffic rate requested by the website. The unit is bytes/s.	100 Bytes/s
Traffic bandwidth trend	Two-line diagram	1 hour (entire point)	Trend chart of website inbound and outbound traffic per minute. The unit is KB/s.	-

Chart	Type	Default time range	Description	Example
Request and interception trends	Two-line diagram	1 hour (entire point)	Trend chart of the total number of requests and intercepted CC attack requests per minute. The unit is number per minute.	-
PV/UV access trends	Two-line diagram	1 hour (entire point)	Trend chart of PV and UV per minute. Measured in units.	-
Visitor distribution	World map	1 hour (relative)	The distribution of visitors PV (page view) in the source country.	-
Visitor heat map	Amap	1 hour (relative)	Visitor geographic access heat map.	-
Inbound traffic distribution	World map	1 hour (relative)	Sum of inbound traffic distribution in the source country. The Unit is MB.	-
Inbound traffic distribution	Map of China	1 hour (relative)	Sum of inbound traffic distribution in the source province. The Unit is MB.	-

Chart	Type	Default time range	Description	Example
Access line distribution	Donut chart	1 hour (relative)	Source-based access BGP-line Anti-DDoS Pro protection line distribution, such as telecommunications, Unicom, and BGP.	-
Inbound traffic network provider distribution	Donut chart	1 hour (relative)	The distribution of inbound traffic that visitors access by network operators. For example, telecommunications, Unicom, mobile connections, education network. The Unit is MB.	-
Most visited clients	Table	1 hour (relative)	The top 100 most visited clients, including IP, city, network, request method distribution, incoming traffic, number of incorrect accesses, number of intercepted CC attacks.	-

Chart	Type	Default time range	Description	Example
Access domain name	Donut chart	1 hour (relative)	The top 20 most visited domain names.	-
Referer	Table	1 hour (relative)	The top 100 most redirected referer URLs, hosts, and frequency.	-
Client type distribution	Donut chart	1 hour (relative)	The top 20 most visited user agents, such as iPhone, iPad, Windows IE, Chrome.	-
Request content type distribution	Donut chart	1 hour (relative)	The top 20 most requested content types, such as HTML, Form, JSON, streaming data.	-

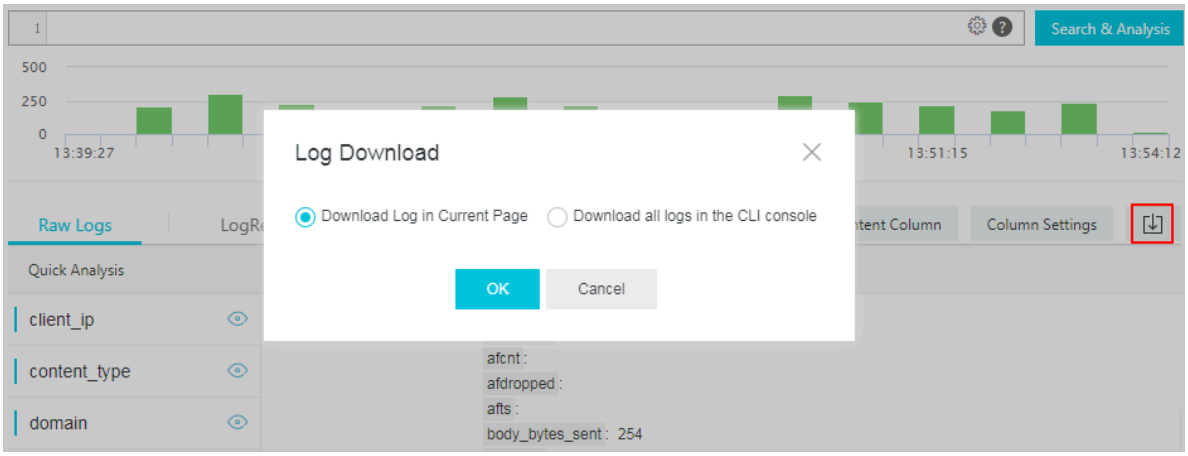
4.5.7 Advanced settings

The Anti-DDoS Pro log collection function supports advanced management. You can click Advanced Management to go to the Logstores page in the Log Service console. You can perform advanced operations on Anti-DDoS Pro logs, including real-time subscription and consumption, data shipping, and other visualization operations.

In the upper-right corner of the Log Service page, click Advanced Management to go to the Log Service console. You can perform advanced operations including exporting log data and configuring log consumption.

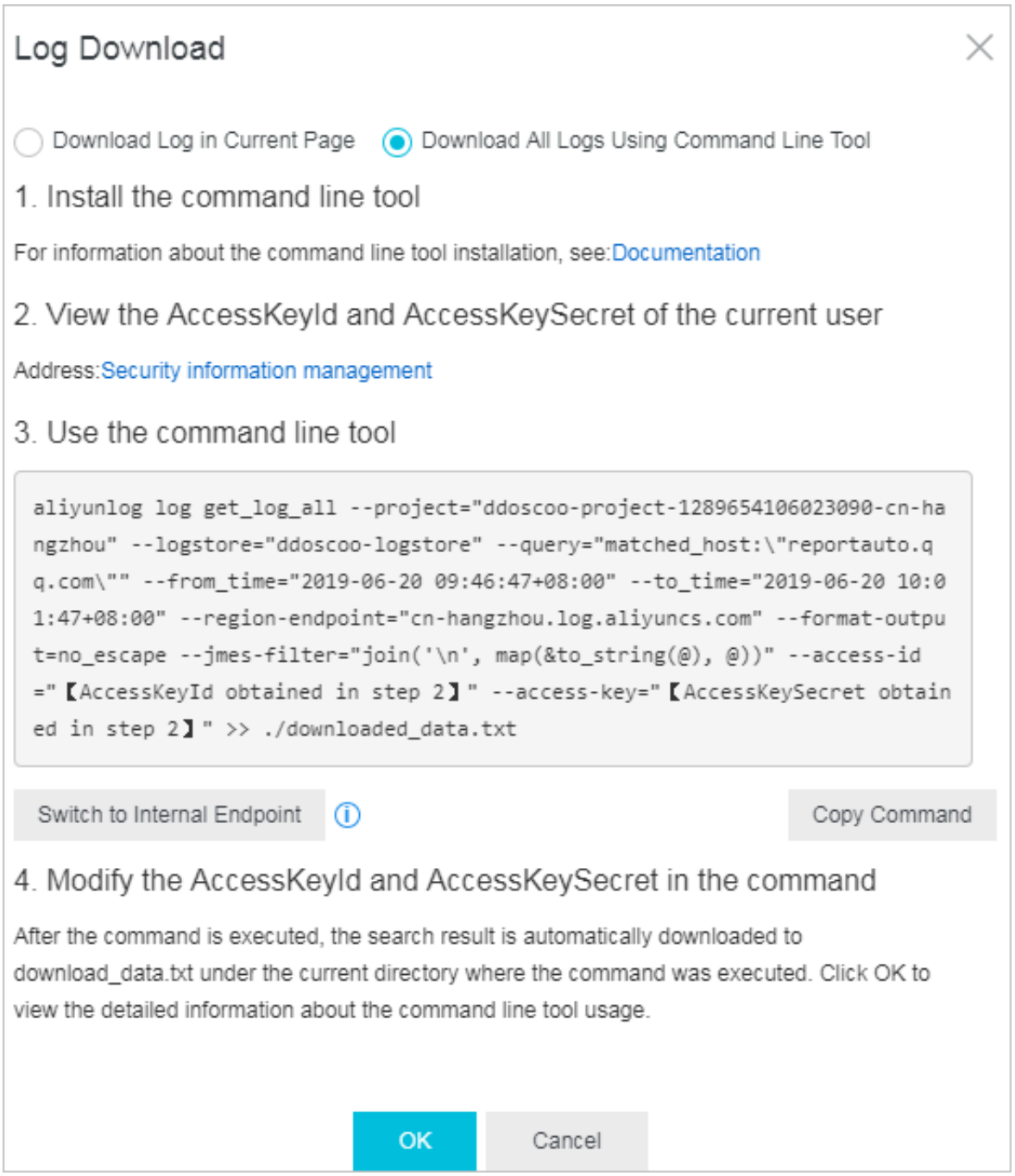
Export log data

1. After you enable log collection, click the Download button on the right of the Raw Logs tab page.



2. In the Log Download dialog box that appears, click Download Log in Current Page to export the log entries displayed on this page into a file in CSV format.

3. You can also click Download All Logs Using Command Line Tool to download all log entries.



a. Click Documentation to see *User Guide for Alibaba Cloud CLI*.

b. Install the command line tool.

c. Click *Security information management* to view and copy the AccessKey ID and AccessKey Secret of the current user.

d. Click Copy Command and replace `AccessKeyI d obtained in step 2` and `AccessKeyS ecret obtained in step 2` with the AccessKey ID and AccessKey Secret of the current user.

e. Run the command in the command line tool.

After you run the command, log entries are automatically downloaded and saved to the download_data.txt file in the directory where the command was run.

Other advanced operations

- *Alerts and notifications*
- *Real-time subscription and consumption*
- *Data shipping*
- *Integration with other visualization tools*

4.6 TDS logs

Alibaba Cloud Threat Detection Service (TDS) provides a log analysis function to collect, analyze, query, store, and distribute risk and threat data in real time. This frees you from the need to manually collect, query, and analyze data, improving your overall O&M efficiency.

Features

Overview

Alibaba Cloud TDS is fully integrated with Log Service and provides TDS log collection and analysis functions, which can help you better understand and more effectively address server security risks and manage your assets on the cloud. TDS is suitable for the following enterprise-level scenarios:

- Large-scale enterprises and organizations, such as finance companies and government agencies, which require strict storage compliance for hosts, networks, and security logs, among other assets on the cloud
- Large-scale real-estate, e-commerce, or finance companies, along with government agencies, which posses on-premises security operations centers (SOCs) and require centralization collection and management of security and alarm logs
- Enterprises with advanced technologies, such as companies in IT, gaming, or finance, which require in-depth analysis of logs collected from various cloud assets and automated alarm handling

Benefits

- Quick analysis capabilities: The analysis of security and host logs can be completed in seconds, and analysis of network logs within an hour.
- Comprehensive support: A total of 14 log types are provided, including network, host, and security logs.
- Fully integrated: TDS is fully integrated with the open-source streaming and big data system solutions on Alibaba Cloud and is publicly open to our partners.
- Flexible to various applications: With support for WYSIWYG analysis capabilities, you can customize service views as needed.

Limits

- TDS-dedicated Logstores cannot store non-TDS data.

  TDS logs are stored in dedicated Logstores. These Logstores cannot store non-TDS data that is written through APIs or SDKs. Dedicated Logstores have no limits on queries, statistics, alarms, and stream consumption.
- Basic settings, such as the storage period of dedicated Logstrores, cannot be modified.
- Dedicated Logstores do not incur charges.

  Dedicated Logstores do not incur charges on the condition that the Log Service functions normally.

   Note:
  The TDS log analysis function is unavailable in the case that Log Service charges are overdue. In such s case, you need to first pay your overdue payments before you can gain access to this function.

Scenarios

- Track host and network logs and trace the source of security threats.

  You can retrieve the `__topic__` field in logs and view the time distribution of different types of logs to track host and network logs in real time.
- View host and network operations in real time to gain insight into security status and trends.

  You can view host and network operations in the Web access center dashboard to assess the security of your assets in a timely manner.

- Understand security operating efficiency and handle issues and threats in a prompt manner.

  You can view your current security operating efficiency in the vulnerability center dashboard.

4.7 WAF logs

4.7.1 Real-time log analysis

Integrated with Log Service, WAF provides access logs and attack logs, and allows you to analyze logs in real time.

The real-time log analysis feature in WAF collects and stores access logs in real time and provides the following capabilities based on Log Service: log querying, analysis, reporting, alerting, forwarding, and computing. The service makes it easy to search log data so that you can focus on log analysis.

Target users

- Large enterprises and institutions that need to meet compliance requirements regarding the use of cloud hosts, networks, and the storage of security logs, such as financial companies and government agencies.
- Enterprises that have private security operations centers (SOCs) and need to collect security logs for centralized operations and management, such as large real estate , e-commence, financial companies, and government agencies.
- Enterprises that have strong technical capabilities and need to perform in-depth analysis on logs of cloud resources, such as IT, gaming, and financial companies.
- Small and medium-sized enterprises and institutions that need to meet compliance requirements regarding their business on the cloud or need to generate business reports on a regular basis, such as monthly, quarterly, and annual reports.

Benefits

- Compliance: Stores the website's access logs for more than six months to help the website meet the compliance requirements.
- Simple configuration: You can easily configure the service to collect access logs and attack logs on your site.

- Real-time analysis: Integrated with Log Service, the service supports real-time log analysis and offers a ready-to-use report center. You can easily gain information about the details of attacks, and visits to your site.
- Real-time alarms: Near real-time monitoring and custom alarms based on specific metrics are available to ensure a timely response to critical service failures.
- Collaboration: The service can be integrated with real-time computing, cloud storage, visualization, and other data solutions to help you gain valuable insights into your data.

Prerequisites and limits

To use the real-time analysis feature in WAF, you must meet the following prerequisites:

- You have activated *Log Service*.
- You have activated *WAF Enterprise Edition* and enabled the log analysis module.

All log data in WAF is stored in an exclusive logstore that has the following limits:

- Users cannot use APIs or SDKs to write data to the logstore or change attributes of the logstore, such as the storage period.

   Note:
  The logstore supports common features, including log querying, reporting, alerting, and stream computing.
- The logstore is free of charge on condition that Log Service is available and your account has no overdue payments.
- The system reports may be updated at irregular intervals.

Scenarios

- Analyze log data to track attacks and identify threats.
- Monitor Web requests in real time to predict traffic trends.
- Quickly learn about the efficiency of security operations and obtain timely feedback.
- Transfer network logs to user-created data centers or computing centers.

4.7.2 Billing method

Web Application Firewall (WAF) Log Service is billed based on the log storage period and the log storage size of your choice.

WAF Log Service is activated on a subscription basis.

> Note:
> To activate WAF Log Service, you must buy a WAF subscription.

In the WAF purchase page, enable Activate Log Service and select the log storage period and the log storage size. Then, the price is automatically calculated based on the log store specification of your choice and the validity of the WAF instance.

Log storage specification

The detailed pricing for each log storage specification for WAF Log Service is shown in the following table.

Log storage period	Log storage size	Recommended scenarios	For International region instances		For Mainland China region instances	
			Monthly subscription	Yearly subscription	Monthly subscription	Yearly subscription
180 days	3 TB	Average daily QPS is up to 80.	USD 450	USD 5,400	USD 225	USD 2,700
	5 TB	Average daily QPS is up to 120.	USD 750	USD 9,000	USD 375	USD 4,500
	10 TB	Average daily QPS is up to 260.	USD 1,500	USD 18,000	USD 750	USD 9,000
	20 TB	Average daily QPS is up to 500.	USD 3,000	USD 36,000	USD 1,500	USD 18,000
	50 TB	Average daily QPS is up to 1,200.	USD 7,500	USD 90,000	USD 3,000	USD 36,000
	100 TB	Average daily QPS is up to 2,600.	USD 15,000	USD 180,000	USD 7,500	USD 90,000

Log storage period	Log storage size	Recommended scenarios	For International region instances		For Mainland China region instances	
			Monthly subscription	Yearly subscription	Monthly subscription	Yearly subscription
360 days	5 TB	Average daily QPS is up to 60.	USD 750	USD 9,000	USD 375	USD 4,500
	10 TB	Average daily QPS is up to 120.	USD 1,500	USD 18,000	USD 750	USD 9,000
	20 TB	Average daily QPS is up to 260.	USD 3,000	USD 36,000	USD 1,500	USD 18,000
	50 TB	Average daily QPS is up to 600.	USD 7,500	USD 90,000	USD 3,000	USD 36,000
	100 TB	Average daily QPS is up to 1,200.	USD 15,000	USD 180,000	USD 7,500	USD 90,000

Upgrade storage capacity

If you have no log storage left, a notification appears to remind you to expand the storage size. You can expand the log storage size at any time.

Notice:
If log storage is full, WAF stops writing new log entries to the exclusive logstore in Log Service. A log entry stored in the logstore is deleted based on the specified period. If the WAF Log Service instance expires and you do not renew it within seven days, all log entries in the logstore are deleted.

Validity

The validity of the WAF Log Service instance is based on your WAF subscription.

- Buy: When you buy a WAF subscription and enable Log Service, the price of Log Service is calculated based on the validity of the subscription.

- Upgrade: When you enable Log Service by upgrading an existing WAF subscription, the price of Log Service is calculated based on the remaining validity of the existing WAF instance. The remaining validity is accurate to minutes.

Service expiration

If your WAF instance expires, WAF Log Service expires at the same time.

- When the service expires, WAF stops writing log entries to the exclusive logstore in Log Service.
- The log entries recorded by WAF Log Service are retained within seven days after the service expires. If you renew the service within seven days after the service expires, you can continue to use WAF Log Service. Otherwise, all stored WAF log entries are deleted.

4.7.3 Activate WAF Log Service

After purchasing a Web Application Firewall instance, you can activate the real-time log query and analysis service for your websites on the App Management page in the console.

Scope

With WAF Log Service, you can collect multiple log entries in real time from your websites that are protected by WAF. You can also perform real-time log query and analysis and display results in dashboards. WAF Log Service fully meets the business protection needs and operational requirements of your websites. You can select the log storage period and the log storage size as needed when enabling WAF Log Service.

 Note:
At the moment, WAF Log Service is only available to WAF subscription instances (Pro, Business, or Enterprise edition).

Enable WAF Log Service

1. Log on to the *Web Application Firewall console*.
2. Choose App Market > App Management, and select the region where your WAF instance is located.
3. Click Upgrade in Real-time Log Query and Analysis Service.

4. On the page that is displayed, enable Log Service, select the log storage period and the log storage size, and then click Buy Now.

   Note:
   For more information about the billing of WAF Log Service, see *WAF Log Service Billing methods*.

5. Return to the WAF console and choose App Market > App Management, and then click Authorize in Real-time Log Query and Analysis Service.
6. Click Agree to authorize WAF to write log entries to your exclusive logstore.

   WAF Log Service is then enabled and authorized.

7. Return to the WAF console and choose App Market > App Management and then, click Configure in Real-time Log Query and Analysis Service.
8. On the Log Service page, select the domain name of your website that is protected by WAF, and turn on the Status switch on the right to enable WAF Log Service.

   Log Service collects all web log recorded by WAF in real time. These log entries can be queried and analyzed in real time.

4.7.4 Log collection

You can enable the Web Application Firewall (WAF) log collection feature for a specified domain in the WAF console.

Prerequisites

- Buy a WAF instance and protect the *domain using WAF*.
- Enable Log Service.

Context

Log Service collects log entries that record visits to and attacks on websites that are protected by Alibaba Cloud WAF, and supports real-time log query and analysis. The query results are displayed in dashboards. You can timely perform analytical investigation on visits to and attacks on your websites and help security engineers to develop protection strategies.

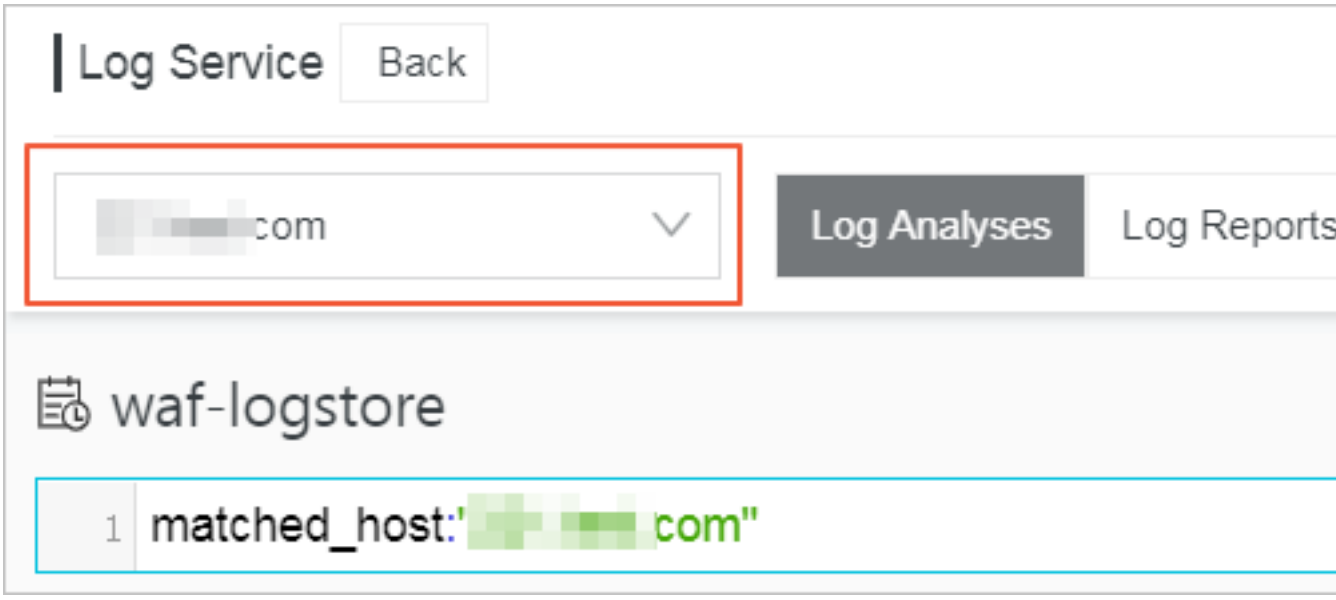
Procedure

1. Log on to the *Web Application Firewall console*.

2. Choose App Market > App Management, and click Real-time Log Query and Analysis Service.

   Note:
   If you are configuring the WAF log collection feature for the first time, click Authorize and follow the instructions on the authorization page to authorize WAF to write all log entries to your exclusive logstore.

3. Select the domain and turn on the Status switch on the right to enable the log collection feature.



   The WAF log collection feature has now been enabled for the domain. Log Service automatically creates an exclusive logstore for your account. WAF automatically

writes log entries to the exclusive logstore. The following *Default configuration* table describes the default configuration of the exclusive logstore.

Table 4-3: Default configuration

Default configuration item	Description
Project	A project is created by default. The project name format is determined by the region of your WAF instance. · If the WAF instance is created in Mainland China, the project name is `waf-project-Your Alibaba Cloud account ID-cn-hangzhou`. · If the WAF instance is created in other regions, the project name is `waf-project-Your Alibaba Cloud account ID-ap-southeast-1`.
Logstore	A logstore `waf - logstore` is created by default. All log entries collected by the WAF log collection feature are saved in this logstore.
Region	· If the WAF instance is created in Mainland China , the project is saved in the Hangzhou region by default. · If the WAF instance is created in other regions, the project is saved in the Singapore region by default.
Shard	Two shards are created by default with the *Automatic shard splitting* feature enabled.

Default configuration item	Description
Dashboard	Three dashboards are created: · Access Center · Operation Center · Security Center For more information about dashboards, see *WAF Log Service—Log Reports*.

Limits and instructions

- Other data cannot be written to the exclusive logstore.

  Log entries generated by WAF are stored in the exclusive logstore. You cannot write other data to this logstore by using API, SDK or other methods.

  > Note:
  > The exclusive logstore has no special limits in query, statistics, alerts, streaming consumption and other functions.

- Basic configurations, such as the storage period of log entries, cannot be modified.
- The exclusive logstore is not billed.

  To use the exclusive logstore, you must enable Log Service for your account. The exclusive logstore is not billed.

  > Note:
  > When your Log Service is overdue, the WAF log collection feature is suspended until you pay the bills in a timely manner.

- Do not delete or modify the configurations of the project, logstore, index, and dashboards, which are created by Log Service by default. Log Service updates the WAF log query and analysis service on an irregular basis. The index of the exclusive logstore and the default reports are also updated automatically.
- If you want to use the WAF log query and analysis service with a RAM user, you must grant the required Log Service permissions to the RAM user. For more information about how to grant permissions, see*Grant log query and analysis permissions to a RAM user*.

4.7.5 Log Analyses

The Real-time Log Query and Analysis Service page in the Web Application Firewall (WAF) console is integrated with the Log Analyses feature and the Log reports feature. After *enabling the WAF log collection feature* for a domain, you can perform real-time query and analysis, view or edit dashboards, and set up monitoring and alarms in the Real-time Log Query and Analysis Service page.

Procedure

1. Log on to the *Web Application Firewall console*, and choose App Market > App Management.
2. Click on the Real-time Log Query and Analysis Service area to open the Log Service page.
3. Select the domain and check that the Status switch on the right is turned on.
4. Click Log Analyses.

   The current page is integrated with the Querying and analyzing page. A query statement is automatically inserted. For example, `matched_ho st : " www . aliyun . com "` is used to query all log entries that is related to the domain in the statement.
5. Enter a query and analysis statement, select a log time range, and then click Search & Analysis.

More operations

The following operations are available in the Log Analyses page.

- Customize query and analysis

  Log Service provides rich query and analysis syntax for querying log entries in a variety of complex scenarios. For more information, see the *Custom query and analysis* in this topic.

- View the distribution of log entries by time period

  Under the query box, you can view the distribution of log entries that are filtered by time period and query statement. A histogram is used to indicate the distribution, where the horizontal axis indicates the time period, and the vertical

axis indicates the number of log entries. The total number of the log entries in the query results is also displayed.

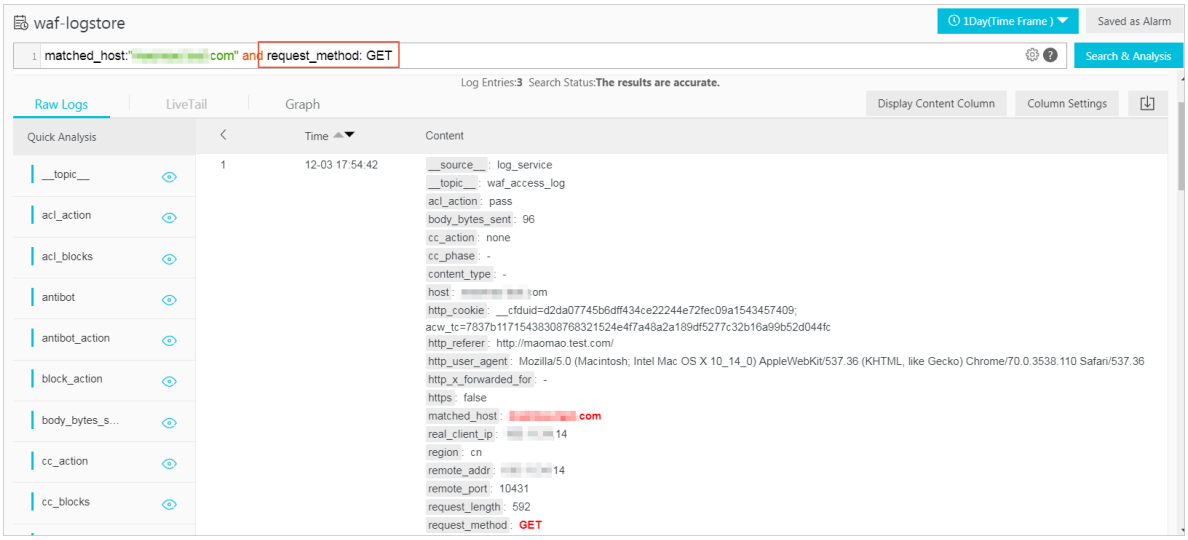
> **Note:**
> You can hold down the left mouse button and drag the histogram to select a shorter period. The `time picker` automatically updates the time period, and the query results are also updated based on the updated time period.

- View raw log entries

  In the Raw Logs tab, each log entry is detailed in a single page, which includes the time when the log entry is generated, the content, and the properties in the log entry. You can click Display Content Column to configure the display mode (Full Line or New Line) for long strings in the Content column. You can click Column Settings to display specific fields, or click the Download Log button to download the query results.

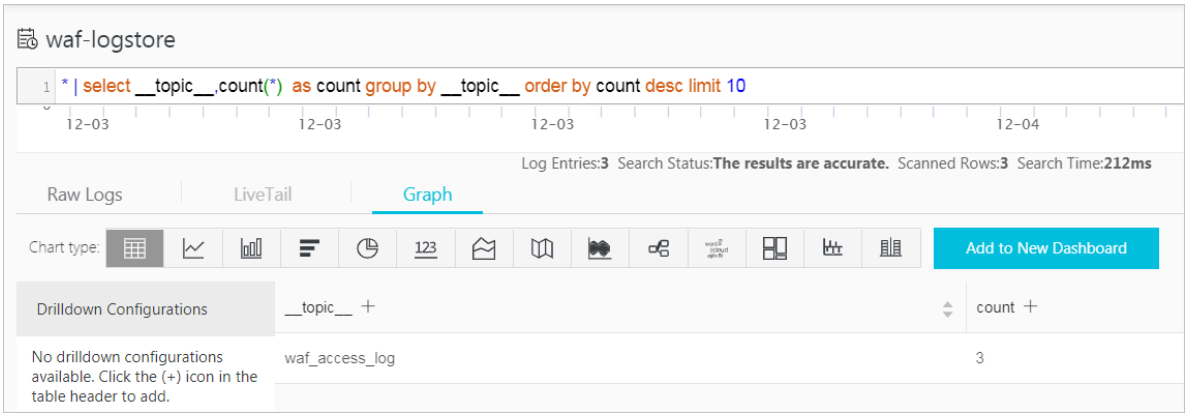
  Additionally, you can click a value or a property name to add a query criterion to the query box. For example, if you click the value `GET` in the `request_me thod : GET` filed, the query statement in the query box is updated to:

```
< The original query statement > and request_me thod :
 GET
```



- View analysis graphs

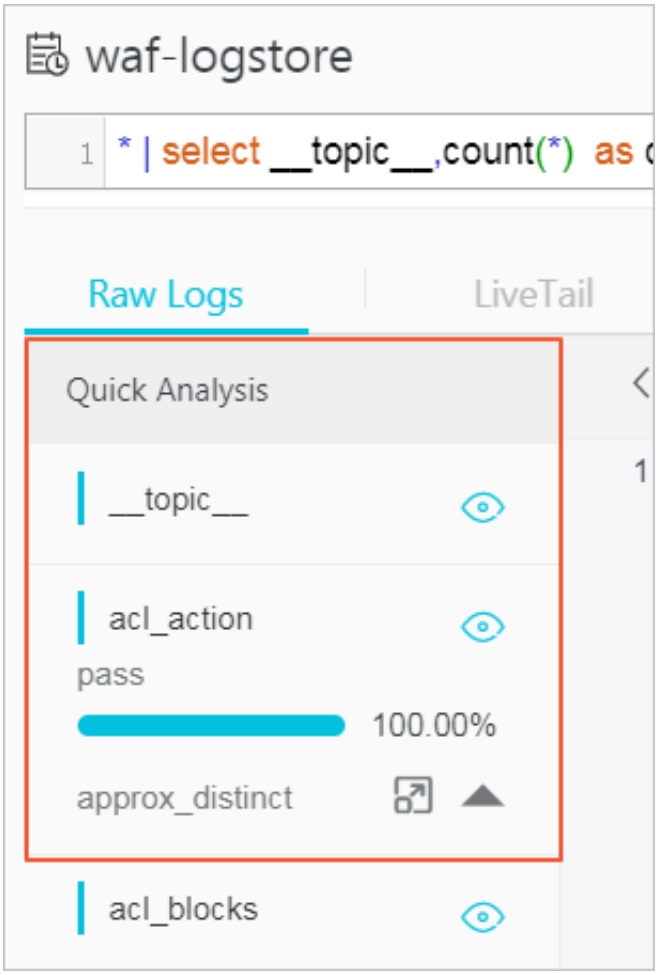
  Log Service enables you to display the analysis results in graphs. You can select the graph type as needed in the Graph tab. For more information, see *Analysis graph*.



- Perform quick analysis

  The Quick Analysis feature in the Raw Logs tab provides you with an one-click interactive experience, which gives you a quick access to the distribution of log entries by a single property within a specified time period. This feature can reduce

the time used for indexing key data. For more information, see *Quick analysis* in the following section.



Customize query and analysis

The log query statement consists of the query (Search) and the analysis (Analytics). These two parts are divided by a vertical bar (|):

```
$ Search  | $ Analytics
```

Type	Description
Query (Search)	A keyword, a fuzzy string, a numerical value, a range, or other criteria can be used in the query criteria. A combined condition can also be used. If the statement is empty or only contains an asterisk (*), all log entries are displayed.
Analysis (Analytics)	Performs computing and statistics to the query results or all log entries.

 Note:

Both the query part and the analysis part are optional.

- When the query part is empty, all log entries within the time period are displayed. Then, the query results are used for statistics.
- When the analysis part is empty, only the query results are returned without statistics.

Query syntax

The query syntax of Log Service supports full-text index and field search. You can enable the New Line display mode, syntax highlighting, and other features in the query box.

- Full text index

  You can enter keywords without specifying properties to perform the query by using the full-text index. You can enter the keyword with double quotation marks ("") surrounded to query log entries that contain the keyword. You can also add a space or `and` to separate keywords.

  Examples

  - Multiple-keywords query

    The following statements can be used to query all log entries that contain `www . aliyun . com` and `error` .

    `www . aliyun . com error` or `www . aliyun . com and error` .

  - Criteria query

    The following statement can be used to search for all log entries that contain `www . aliyun . com` , `error` or `404` .

    ```
    www . aliyun . com and ( error or 404 )
    ```

  - Prefix query

    The following statement can be used to query all log entries that contain `www . aliyun . com` and start with `failed_` .

    ```
    www . aliyun . com and failed_ *
    ```

     Note:

An asterisk (*) can be added as a suffix, but it cannot be added as a prefix. For example, the statement cannot be `* _error` .

- Field search

  You can perform a more accurate query based on specified fields.

  The field search supports comparison queries for fields of numeric type. The format is `field name : value` or `field name >= value` . Moreover, you can perform combination queries using `and` or `or` , which can be used in combination with the full text index.

   Note:
  The log entries that record access, operation, and attack on the domain name in WAF Log Service can also be queried by fields. For more information about the meaning, type, format, and other information of the fields, see *Fields in the WAF log entries*.

  Examples

  - Multiple-fields query

    The following statement can be used to query all log entries that record the HTTP flood attack on the `www . aliyun . com` domain and are intercepted by WAF .

    ```
    matched_ho st : www . aliyun . com and cc_blocks : 1
    ```

    If you want to query all log entries that record access from a specific client whose IP address is `1 . 2 . 3 . 4` to `www . aliyun . com` , and access is blocked by the 404 error, you can use the following statement.

    ```
    real_clien t_ip : 1 . 2 . 3 . 4 and matched_ho st : www .
    aliyun . com and status : 404
    ```

     Note:

In this example, the `matched_ho st`, `cc_blocks`, `real_clien t_ip`, and `status` fields are the fields defined in the WAF log.

- Numeric fields query

  The following statement can be used to query all log entries where the response time exceeds five seconds.

  ```
  request_ti me_msec > 5000
  ```

  Range query is also supported. For example, you can query all log entries where the response time exceeds five seconds and is no more than 10 seconds.

  ```
  request_ti me_msec in ( 5000 10000 ]
  ```

   Note:

  The following query statement has the same function.

  ```
  request_ti me_msec > 5000 and request_ti me_msec <=
  10000
  ```

- Field existence query

  You can perform a query based on the existence of a field.

  ■ The following statement can be used to search for all log entries where the `ua_browser` field exists.

  ```
  ua_browser : *
  ```

  ■ The following statement can be used to search for all log entries where the `ua_browser` field does not exist.

  ```
  not ua_browser : *
  ```

For more information about the query syntax that is supported by Log Service, see*Index and query*.

Syntax for analysis

You can use the SQL/92 syntax for log analysis and statistics.

For more information about the syntax and functions supported by Log Service, see*Syntax description*.

 Note:

- The `from table name` part that follows the SQL standard syntax can be omitted from the analysis statement. In WAF Log Service, `from log` can be omitted.
- The first 100 results are returned by default, and you can modify the number of results that are returned by using the *LIMIT syntax*.

Examples of query and analysis

Time-based log query and analysis

Each WAF log entry has a `time` field, which is used to represent the time when the log entry is generated. The format of the value in this field is `< year >-< month >-< day > T < hour >:< minute >:< second >+< time zone >`. For example, `2018 - 05 - 31T20 : 11 : 58 + 08 : 00` is 20:11:58 `UTC + 8` (Beijing Time), May 15, 2018.

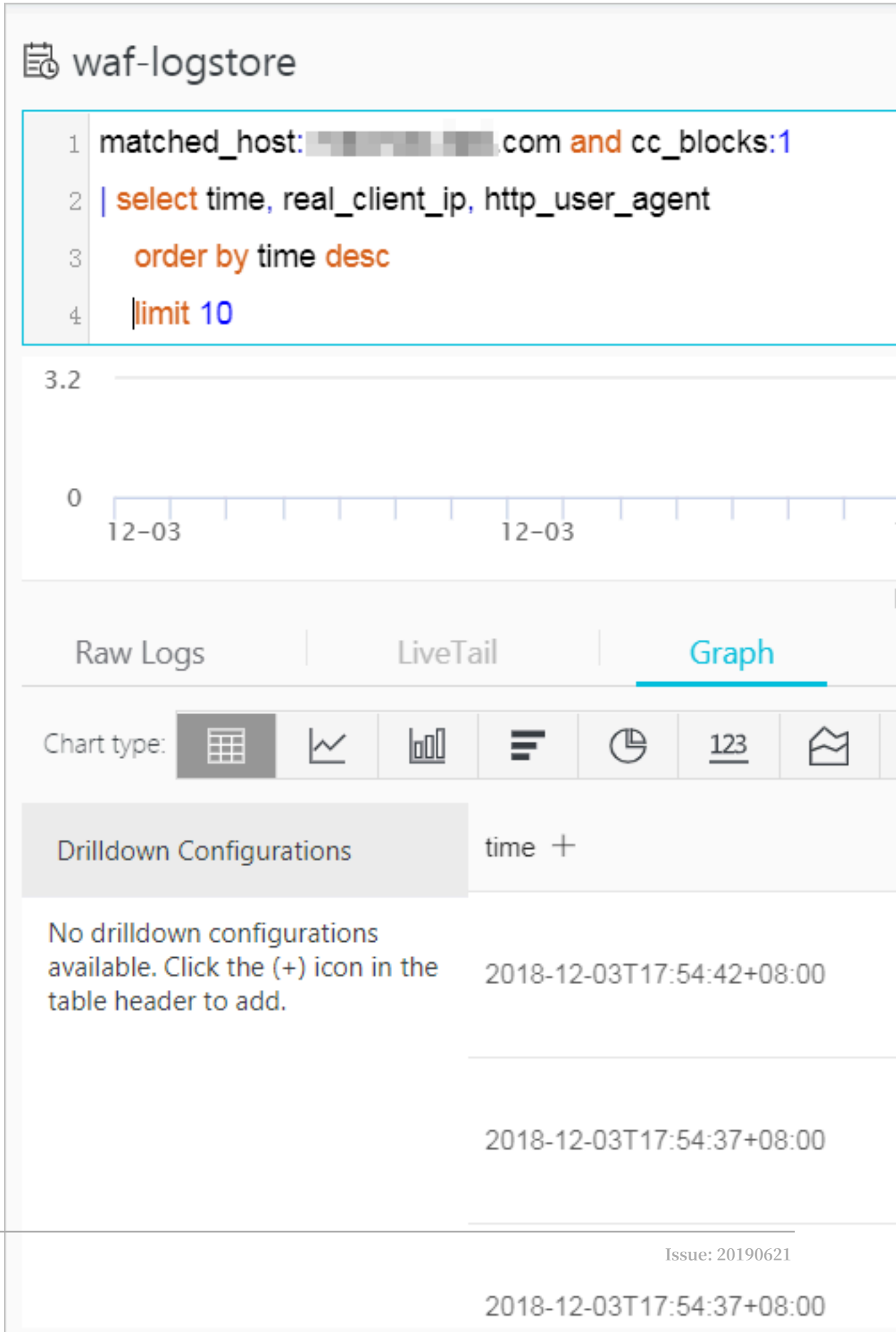
In addition, each log entry has a built-in field `__time__`, which is also used to indicate the time when the log entry is generated. This field is used for calculation when performing statistics. The format of this field is a *Unix timestamp*, and the value of this field indicates the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), January 1, 1970. Therefore, if you want to display a calculated result, you must convert the format first.

- Select and display the time

  You can query the log based on the `time` field. For example, you can search for the last 10 log entries that record the HTTP flood attacks on `www . aliyun . com` and are intercepted by WAF. Then, you can display the time field, the source IP field, and the client field.

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| select time , real_clien t_ip , http_user_ agent
  order by time desc
```

```
limit   10
```


waf-logstore
1 matched_host: com and cc_blocks:1
2 | select time, real_client_ip, http_user_agent
3 order by time desc
4 limit 10
3.2
0
12-03
12-03
Raw Logs
LiveTail
Graph
Chart type:
123
Drilldown Configurations
time +
No drilldown configurations available. Click the (+) icon in the table header to add.
2018-12-03T17:54:42+08:00
2018-12-03T17:54:37+08:00
2018-12-03T17:54:37+08:00

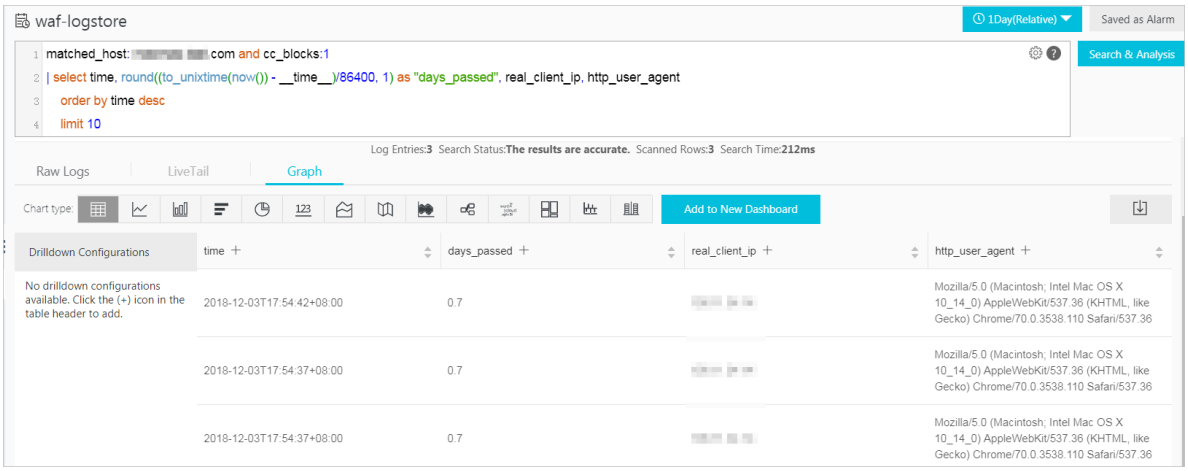
- Calculate using time.

  You can use the `__time__` field to calculate using time. For example, you can calculate the number of days that have elapsed since the domain suffered a HTTP flood attack.

```
matched_ho  st :  www . aliyun . com  and  cc_blocks :  1
round (( to_unixtim  e ( now ()) -  __time__ )/ 86400 ,  1 )  as  "
days_passe  d ",  real_clien  t_ip ,  http_user_  agent
      order  by  time  desc
      limit  10
```

  Note:

  In this example, `round (( to_unixtim  e ( now ()) -  __time__ )/ 86400 ,  1 )` is used to calculate the number of days that have elapsed since the domain had a HTTP flood attack. First, use `now ()` to get the current time, and convert the current time into a Unix timestamp using `to_unixtim  e` . Then, subtract the converted time with the value of the built-in field `__time__` to get the number of seconds that have elapsed. Finally, divide it by `86400` (the total number of seconds in a day) and apply the `round ( data ,  1 )` function to keep one decimal place. The result is the number of days that have elapsed since each attack log entry is generated.



- Perform group statistics based on a specific time

  You can query the log based on the trend of HTTP flood attacks on the domain within a specified time period.

```
 matched_ho  st :  www . aliyun . com  and  cc_blocks :  1
|  select  date_trunc (' day ',  __time__ )  as  dt ,  count ( 1
 )  as  PV
      group  by  dt
```

```
        order   by   dt
```

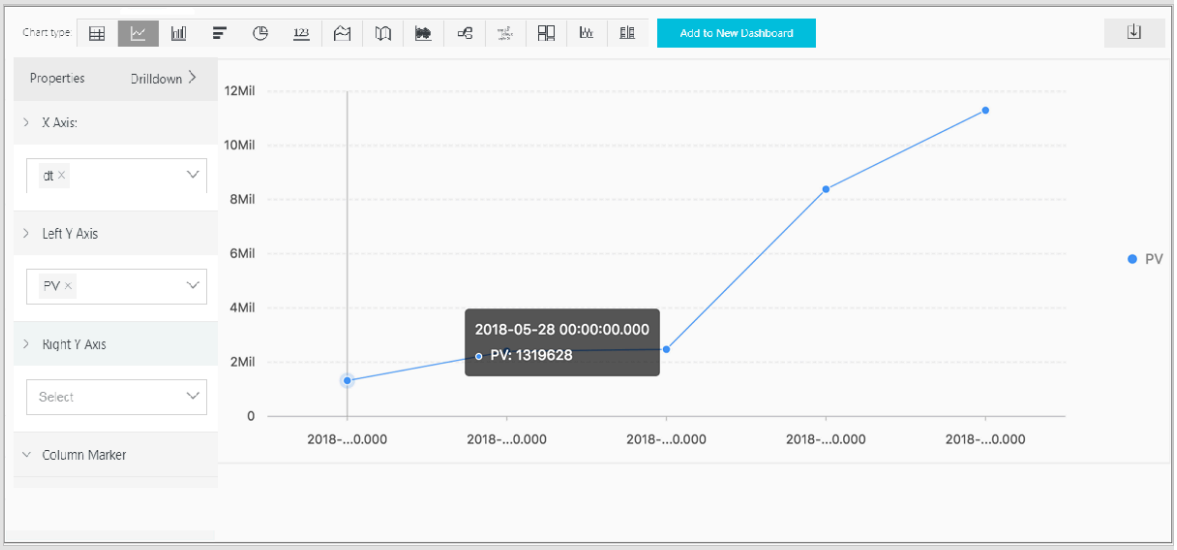
Note:

In this example, the built-in field `__time__` is used by the `date_trunc ('day', ..)` function to align the time of the entries by day. Each log entry is assigned to a group based on the day when the log entry is generated. The total number of log entries in each group is counted using count(1). Then, these entries are ordered by the group. You can use other values for the first parameter of the `date_trunc` function to group the log entries based on other time units, such as `second`, `minute`, `hour`, `week`, `month`, and `year`. For more information about this function, see *Date and time functions*.

dt	PV
2018-12-03 00:00:00.000	3

Note:

You can also display the results with a line chart.



- Perform group statistics based on time.

  If you want to analyze the log based on time using more flexible groupings, complex calculations are required. For example, you can query the log based on the trend of HTTP flood attacks on the domain within every five minutes.

```
 matched_ho  st :  www . aliyun . com   and   cc_blocks :  1
|  select   from_unixt  ime ( __time__  -  __time__ %  300 )  as
 dt ,
         count ( 1 )  as   PV
```

```
group by dt
order by dt
limit 1000
```

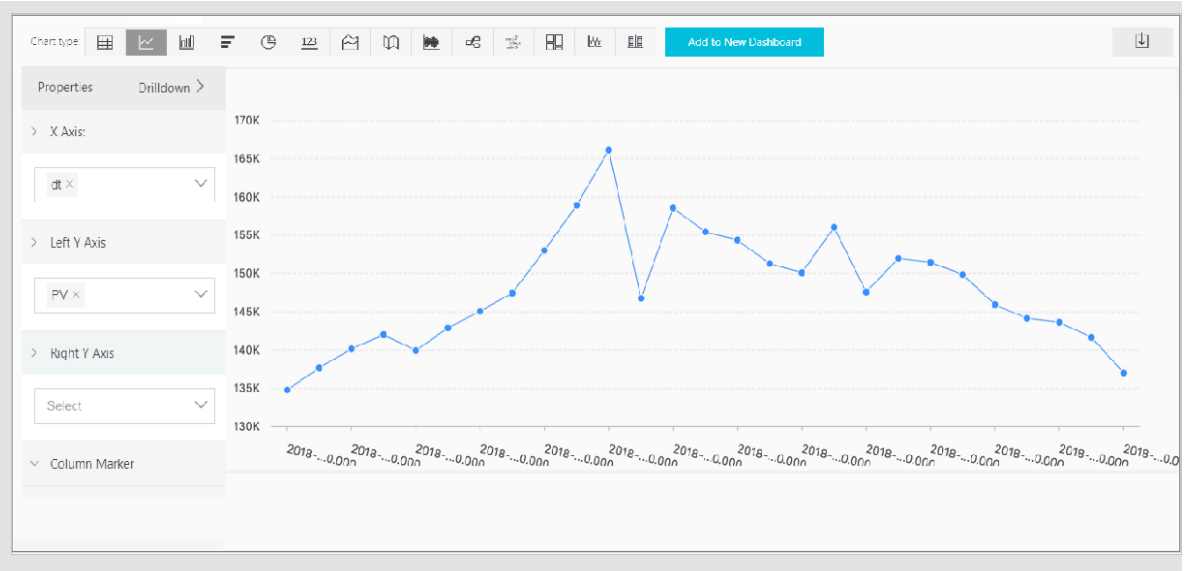
Note:

In this example, the built-in field is used for aligning the time by using the formula `__time__ - __time__ % 300`, and the `from_unixtime` function converts the format of the result. Then, each entry is assigned to a group that indicates a time period of five minutes (300 seconds), and the total number of log entries in each group is counted using count(1). Finally, the query results are ordered by group and the first 1,000 results are returned, which include the log entries that are generated within 83 hours before the specified time period.

dt	PV
2018-05-31 21:30:00.000	134795
2018-05-31 21:35:00.000	137691
2018-05-31 21:40:00.000	140171
2018-05-31 21:45:00.000	142037
2018-05-31 21:50:00.000	139958
2018-05-31 21:55:00.000	142906
2018-05-31 22:00:00.000	145093
2018-05-31 22:05:00.000	147474

Note:

You can also display the results with a line graph.



The `date_parse` and `date_forma t` functions are used to convert the time format. For more information about the functions that can be used to parse the time field, see *Date and time functions*.

Client IP address-based log query and analysis

The WAF log contains the field `real_clien t_ip`, which reflects the real client IP address. In cases where the user accesses your website through a proxy server, or the IP address in the request header is wrong, you cannot get the real IP address of the user. However, the `remote_add r` field forms a direct connection to the client, which can be used to get the real IP address.

- Classify attackers by country

  You can query the log based on the distribution of HTTP flood attackers by country.

```
 matched_ho  st :  www . aliyun . com  and  cc_blocks :  1
|  SELECT  ip_to_coun  try ( if ( real_clien  t_ip ='-',
 remote_add  r ,  real_clien  t_ip ))  as  country ,
          count ( 1 )  as  " number  of  attacks "
          group  by  country
```



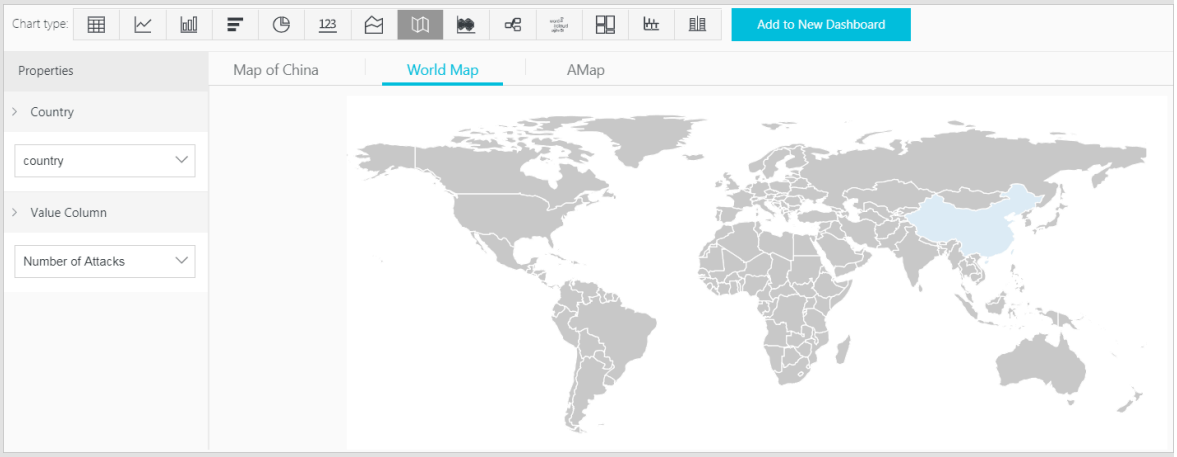
Note:

In this example, the function `if ( condition , option1 , option2 )` returns the real client IP address. If `real_clien t_ip` is `-`, the function returns the value of `remote_add r`. Otherwise, the function returns

`real_clien t_ip` . Then, use the `ip_to_coun try` to get the country information from the IP address of the client.

Note:

You can also display the results with a world map.



- Distribution of visitors by province

  If you want to get the distribution of visitors by province, you can use the `ip_to_prov ince` function to get the province information from the IP addresses.

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_prov ince ( if ( real_clien t_ip ='-',
remote_add r , real_clien t_ip )) as province ,
         count ( 1 ) as " number of attacks "
         group by province
```

Note:

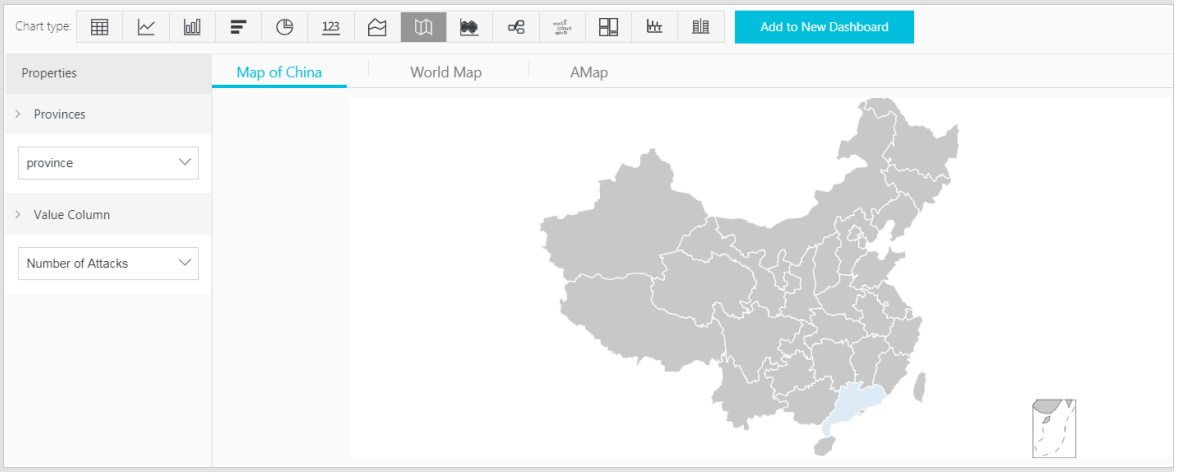
In this example, the `ip_to_prov ince` function is used to get the country information from the real IP address of the client. If the IP address is not in the Mainland of China, the function returns the province or state of the IP address

in the country field. However, if you choose to display the results with a map of China, IP addresses that are not in the Mainland of China are not displayed.

province	Number of Attacks
广东省	3

Note:

You can also display the results with a map of China.



- Heat map that indicates the distribution of attackers

  You can use the `ip_to_geo` function to get the geographic information (the latitude and the longitude) from the real IP addresses of the clients. This information can be used to generate a heat map to indicate the density of attacks.

```
matched_ho st : www . aliyun . com and cc_blocks : 1
| SELECT ip_to_geo ( if ( real_clien t_ip ='-', remote_add r
, real_clien t_ip )) as geo ,
        count ( 1 ) as " number of attacks "
        group by geo
        limit 10000
```

Note:

In this example, the `ip_to_geo` function is use to get the latitude and the longitude from the real IP addresses of the clients. The first 10,000 results are returned.

Select Amap and click Show Heat Map.

The `ip_to_prov ider` function can be used to get the IP provider name, and the `ip_to_doma in` function can be used to determine whether the IP is a public IP or

a private IP. For more information about the functions that can be used to resolve IP addresses, see *IP functions*.

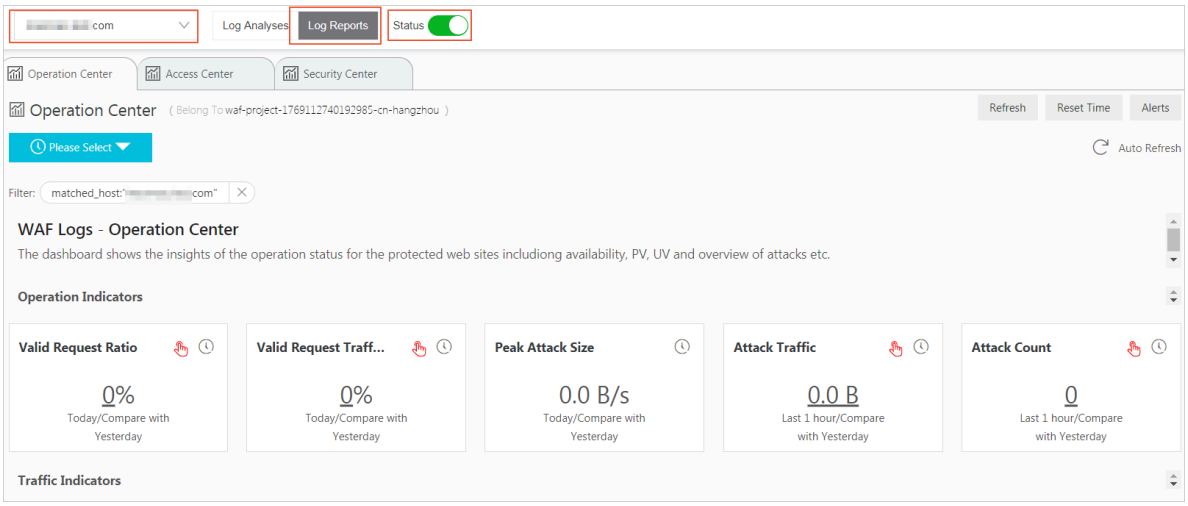
4.7.6 Log Reports

The Log Reports page is integrated with the Dashboard page of Log Service. On this page, you can view default dashboards. You can filter business and security data about your website by modifying the time range or adding filters.

View reports

1. Log on to the *Web Application Firewall console*, and choose App Market > App Management.
2. Click the Real-time Log Query and Analysis Service area to open the Log Service page.
3. [DO NOT TRANSLATE]
4. Select a domain and check that the Status switch on the right is turned on.
5. Click Log Reports.

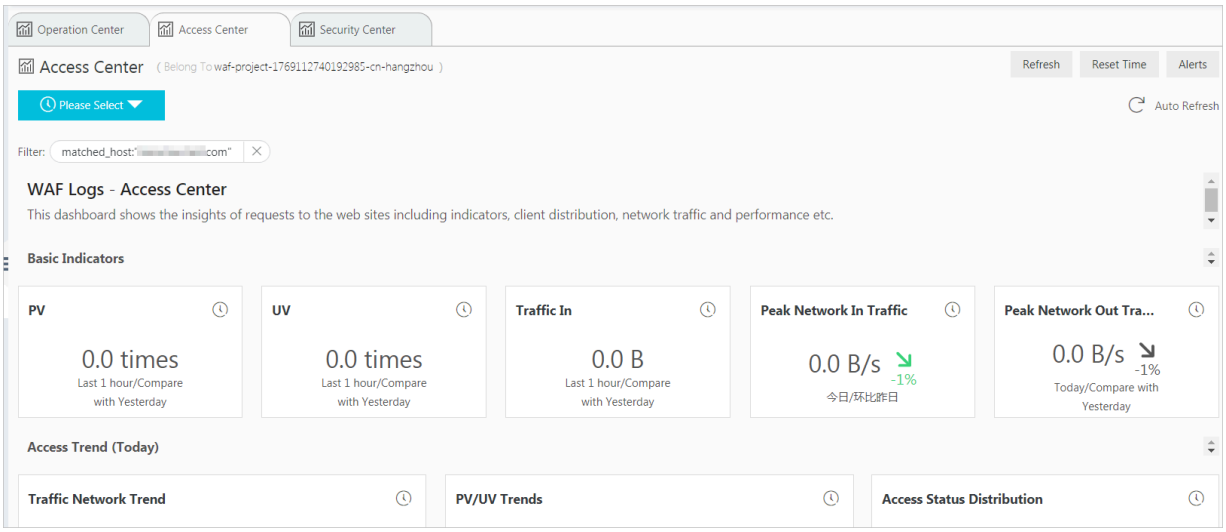
   The page that appears is integrated with the Dashboard page of Log Service. A filter is automatically added to display all log entries that are recorded for the domain you selected. In this example, the filter is `matched_ho st : www . aliyun . com` .



After you enable the WAF log collection feature, Log Service creates three dashboards by default: the Operation Center, Access Center, and Security Center.

> **Note:**
> For more information about the default dashboards, see *Default dashboards*.

Dashboard	Description
Operation Center	Displays operation details such as the proportion of valid requests and the statistics of attacks, traffic details such as the peak of both inbound and outbound throughput and the number of requests received, operation trends, attack overview, and other information.
Access Center	Displays basic access details such as the number of page views (PV) and the number of unique visitors (UV), the access trend, the distribution of visitors, and other information.
Security Center	Displays basic index information of attacks, attack types, attack trend, attacker distribution, and other information.



Note:

Dashboards displays various reports using the layout that is predefined in WAF Log Service. The following table describes the graph types supported for reports. For more information about the graph types supported by Log Service, see *Graph description*.

Type	Description
Number	Graphs of this type display important metrics, such as the valid request ratio and the peak of attacks.

Type	Description
Line chart and area chart	Graphs of these types display the trend of important metrics within a specified time period, such as the trend of inbound throughput and the trend of attack interceptions.
Map	Graphs of this type display the geographical distribution of visitors and attackers, for example, by country. Heat maps are also supported to illustrate the distribution of attackers.
Pie chart	Graphs of this type display a distribution, such as the distribution of attackers and the distribution of client types.
Table	Graphs of this type display a table that contains information, such as information of attackers.
Map	Graphs of this type display the geographical distribution of data.

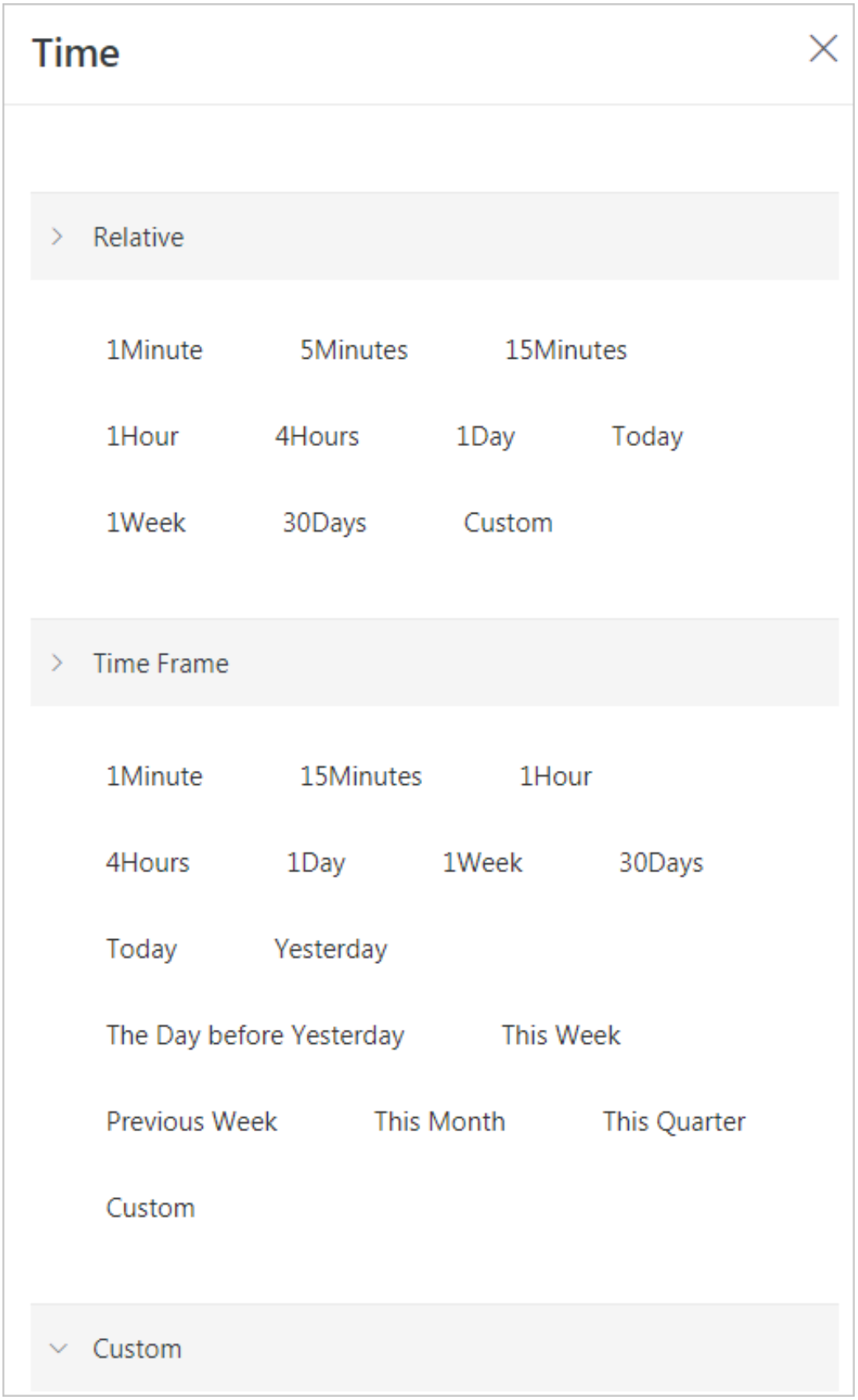
Time selector

The data in all graphs on the dashboard page are generated based on different time ranges. If you want to unify the time ranges, configure the time selector.

1. On the Log Reports page, click Please Select and
2. select a time range in the pane that appears. You can select a relative time, a time frame, or customize a time range.

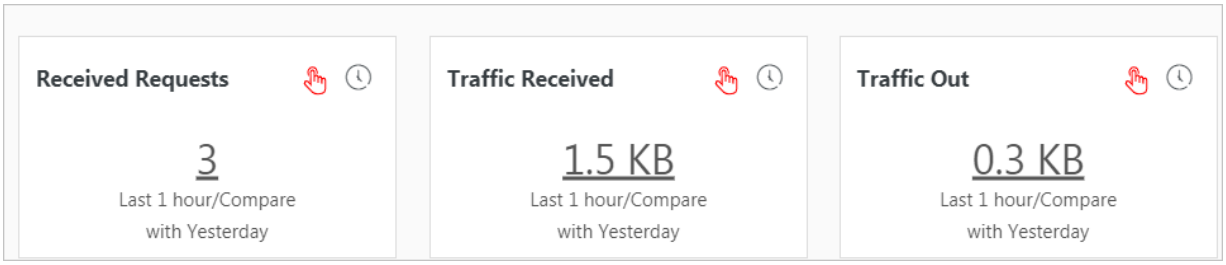
Note:

- After you set a time range, the time range is applied to all reports.
- If you set a time range, a temporary view is generated on the current page. When you view reports next time, the default time range is used.
- To change the time range for a single report in the dashboard, click  in the upper-right corner.


Time
Relative
1Minute
5Minutes
15Minutes
1Hour
4Hours
1Day
Today
1Week
30Days
Custom
Time Frame
1Minute
15Minutes
1Hour
4Hours
1Day
1Week
30Days
Today
Yesterday
The Day before Yesterday
This Week
Previous Week
This Month
This Quarter
Custom
Custom

Data drilldown

The drilldown operation is enabled for some graphs on the dashboard page, which provides you a quick access to the detailed data.



The drilldown operation is available for graphs marked with a  icon in the upper-right corner. You can click a number with an underline to view the detailed underlying data. For example, to quickly find the domains that are attacked and the number of attacks, click the number in the Attacked Hosts graph of the Security Center report.

Note:
Alternatively, switch to the Raw Log tab to find the relevant log entries.

Description of values in default dashboards

- Operation Center: Displays operation details such as the proportion of valid requests and the statistics of attacks, traffic details such as the peak of both inbound and outbound throughput and the number of requests received, the operation trend, the attack overview, and other information.

Graph	Type	Default time range	Description	Example
Valid Request Ratio	Single value	Today (time frame)	Displays the percentage of valid requests in all requests. A valid request is a request that is neither an attack nor a request that is blocked by a 400 error.	95%

Graph	Type	Default time range	Description	Example
Valid Request Traffic Ratio	Single value	Today (time frame)	Displays the percentage of the traffic generated by valid requests in the traffic generated by all requests.	95%
Peak Attack Size	Single value	Today (time frame)	Displays the peak of attack traffic, which is measured in Bps.	100 B/s
Attack Traffic	Single value	1 hour (relative)	Displays the total attack traffic, which is measured in B.	30 B
Attack Count	Single value	1 hour (relative)	The total number of attacks.	100
Peak Network In	Single value	Today (time frame)	Displays the peak inbound throughput , which is measured in KB/s.	100 KB/s
Peak Network Out	Single value	Today (time frame)	Displays the peak outbound throughput, which is measured in KB/s.	100 KB/s
Received Requests	Single value	1 hour (relative)	Displays the total number of valid requests.	7,800
Received traffic	Single value	1 hour (relative)	Displays the total inbound traffic that is generated by valid requests, which is measured in MB.	1.4 MB
Traffic Out	Single value	1 hour (relative)	Displays the total outbound traffic that is generated by valid requests, which is measured in MB.	3.8 MB

Graph	Type	Default time range	Description	Example
Network Traffic In And Attack	Area chart	Today (time frame)	Displays the trends of throughput generated by valid requests and attacks, which is measured in Kbit/s.	-
Request And Interception	Line chart	Today (time frame)	Displays the trends of valid requests and requests that are intercepted, which is measure in Kbit/h.	-
Access Status Distribution	Flow chart	Today (time frame)	Displays the trends of requests with different status codes (404, 304, 200, and other status codes), which is measured in Kbit/h.	-
Attack Source (World)	World map	1 hour (relative)	Displays the distribution of attackers by country.	-
Attack Source (China)	Map of China	1 Hour (Relative)	Displays the distribution of attackers in China by province.	-
Attack Type	Pie chart	1 hour (relative)	Displays the distribution of attacks by attack type.	-
Attacked Hosts	Tree map	1 hour (relative)	Displays the domains that are attacked and the number of attacks.	-

- Access center: Displays basic access details such as the number of PV and the number of UV, the access trend, the distribution of visitors, and other information.

Graph	Type	Default time range	Description	Example
PV	Single value	1 hour (relative)	Displays the total number of PV.	100,000
UV	Single value	1 hour (relative)	Displays the total number of UV.	100
Traffic In	Single value	1 hour (relative)	Displays the total inbound traffic, which is measured in MB.	300 MB
Peak Network In Traffic	Single value	Today (time frame)	Displays the peak inbound throughput , which is measured in KB/s.	0.5 KB/s
Peak Network Out Traffic	Single value	Today (time frame)	Displays the peak outbound throughput, which is measured in KB/s.	1.3 KB/s
Traffic Network Trend	Area chart	Today (time frame)	Displays the trends of inbound and outbound throughput, which are measured in KB/ s.	-
PV/UV Trends	Line chart	Today (time frame)	Displays the trends of PV and UV, which is measured in Kbit/ h.	-
Access Status Distribution	Flow chart	Today (time frame)	Displays the trends of requests with different status codes (404, 304, 200 , and other status code), which is measured in Kbit/h.	-

Graph	Type	Default time range	Description	Example
Access Source	World map	1 hour (relative)	Displays the distribution of attackers by country.	-
Traffic In Source (World)	World map	1 hour (relative)	Displays the distribution (by country) of inbound traffic from requests.	-
Traffic In Source (China)	Map of China	1 hour (relative)	Displays the distribution (by province) of inbound traffic from requests in China.	-
Access Heatmap	Amap	1 hour (relative)	Displays the heat map that indicates the source distribution of requests by geographical position.	-
Network Provider Source	Pie chart	1 hour (relative)	Displays the source distribution of requests by Internet service provider that provides network for the source, such as China Telecom , China Unicom, China Mobile, and universities.	-
Referer	Table	1 hour (relative)	Displays the first 100 referer URLs which the hosts are most often redirected from, and displays the information of hosts and redirection frequency.	-

Graph	Type	Default time range	Description	Example
Mobile Client Distribution	Pie chart	1 hour (relative)	Displays the distribution of requests from mobile clients, by client type.	-
PC Client Distribution	Pie chart	1 hour (relative)	Displays the distribution of requests from PC clients, by client type .	-
Request Content Type Distribution	Pie chart	1 hour (relative)	Displays the distribution of request sources by content type, such as HTML, form, JSON, and streaming data.	-
Accessed Sites	Tree map	1 Hour (Relative)	Displays the addresses of 30 domains that are visited most.	-
Top Clients	Table	1 hour (relative)	Displays the information of 100 clients that visit your domains most. The information includes the client IP address , the region and city, network information , the request method , inbound traffic , the number of incorrect accesses , the number of attacks, and other information.	-

Graph	Type	Default time range	Description	Example
URL With Slowest Response	Table	1 hour (relative)	Displays the information of 100 URLs that have the longest response times. The information includes the website address, the URL, the average response time, the number of accesses, and other informatio n.	-

- Security Center: Displays basic details of attacks, attack types, the attack trend, the distribution of attackers, and other information.

Chart	Type	Default time range	Description	Example
Peak Attack Size	Single value	1 hour (relative)	Displays the peak of the throughput when your website is suffering attacks, which is measured in Bps.	100 B/s
Attacked Hosts	Single value	Today (time frame)	Displays the number of domains that are attacked.	3
Source Country Of Attack	Single value	Today (time frame)	Displays the number of countries that are attack sources.	2
Attack Traffic	Single value	1 hour (relative)	Displays the total amount of traffic that is generated by attacks, which is measured in B.	1 B
Attacker UV	Single value	1 hour (relative)	Displays the number of unique clients that are attack sources.	40

Chart	Type	Default time range	Description	Example
Attack type distribution	Flow chart	Today (time frame)	Displays the distribution of attacks by attack type.	-
Intercepted Attack	Single value	1 hour (relative)	Displays the number of attacks that are intercepted by WAF.	100
HTTP flood attack Interception	Single value	1 hour (relative)	Displays the number of HTTP flood attacks that are intercepted by WAF.	10
Web Attack Interception	Single value	1 hour (relative)	Displays the number of Web applicatio n attacks that are intercepted by WAF.	80
Access Control Event	Single value	1 hour (relative)	Displays the number of requests that are intercepted by the HTTP ACL policies of WAF.	10
HTTP flood attack (World)	World map	1 hour (relative)	Displays the distribution of HTTP flood attackers by country.	-
HTTP flood attack (China)	China map	1 hour (relative)	Displays the distribution of HTTP flood attackers by province in China.	-
Web Attack (World)	World map	1 Hour (Relative)	Displays the distribution of Web application attacks by country.	-
Web Attack (China)	Map of China	1 hour (relative)	Displays the distribution of Web application attacks by province in China .	-

Chart	Type	Default time range	Description	Example
Access Control Attack (World)	World Map	1 hour (relative)	Displays the distribution by country of requests that are intercepte d by the HTTP ACL policies of WAF.	-
Access Control Attack (China)	Map of China	1 Hour (Relative)	Displays the distribution by province in China of requests that are intercepted by the HTTP ACL policy of WAF.	-
Attacked Hosts	Tree map	1 hour (relative)	Displays the websites that are attacked most.	-
HTTP flood attack Strategy Distribution	Pie chart	1 hour (relative)	Displays the distribution of security policies being activated for HTTP flood attacks.	-
Web Attack Type Distribution	Pie chart	1 hour (relative)	Displays the distribution of Web attacks by attack type.	-
Top Attackers	Table	1 hour (relative)	Displays IP addresses, provinces , and network providers of the first 100 clients that launch the recent attacks, and displays the number of attacks and the amount of traffic generated by these attacks.	-

Chart	Type	Default time range	Description	Example
Attacker Referer	Table	1 Hour (Relative)	Displays the information in referers of attack requests, which includes referer URLs, referer hosts , and the number of attacks.	-

4.7.7 Fields in the log entry

WAF keeps detailed log entries for your domains, including access requests and attack logs. Each log entry contains dozens of fields. You can perform query and analysis based on specific fields.

Field	Description	Example
__topic__	The topic of the log entry. The value of this field is waf_access _log, which cannot be changed.	waf_access_log
acl_action	The action generated by the WAF HTTP ACL policy to the request, such as pass, drop, and captcha. Note: If the value is null or -, it indicates that the action is pass.	pass
acl_blocks	Indicates whether the request is blocked by the HTTP ACL policy. · If the value is 1, the request is blocked. · If the value is not 1, the request is passed.	1

Field	Description	Example
antibot	The type of the Anti-Bot Service protection strategy that applies, which includes: · ratelimit: Frequency control · sdk: APP protection · intelligence: Algorithmic model · acl: HTTP ACL policy · blacklist: Blacklist	ratelimit
antibot_action	The action performed by the Anti-Bot Service protection strategy, which includes: · challenge: Verifying using an embedded JavaScript script · drop: Blocking · report: Logging the access event · captcha: Verifying using a slider captcha	challenge
block_action	The type of the WAF protection that is activated, which includes: · tmd: Protection against HTTP flood attacks · waf: Protection against Web application attacks · acl: HTTP ACL policy · geo: Blocking regions · antifraud: Risk control for data · antibot: Blocking Web crawlers	tmd
body_bytes_sent	The size of the body in the access request, which is measured in Bytes.	2
cc_action	Protection strategies against HTTP flood attacks, such as none, challenge, pass, close, captcha, wait, login, and n.	close

Field	Description	Example
cc_blocks	Indicates whether the request is blocked by the CC protection. · If the value is 1, the request is blocked. · If the value is not 1, the request is passed.	1
cc_phase	The CC protection strategy that is activated, which can be seccookie, server_ip_blacklist, static_whitelist, server_header_blacklist, server_cookie_blacklist, server_args_blacklist, or qps_overmax.	server_ip_blacklist
content_type	The content type of the access request.	application/x-www-form-urlencoded
host	The source website.	api.aliyun.com
http_cookie	The client-side cookie, which is included in the request header.	k1=v1;k2=v2
http_referer	The URL information of the request source, which is included in the request header. - indicates no URL information.	http://xyz.com
http_user_agent	The User Agent field in the request header, which contains information such as the client browser and the operating system.	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build/HUAWEIEDISON-AL10)
http_x_forwarded_for	The X-Forwarded-For (XFF) information in the request header, which identifies the original IP address of the client that connects to the Web server using a HTTP proxy or load balancing.	-

Field	Description	Example
https	Indicates whether the request is an HTTPS request. · true: the request is an HTTPS request. · false: the request is an HTTP request.	true
matched_host	The matched domain name (extensive domain name) that is protected by WAF. If no domain has been matched, the value is -.	*.aliyun.com
querystring	The query string in the request.	title=tm_content%3Darticle&pid=123
real_client_ip	The real IP address of the client. If the system cannot get the real IP address, the value is -.	1.2.3.4
region	The information of the region where the WAF instance is located.	cn
remote_addr	The IP address of the client that sends the access request.	1.2.3.4
remote_port	The port of the client that sends the access request.	3242
request_length	The size of the request, measured in Bytes.	123
request_method	The HTTP request method used in the access request.	GET
request_path	The relative path of the request. The query string is not included.	/news/search.php
request_time_msec	The request time, which is measured in microseconds.	44
request_traceid	The unique ID of the access request that is recorded by WAF.	7837b**********ea1f0
server_protocol	The response protocol and the version number of the origin server.	HTTP/1.1

Field	Description	Example
status	The status of the HTTP response to the client returned by WAF.	200
time	The time when the access request occurs.	2018-05-02T16:03:59+08:00
ua_browser	The information of the browser that sends the request.	ie9
ua_browser_family	The family of the browser that the sent the request.	internet explorer
ua_browser_type	The type of the browser that the sent the request.	web_browser
ua_browser_version	The version of the browser that sends the request.	9.0
ua_device_type	The type of the client device that sends the request.	computer
ua_os	The operating system used by the client that sends the request.	windows_7
ua_os_family	The family of the operating system used by the client.	windows
upstream_addr	A list of origin addresses, separated by commas. The format of an address is `IP : Port`.	1.2.3.4:443
upstream_ip	The origin IP address that corresponds to the access request. For example, if the origin server is an ECS instance , the value of this field is the IP address of the ECS instance.	1.2.3.4
upstream_response_time	The time that the origin site takes to respond to the WAF request, which is measured in seconds. "-" indicates the timeout of the request.	0.044

Field	Description	Example
upstream_status	The response status that WAF receives from the origin server. "-" indicates that no response is received. The reason can be the response timeout, or the request being blocked by WAF.	200
user_id	Alibaba Cloud account ID.	12345678
waf_action	The action from the Web attack protection policy. · If the value is block, the attack is blocked. · If the value is bypass or other values, the attack is ignored.	block
web_attack_type	The Web attack type such as xss, code_exec, webshell, sqli, lfilei, rfilei, and other.	xss
waf_rule_id	The ID of the WAF rule that is matched.	100

4.7.8 Advanced settings

If you click Advanced Settings on the page of WAF log query and analysis service, you will be redirected to the Log Service console. Then you can set advanced features for Log Service. For example, you can set alarms and notifications, real-time log collection and consumption, shipping log data, or provide visual representations with other products.

Procedure

1. Log on to the *Web Application Firewall console*, choose App Market > App Management.
2. Click the Real-time Log Query and Analysis Service area to open the Log Service page.
3. Click Advanced Settings in the upper-right corner.
4. In the dialog box that appears, click Go to open the Log Service console.

5. In the Log Service console, you can set the following advanced features for log projects and logstores:
   - *Real-time log collection and consumption*
   - *Shipping log data to other Alibaba Cloud storage services in real time*
   - *Providing visual representations with other products*

4.7.9 Grant log query and analysis permissions to a RAM user

If you want to use the WAF log query and analysis service with a RAM user, you must grant required permissions to the RAM user using the Alibaba Cloud account.

Context

The following permissions are required for enabling and using the WAF log query and analysis service.

Operation	Required account type and permissions
Enable Log Service (the service remains enabled after this operation)	Alibaba Cloud account
Authorize WAF to write log data to the exclusive logstore in Log Service in real-time (the authorization remains valid after this operation)	· Alibaba Cloud account · RAM user that has the `AliyunLogFullAccess` permission · RAM user that has specific permissions
Use the log query and analysis service	· Alibaba Cloud account · RAM user that has the `AliyunLogFullAccess` permission · RAM user that has specific permissions

Grant permissions to RAM users as required.

Scenario	Permission	Procedure
Grant permissions on all Log Service operations to a RAM user.	`AliyunLogFullAccess`	For more information, see *RAM users*.

Scenario	Permission	Procedure
Grant the log viewing permission to a RAM user after you enable the WAF log query and analysis service and complete the authorization on the Alibaba Cloud account.	`AliyunLogR eadOnlyAcc ess`	For more information, see *RAM users*.
Grant the RAM user permissions on enabling and using the WAF log query and analysis service . This RAM user is not granted other administra tive permissions on Log Service.	Custom authorization policy	For more information, see the following procedure.

Procedure

1. Log on to the *RAM console*.
2. On the Policies page, select the Custom Policy tab.
3. In the upper-right corner of the page, click Create Authorization Policy.
4. Click Create Authorization Policy. In the template, specify the Authorization Policy Name, and then enter the following in the Policy Content field.



Note:

Replace `${ Project }` and `${ Logstore }` in the following policy content with the names of the exclusive project and logstore in WAF Log Service.

```
{
  " Version ": " 1 ",
  " Statement ": [
   {
      " Action ": " log : GetProject ",
      " Resource ": " acs : log :*:*: project /${ Project }",
      " Effect ": " Allow "
    },
    {
      " Action ": " log : CreateProj  ect ",
      " Resource ": " acs : log :*:*: project /*",
      " Effect ": " Allow "
    },
 {
      " Action ": " log : ListLogSto  res ",
      " Resource ": " acs : log :*:*: project /${ Project }/
 logstore /*",
```

```
      " Effect ": " Allow "
    },
    {
      " Action ": " log : CreateLogS  tore ",
      " Resource ": " acs : log :*:*: project /${ Project }/
logstore /*",
      " Effect ": " Allow "
    },
{
      " Action ": " log : GetIndex ",
      " Resource ": " acs : log :*:*: project /${ Project }/
logstore /${ Logstore }",
      " Effect ": " Allow "
    },
    {
      " Action ": " log : CreateInde  x ",
      " Resource ": " acs : log :*:*: project /${ Project }/
logstore /${ Logstore }",
      " Effect ": " Allow "
    },
    {
      " Action ": " log : UpdateInde  x ",
      " Resource ": " acs : log :*:*: project /${ Project }/
logstore /${ Logstore }",
      " Effect ": " Allow "
    },
    {
      " Action ": " log : CreateDash  board ",
      " Resource ": " acs : log :*:*: project /${ Project }/
dashboard /*",
      " Effect ": " Allow "
    },
{
      " Action ": " log : UpdateDash  board ",
      " Resource ": " acs : log :*:*: project /${ Project }/
dashboard /*",
      " Effect ": " Allow "
    },
{
      " Action ": " log : CreateSave  dSearch ",
      " Resource ": " acs : log :*:*: project /${ Project }/
savedsearc  h /*",
      " Effect ": " Allow "
    },
{
      " Action ": " log : UpdateSave  dSearch ",
      " Resource ": " acs : log :*:*: project /${ Project }/
savedsearc  h /*",
      " Effect ": " Allow "
    }
  ]
}
```

5. Click Create Authorization Policy.
6. Go to the Users page, find the RAM user, and then click Authorize.
7. Add the authorization policy that you created and click OK.

   This RAM user can enable and use the WAF log query and analysis service, and cannot use other features of Log Service.

4.7.10 Manage log storage

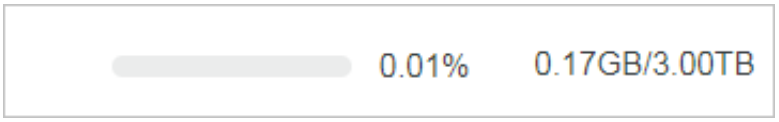
After WAF Log Service is activated, log storage is allocated for your WAF Log Service based on the specified log storage size. You can view the usage of the log storage on the Log Service page in the Web Application Firewall console.

View the usage of the log storage

You can view the usage of the log storage that is generated by the WAF log query and analysis service at any time.

> Note:
> It takes two hours for changes in the storage usage to be updated in the console. You need to upgrade the log storage when only a little log storage space is available.

1. Log on to the *Web Application Firewall console*.
2. Choose App Market > App Management, select the region where your WAF instance is located, and then click Real-time Log Query and Analysis Service.
3. At the top of the Log Service page, view the usage of log storage.



Upgrade log storage

To upgrade the log storage size, click Upgrade Storage at the top of the Log Service page.

> Note:
> If log storage is full, new log data cannot be written to the exclusive logstore. We recommend that you upgrade log storage before log storage is full.

Clear log storage

You can delete all log entries in the log storage as needed. For example, you can delete the log entries generated during the test phase to make full use of the log storage by recording only log entries that is generated during the production phase.

Click Clear at the top of the Log Service page, and click Confirm to delete all log entries in the log storage.

> Notice:

Log entries that are deleted cannot be restored. Delete log entries with caution.

Note:
You can clear the log storage for only a limited number of times.

4.8 Anti-Bot logs

4.8.1 Enable Log Service for Anti-Bot

Log Service can collect access logs and protection logs from the websites protected by Anti-Bot in real time. Also, it can retrieve and analyze the collected log data in real time.

You can analyze the website access and attack behaviors based on the website logs collected in the Anti-Bot console in real time. This further allows you to assist your security management personnel in developing protection policies.

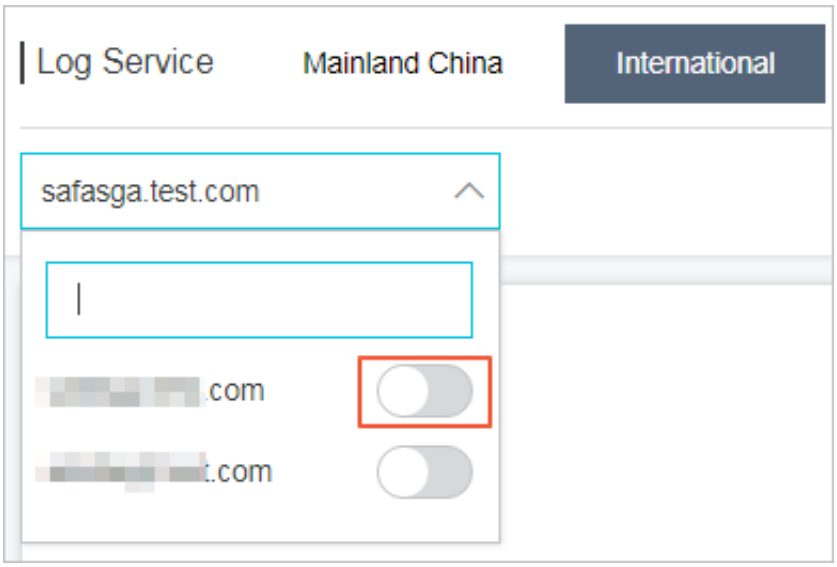
Procedure

1. Log on to the *Anti-Bot console*.
2. Choose Reports > Log Service. Select the region where your instance is located.

   Note:
   If you are using Log Service for Anti-Bot for the first time, click Authorize to authorize Anti-Bot to store all the recorded logs in your logstore as instructed.
3. From the Website Domain drop-down list, select the website domain name for which you want to enable Log Service. Then, click Enable.

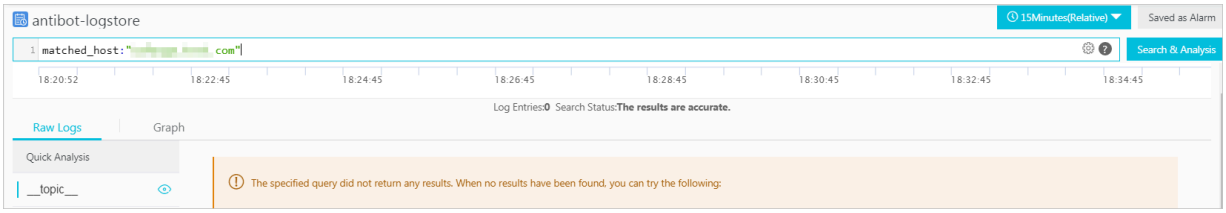
   Note:

The Website Domain drop-down list displays all the website domain names configured with Anti-Bot.



Now, Log Service has been enabled for the website domain name. Log Service automatically creates a dedicated logstore for your Alibaba Cloud account. Anti-Bot automatically imports the logs of all the website domain names enabled with Log Service to the dedicated logstore in real time.

Then, you can retrieve and analyze the access logs of the website domain names enabled with Log Service.



Restrictions and instructions

- Other data cannot be written to the dedicated logstore.

  Note:
  The website logs recorded by Anti-Bot are stored in the dedicated logstore, where you cannot write other data through APIs and SDKs.
- Currently, the basic settings (such as the storage period) of the dedicated logstore cannot be modified.
- Do not delete or modify the default settings created by Log Service, such as the default project, logstore, index, and dashboard.

- Log Service updates and upgrades the log query analysis function from time to time. The indexes and default reports of the dedicated logstore will be automatica lly updated.
- If the RAM user requires the log query analysis function, grant the related Log Service permissions to the RAM user through RAM.

4.8.2 Log field description

Log Service for Anti-Bot Service (Anti-Bot) records the access logs and attack and defense logs of protected website domain names in detail. A log contains dozens of fields. You can select specific fields for query analysis as needed.

Field	Description	Example
__topic__	The log topic. This field is invariably set to antibot_ac cess_log.	antibot_access_log
antibot	The type of the triggered Anti-Bot protection policy, including: · ratelimit: rate limiting · sdk: app protection · algorithm: algorithm pattern · intelligence: bot intelligence · acl: access control list · blacklist: blacklist	ratelimit
antibot_action	The operation specified by the Anti-Bot protection policy, including: · `challenge` : Deliver a JavaScript script for verification · `drop` : Intercept · `captcha` : Verify by dragging a slider · `report` : Monitor only	drop
antibot_rule	The ID of the triggered Anti-Bot protection rule.	5472

Field	Description	Example
antibot_verify	The result of the verification performed by Anti-Bot. Note: This value is recorded when the antibot_action field is set to challenge or captcha. · challenge_fail: JavaScript verification fails. · challenge_pass: JavaScript verification is passed. · captcha_fail: Slide captcha verification fails. · captcha_pass: Slide captcha verification is passed.	challenge_fail
block_action	The type of the bot protection that is triggered. The value is invariably set to `antibot` .	antibot
body_bytes_sent	The size of HTTP body (in byte) sent to the client.	2
content_type	The content type of the access request.	application/x-www-form-urlencoded
host	The source website.	api.aliyun.com
http_cookie	The cookie information about the access client, which is included in the access request header.	k1=v1;k2=v2
http_referer	The source URL of the access request, which is included in the access request header. `-` is displayed if no source URL is available.	http://xyz.com
http_user_agent	The User Agent field in the access request header, which typically includes the web browser identifier and operating system identifier of the source client.	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build/HUAWEIEDISON-AL10)

Field	Description	Example
http_x_for warded_for	The XFF header information in the access request header , which is used to identify the original IP addresses of the clients connected to a web server through the HTTP proxy or SLB.	-
https	Whether the access request is an HTTPS request. Valid values: · true: The access request is an HTTPS request. · false: The access request is an HTTP request.	true
matched_host	The matched domain name configured with Anti-Bot, which may be a wildcard domain name. - is displayed if no related domain name configuration is matched.	*.aliyun.com
real_client_ip	The actual IP address of the access client. - is displayed if no actual IP address is retrieved.	1.2.3.4
region	The information about the region where the Anti-Bot instance is located.	cn
remote_addr	The IP address of the client that initiates the access request.	1.2.3.4
remote_port	The port of the client that initiates the access request.	23713
request_length	The length of the access request . Unit: bytes.	123
request_method	The HTTP request method of the access request.	GET
request_path	The relative path of the request (excluding the query string).	/news/search.php
request_time_msec	The duration of the access request. Unit: milliseconds.	44

Field	Description	Example
request_traceid	The unique ID of the access request.	7837b117154103869434 37009ea1f0
server_protocol	The protocol and version of the response returned by the origin server.	HTTP/1.1
status	The status of the HTTP response that Anti-Bot returns to the client.	200
time	The occurrence time of the access request.	2018-05-02T16:03:59+08:00
ua_browser	The information about the web browser that initiates the access request.	ie9
ua_browser_family	The family of the web browser that initiates the access request.	internet explorer
ua_browser_type	The type of the web browser that initiates the access request.	web_browser
ua_browser_version	The version of the web browser that initiates the access request.	9.0
ua_device_type	The device type of the client that initiates the access request.	computer
ua_os	The operating system of the client that initiates the access request.	windows_7
ua_os_family	The operating system family of the client that initiates the access request.	windows
upstream_addr	The origin address list of Anti-Bot in the format of `IP address : Port`. Separate multiple IP addresses with commas (,).	1.2.3.4:443

Field	Description	Example
upstream_ip	The origin IP address corresponding to the access request. For example, if Anti-Bot forwards the access request to an ECS instance, this parameter returns the IP address of the back-to-origin ECS instance.	1.2.3.4
upstream_response_time	The time for the origin server to respond to an Anti-Bot request. Unit: seconds. The response times out if "-" is returned.	0.044
upstream_status	The status of the response that the origin server returns to Anti-Bot. No response is available if "-" is returned. For example, the request is intercepted by Anti-Bot, or the response returned by the origin server times out.	200
user_id	AliUID of the Alibaba Cloud account.	12345678
wxbb_action	If the protection type of Anti-Bot is app protection, the following actions are supported: · `close` : intercepts requests. That is, the antibot_action field is set to `drop` . · `test` : only monitors requests. That is, the antibot_action field is set to `report` . · `pass` : allows requests. Note: This field is set to – if SDK protection is not configured.	close
wxbb_invalid_wua	For more information about app protection, consult your technical engineer.	valid wua

4.9 ActionTrail access logs

4.9.1 Overview

At present, ActionTrail of Alibaba Cloud is in connection with Log Service, which provides functions of log collection and analysis in real time. The operation log data collected by ActionTrail is delivered to Log Service in real time. Log Service provides rich functions such as real-time query and analysis, and dashboard presentation for this part of logs.

As more and more enterprises adopt information technology and cloud computing technology to improve efficiency and service quality, attacks on networks, devices, and data of enterprises and organizations never stops upgrading. These attacks are generally aimed at making profits other than causing damages, and are increasing ly good at hiding themselves. As a result, discovering and recognizing these attacks become increasingly challenging.

As the basis of audit and security backtracing, operation logs of enterprise IT and data resources are always of high significance. With the mature development of network information technology and the in-depth implementation of the "Network Security Law", enterprises and organizations are paying more and more attention to the preservation and analysis of operation logs. Operation records of resources in cloud computing are a very important type of logs.

ActionTrail records operations on your cloud account resources, provides operation record query, and saves record files to your specified Object Storage Service (OSS) or Log Service. With all operation records saved by ActionTrail, you can perform security analysis, resource change tracking and compliance audit.

ActionTrail collects API calling records of cloud services (including API calling records triggered by operations on the console). After the normalization process, the operation records are saved in the form of JSON and are available for delivery. In general, when you initiate a calling operation through the console or SDK, ActionTrail collects a log of the operation behavior in ten minutes.

At present, *ActionTrail* is in connection with Log Service, which provides functions of log collection and analysis in real time. The operation log data collected by ActionTrail is delivered Log Service in real time. Log Service provides rich functions such as real-time query and analysis, and dashboard presentation for this part of log.

Benefits

- Simple configuration: Easily configure to collect real-time logs. For information about configuration steps and log fields, see *Procedure*.
- Real-time analysis: Relying on Log Service, it provides real-time log analysis, an out-of-the-box report center, and details available for real-time mining with records of operations on important cloud assets.
- Real-time alarms: Supports custom quasi-real-time monitoring and alarming based on specific indicators to ensure timely response to critical business exceptions.
- Ecosystem: Supports dock with other ecosystems such as stream computing, cloud storage, and visualization solutions to further explore data value.
- Free quota: Provides 500 MB free quotas of data import and storage per month. You can expand the storage time for compliance, traceability, and filing. The storage service without time limitation is provided at a low price of 0.0875 USD/GB/month. For information about billing, see *Billing method*.

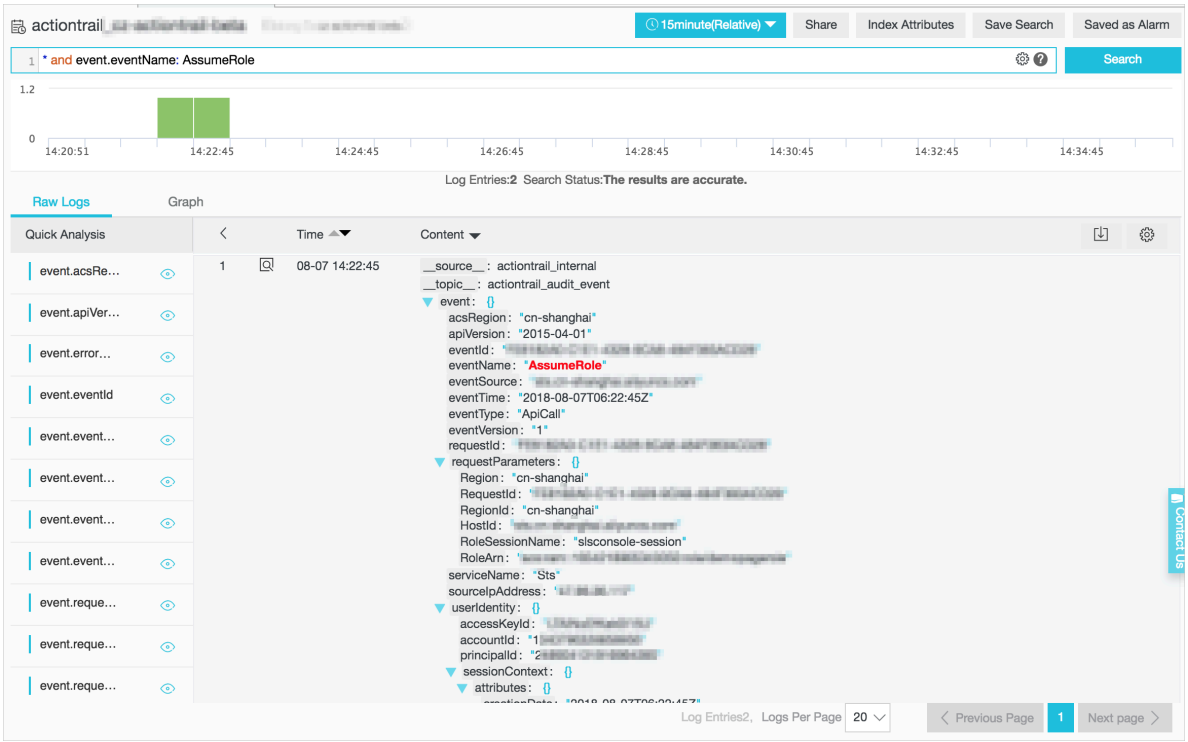
Application scenarios

- Troubleshooting and analysis for abnormal operations

  Monitors cloud resource operations under all names in real time and supports real-time troubleshooting and analysis for abnormal operations. Accidental deletion, high-risk operations, and other operations can be traced through logging.

  For example, to view the Elastic Compute Service (ECS) release operation log:

  Figure 4-28: View the ECS release operation log

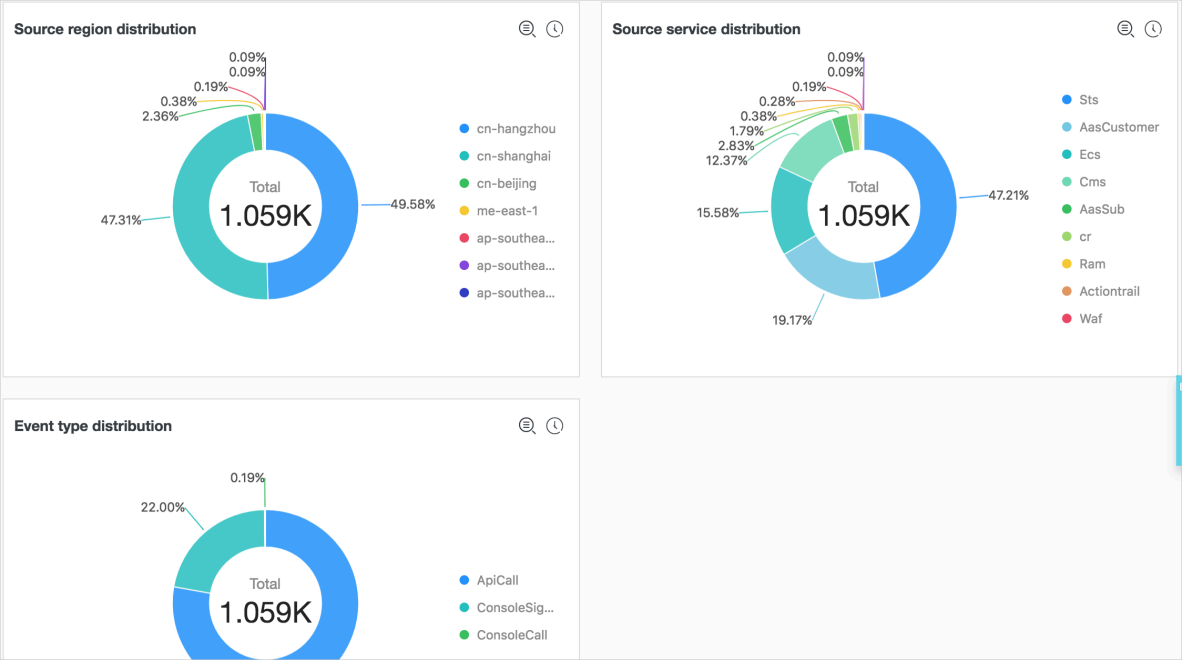


- Distribution and source tracking of important resource operations

  You can track and trace the distribution and source of important resource operations by analyzing the log content, and specify and optimize resolution strategies based on the analysis results.

  For example, to view the country distribution of operators who deleted the Relational Database Service (RDS):

  Figure 4-29: View the distribution of RDS deletion



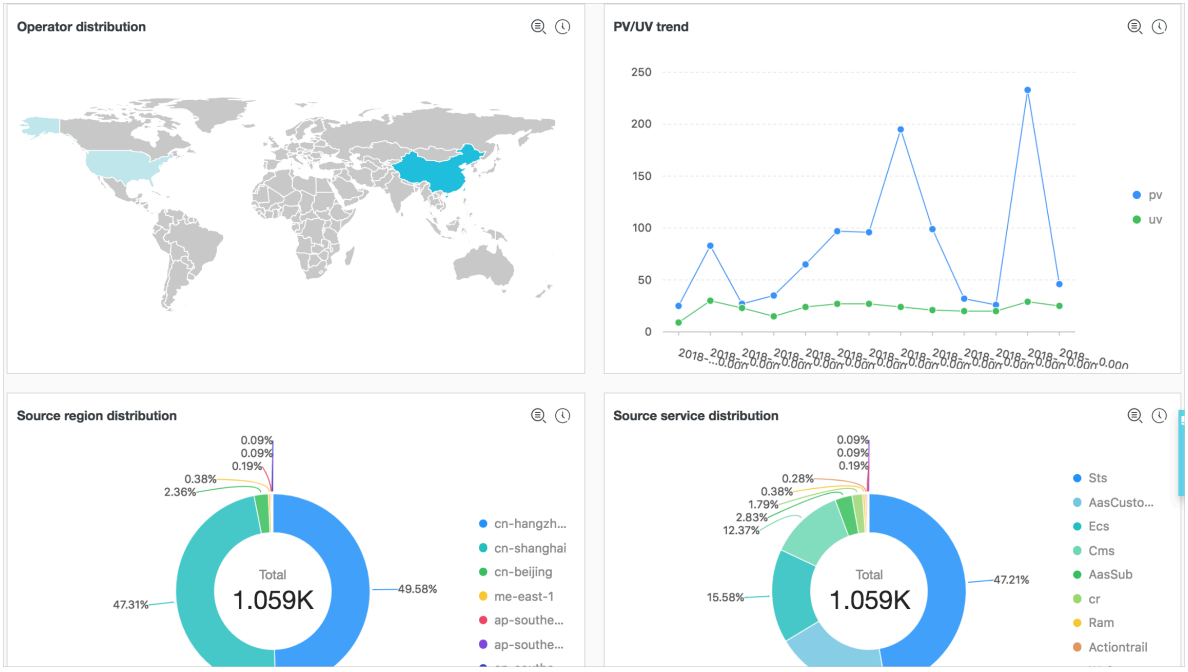
- Resource operation distribution view

  You can query and analyze the collected ActionTrail operation logs through SQL query statements in real time, and view the distribution and time trends of all resource operations, and other operation and maintenance actions. By doing this , you assist the operation and maintenance personnel to monitor the resource

running status in real time. Operation and maintenance reliability indicators are clear at a glance.

For example, to view trends of failed operations:

Figure 4-30: Trends of failed operations



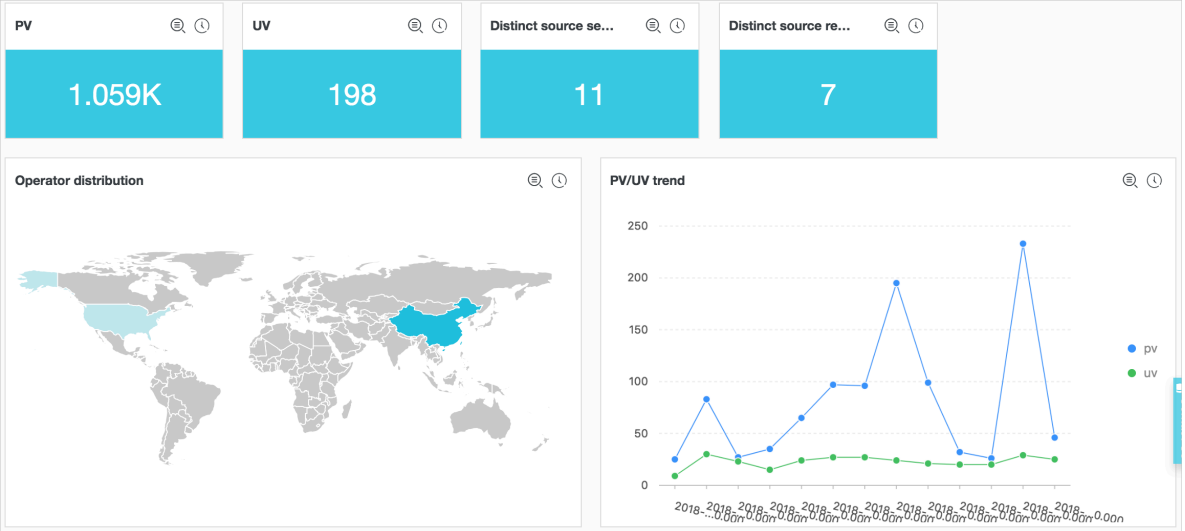
- Real-time analysis of operation data

  Customize diverse query statements based on operation requirements, customize fast queries and analysis dashboard for different data requirements, and you can

also customize real-time data dashboard for data such as resource usage status and user logon status.

For example, to view the frequency distribution of operators from network operators:

Figure 4-31: Frequency distribution of operators from network operators



4.9.2 Procedure

At present, ActionTrail is in connection with Log Service. Operation log data collected by ActionTrail is delivered to Log Service in real time. This document introduces the log fields and collection procedures of ActionTrail logs.

Prerequisites

1. Enable Log Service
2. Enable *ActionTrail service*.

Procedure

1. Log on to the ActionTrail console.
2. Click Trail list in the left-side navigation pane to go to the Trail list page.
3. Click Create Trail in the upper-right corner to go to the Create Trail page.

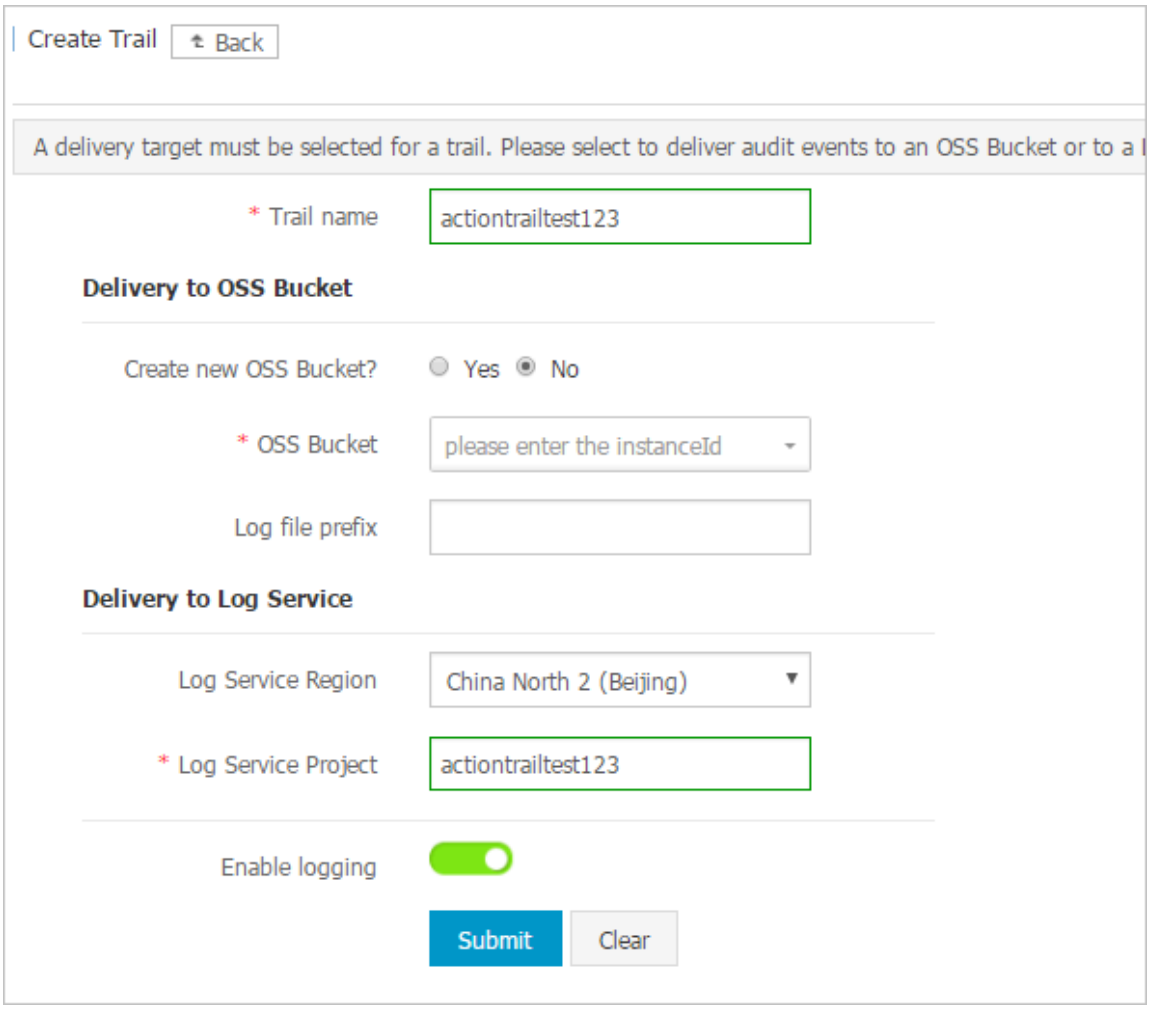
4. Configure trail parameters.
   a. Enter Trail name.
   b. Deliver audit events to an OSS Bucket (optional).

      For more information, see *Create trail*.
   c. Select an region in Log Service Region.
   d. Enter Log Service Project

      The project is used to store ActionTrail logs. You can enter an existing project name under the selected region or enter a new project name to deliver the logs to the new project.
   e. Enable logging.

      Click Enable logging. After you enable this feature, operation logs of cloud resource recorded by your ActionTrail is delivered to Log Service.

      Figure 4-32: Configure trail parameters.

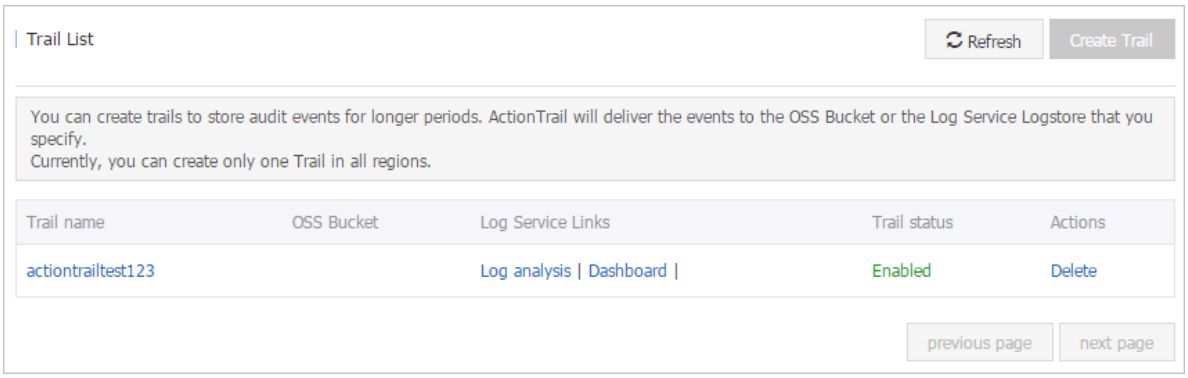


5. Click Submit to complete the configuration.

   You have created a trail and you can view the created trail in Trail List.

   > Note:
   > If you configure ActionTrail log collection for the first time, please authorize ActionTrail to upon prompts on the page. The authorization enables ActionTrail to distribute ActionTrail logs to your Logstore. Click Submit again after the authorization is complete to end the configuration.

   Figure 4-33: Trail List



Limits

- Only one trail can be created for an account.

  Trail helps you deliver audit events to an OSS bucket or Log Service Logstore specified by you. Currently, only one trail can be created for an account in all regions. This trail delivers audit events across all regions to both or either of the OSS bucket and Logstore.

- If you have created a trail, you can handle the trail in only the region where the trail was created.

  If you have created a trail, you can view, modify, or delete the trail in only the region where the trail was created. For example, if you need to configure a trail of Log Service when you have created a trail of OSS, add Log Service configuration to your created trail of OSS.

- The exclusive Logstoree does not support writing additional data.

  The exclusive Logstore is used to store only operation logs of Action Trail. Therefore, this Logstore does not support writing other data. Other functions, such as query, statistics, alarms, and streaming consumption, have no restrictions.

- Pay-As-You-Go.

  The ActionTrail log collection feature uses the billing method of Log Service. Log Service supports Pay-As-You-Go billing method, and provides a certain amount of free quota. For more information, see *Billing method*.

Query and analysis

To query and analyze collected log data after you complete trail configuration, click Log Analysis and Log Report under Log Service list in the Trail List page.

- Log Analysis: Enter the log query and analysis page.

  Log Service provides log query and analysis. In this page, you can query and analyze collected ActionTrail logs in real time.

  By defining query syntax and analysis syntax, Log Service provides log queries in a variety of complex scenarios. For information about query and analysis syntax, see *Query syntax* and *Analysis syntax*.

  To monitor important log data at intervals and set alarm notifications for abnormal conditions, save the current query conditions as quick queries and alarms on the query page. For detailed procedures, see *Configure an alarm*.

- Log Report: Enter the dashboard page.

  Log Service shows an overall view of real-time dynamics, such as event types and event sources, by a built-in dashboard exclusive to ActionTrail.

  You can modify the exclusive dashboard, create a custom dashboard, and add custom analysis charts in a variety of scenarios to your dashboard. For more information about dashboards, see *Create and delete a dashboard*.

Default configuration

When the configuration is completed, Log Service creates an exclusive project and an exclusive Logstore for you. Operation logs of cloud resource collected by ActionTrail is delivered to the Logstore in real time. In addition, Log Service also creates a dashboard for you to view cloud resource operations in real time. For information about default configurations such as the project and Logstore, see the following table.

Table 4-4: Default configuration

Default configuration item	Configuration content
Project	A project that you select or customize when you create the trail.
Logstore	By default, Logstore is created. The Logstore name is `actiontrail_Trail name`. All logs of ActionTrail are saved in this Logstore.
Region	A region that you select when you create the trail.
Shard	By default, two shards are created and the *Auto Split Shard* feature is enabled.
Log storage time	By default, logs are saved permanently. You can customize the log storage time to a value in the range of 1 to 3000 days. For detailed procedures, see *Manage a Logstore*.
Dashboard	By default, a dashboard is created: · Chinese environment: `actiontrail_Trail name_audit_center_cn` · English environment: `actiontrail_Trail name_audit_center_en`

Log field

Field name	Name	Example
__topic__	Log topic.	This field is fixed at `actiontrail_audit_event`
event	Event body, which is in the JSON format. The content of the event body varies with the event.	*event example*
event.eventId	The ID of the event, which uniquely indicates the event.	07F1234-3E1D-4BFF-AC6C-12345678
event.eventName	Event name.	CreateVSwitch

Field name	Name	Example
event.eventSource	The source of the event.	http://account.aliyun.com:443/login/login_aliyun.htm
event.eventType	Event type.	ApiCallApicall
event.eventVersionEvent.eventversion	The version of the data format of ActionTrail, which is currently fixed to 1.	1
event.acsRegion	The region where the event is located.	cn-hangzhou
event.requestId	The request ID of the cloud service operation.	07F1234-3E1D-4BFF-AC6C-12345678
event.apiVersion	The version of the related API.	2017-12-04
event.errorMessage	The error message of an event failure.	unknown confidential
event.serviceName	The event-related service name.	Ecs
event.sourceIpAddress	The Source IP associated with the event.	1.2.3.4
event.userAgent	The event-related client agent.	Mozilla/5.0 (...)
event.requestParameters.HostId	The host ID in the request-related parameter.	ecs.cn-hangzhou.aliyuncs.com
event.requestParameters.Name	The name in the request-related parameter.	ecs-test
event.requestParameters.Region	The domain in the request-related parameter.	cn-hangzhou
event.userIdentity.accessKeyId	The AccessKey ID used by the request.	25 **********
event.userIdentity.accountId	The ID of the account requested.	123456
event.userIdentity.principalId	The voucher ID of the account requested.	123456
event.userIdentity.type	The type of account requested.	root-account

Field name	Name	Example
event.userIdentity.userName	The name of account requested.	root

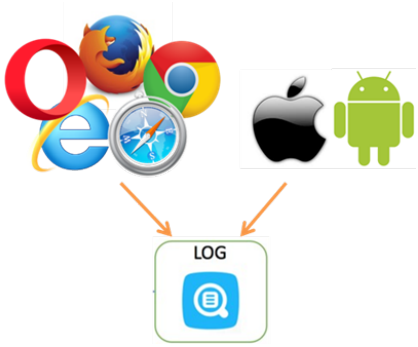
event example

```
{
  " acsRegion ": " cn - hangzhou ",
  " additional  EventData ": {
    " isMFACheck  ed ": " false ",
    " loginAccou  nt ": " test1234 @ aliyun . com "
  },
  " eventId ": " 7be1e173 - 1234 - 44a1 - b135 - 1234 ",
  " eventName ": " ConsoleSig  nin ",
  " eventSourc  e ": " http :// account . aliyun . com : 443 / login /
 login_aliy  un . htm ",
  " eventTime ": " 2018 - 07 - 12T06 : 14 : 50Z ",
  " eventType ": " ConsoleSig  nin ",
  " eventVersi  on ": " 1 ",
  " requestId ": " 7be1e173 - 1234 - 44a1 - b135 - 1234 ",
  " serviceNam  e ": " AasCustome  r ",
  " sourceIpAd  dress ": " 42 . 120 . 75 . 137 ",
  " userAgent ": " Mozilla / 5 . 0  ( Macintosh ;  Intel   Mac   OS
 X   10_13_6 )  AppleWebKi  t / 537 . 36  ( KHTML ,  like   Gecko )
 Chrome / 67 . 0 . 3396 . 99   Safari / 537 . 36 ",
  " userIdenti  ty ": {
    " accessKeyI  d ": " 25 ****************",
    " accountId ": " 1234 ",
    " principalI  d ": " 1234 ",
    " type ": " root - account ",
    " userName ": " root "
  }
}
```

5 Other collection methods

5.1 Web Tracking

Log Service supports collecting logs from HTML, H5, iOS, and Android platforms by using Web Tracking, and customizing dimensions and metrics.



As shown in the preceding figure, you can collect user information from various browsers, iOS apps, and Android apps (apart from *iOS/Android SDK*) by using Web Tracking. For example:

- Browsers, operating systems, and resolutions used by users.
- Browsing behaviors of users, such as the clicking behaviors and purchasing behaviors on the website.
- The staying time in the app for users and whether the users are active or not.

> Note:
> Using Web Tracking means that this Logstore enables the anonymous write permission of the Internet, and dirty data may be generated.

Precautions

- Using Web Tracking means that this Logstore enables the anonymous write permission of the Internet without valid authentication, and dirty data may be generated.
- Only Get requests are supported. A request body exceeding 16 KB cannot be uploaded.

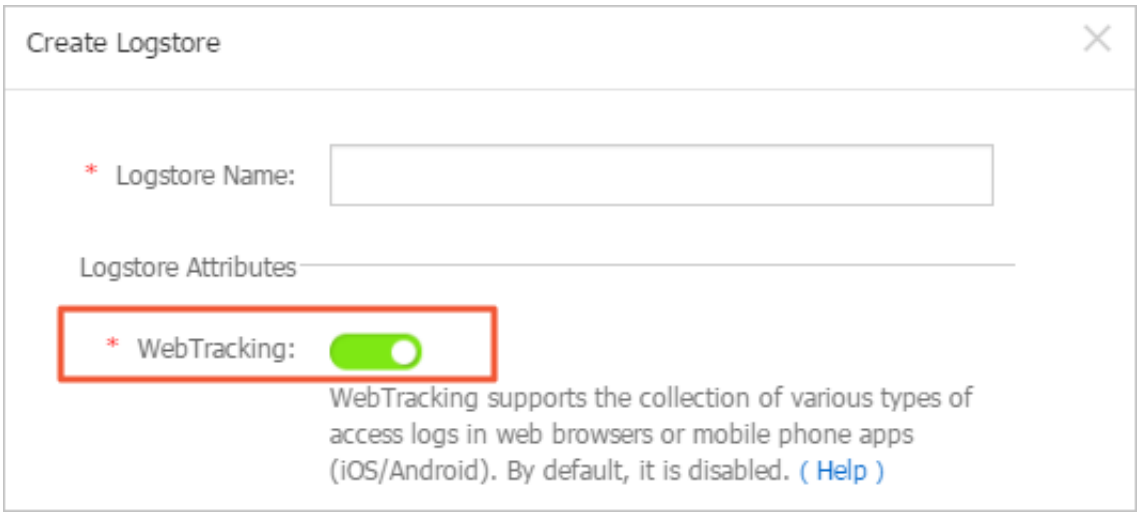
Procedure

Step 1 Enable Web Tracking

You can enable Web Tracking in the console or by using Java SDK.

- Enable Web Tracking in the console

  1. On the Logstore List page, click Modify at the right of the Logstore that must enable the Web Tracking function.
  2. Turn on the Web Tracking switch.



- Enable Web Tracking by using Java SDK

  Java SDK:

```
import com . aliyun . openservic es . log . Client ;
import com . aliyun . openservic es . log . common . LogStore ;
import com . aliyun . openservic es . log . exception .
LogExcepti on ;
public class WebTrackin g {
  static private String accessId = " your accesskey id
";
  static private String accessKey = " your accesskey ";
  static private String project = " your project ";
  static private String host = " log service data
address ";
  static private String logStore = " your logstore ";
  static private Client client = new Client ( host ,
accessId , accessKey );
  public static void main ( String [] args ) {
      try {
        // Enable the Web Tracking function on the
created Logstore .
         LogStore logSt = client . GetLogStor e ( project ,
logStore ). GetLogStor e ();
         client . UpdateLogS tore ( project , new LogStore (
logStore , logSt . GetTtl (), logSt . GetShardCo unt (), true
));
        // Disable the Web Tracking function .
```

```
            // client . UpdateLogS  tore ( project ,  new   LogStore (
 logStore ,  logSt . GetTtl (),  logSt . GetShardCo  unt (),  false
 ));
            // Create   a   Logstore   that   supports   the   Web
 Tracking   function .
            // client . UpdateLogS  tore ( project ,  new   LogStore (
 logStore ,  1 ,  1 ,  true ));
        }
         catch  ( LogExcepti  on   e ){
             e . printStack  Trace ();
        }
    }
}
```

Step 2 Collect logs

After the Web Tracking function is enabled for the Logstore, you can use any of the following methods to upload data to the Logstore.

> **Note:**
> We recommend that you use the SDK to upload logs.

- Use the JS SDK

  1. Copy *loghub - tracking . js* to the *web* directory, and introduce the following script on the page:

     Click to download.

     ```
     < script   type =" text / javascript "  src =" loghub - tracking .
      js "  async ></ script >
     ```

     > **Note:**
     > To keep page loading running, the script sends HTTP requests asynchronously. If data must be sent several times in the page loading process, the subsequent request overwrites the preceding HTTP request, and the browser shows the tracking request exits. Sending requests synchronously can help to avoid this problem. To send requests synchronously, replace the statement in the script.

Original script:

```
this . httpReques  t_ . open (" GET ",  url ,  true )
```

Replace the last parameter to send requests synchronously:

```
this . httpReques  t_ . open (" GET ",  url ,  false )
```

2. **Create a Tracker object.**

```
var   logger  =  new   window . Tracker ('${ host }','${ project
}','${ logstore }');
logger . push (' customer ', ' zhangsan ');
logger . push (' product ', ' iphone   6s ');
logger . push (' price ',  5500 );
logger . logger ();
logger . push (' customer ', ' lisi ');
logger . push (' product ', ' ipod ');
logger . push (' price ',  3000 );
logger . logger ();
```

The parameter meaning are as follows:

Field	Definition
`${ host }`	The domain name of the region where your Log Service is located.
`${ project }`	The name of the project created in Log Service.
`${ logstore }`	The name of the Logstore with the Web Tracking function enabled under `${ project }`.

After running the preceding commands, you can see the following two logs in Log Service:

```
customer : zhangsan
product : iphone   6s
price : 5500
```

```
customer : lisi
product : ipod
```

```
price : 3000
```

- Use HTTP GET request

```
curl -- request GET ' http ://${ project }.${ host }/ logstores
/${ logstore }/ track ? APIVersion = 0 . 6 . 0 & key1 = val1 &
key2 = val2 '
```

The parameter meanings are as follows.

Field	Definition
`${ project }`	The name of the project created in Log Service.
`${ host }`	The domain name of the region where your Log Service is located.
`${ logstore }`	The name of the Logstore with the Web Tracking function enabled under `${ project }`.
`APIVersion = 0 . 6 . 0`	The reserved field, which is required.
`__topic__ = yourtopic`	Specify the log topic, reserved fields (optional).
`key1 = val1 , key2 = val2`	The key-value pairs to be uploaded to Log Service. Multiple key-value pairs are supported, but you must make sure that the URL length is less than 16 KB.

- Use the HTML IMG tag

```
< img src =' http ://${ project }.${ host }/ logstores /${
logstore }/ track . gif ? APIVersion = 0 . 6 . 0 & key1 = val1 &
key2 = val2 '/>
< img src =' http ://${ project }.${ host }/ logstores /${
logstore }/ track_ua . gif ? APIVersion = 0 . 6 . 0 & key1 = val1
& key2 = val2 '/>
```

The parameter meanings are the same as those in Use HTTP GET request. In addition to uploading custom parameters, track_ua.gif transmits UserAgent and referer of in the HTTP header as log fields on the server.



Note:

To collect referer of the HTTPS page, the link of the preceding Web Tracking must be the HTTPS type.

After data is uploaded to Log Service, you can use *LogSearch/Analytics* of Log Service to search and analyze log data in real time, and display real-time analysis results with

various visualization solutions. You can also consume data by using *Consumer Library* provided by Log Service.

5.2 Use the Kafka protocol to collect data to Log Service

This topic describes how to use the Kafka protocol to collect data to Log Service.

Limits

- Supported Kafka protocol versions are 0.8.0 to 2.1.IV2.
- You must use the SASL_SSL connection protocol for secure data transmission.
- If multiple shards are contained in your Logstore, enable load balance mode before you enter data.

Required parameters

If you use the Kafka protocol, you must set the following parameters.

Configuration	Description	Example
Connection protocol	You must use the SASL_SSL connection protocol for secure data transmission.	SASL_SSL
`hosts`	The cluster address for the initial connection. The port number of an intranet (for either a Classic network or VPC) address is 10011. The port number of the Internet address is 10012. You need to select the service port where your target project is located. For more information, see *Service endpoint*.	cn-hangzhou-intranet.log.aliyuncs.com:10011 cn-hangzhou.log.aliyuncs.com:10012
`topic`	The `topic` parameter in Kafka is mapped to the Logstore name in Log Service. You must create a Logstore before you begin to collect data.	test-logstore-1

Configuration	Description	Example
`username`	This parameter is mapped to the project name in Log Service.	<yourusername>
`password`	This parameter denotes your AK, which is in the format of ${access-key-id}#${access-key-secret}. You need to replace ${access-key-id} with your AccessKey ID, and replace ${access-key-secret} with your AccessKey Secret. Furthermore, we recommend that you use the AK of a RAM user. For more information, see *Grant a RAM user the permissions to access Log Service*.	<yourpassword>
`Certificat e`	Each domain name in Log Service has a valid certificate. You only need to use the default root certificate.	/etc/ssl/certs/ca-bundle.crt

Errors

If you fail to use the Kafka protocol to collect log data, the system returns a corresponding error code for the specific cause of failure. For more information about the errors corresponding the Kafka protocol, see *Error list*. The following are common errors and their description and corresponding solutions.

Error code	Description	Solution
NetworkException	Indicates a network failure.	Try again one second later.

Error code	Description	Solution
TopicAuthorizationException	Indicates an authorization exception. Your AccessKey is invalid or does not have the permissions for the corresponding project or Logstore.	Enter a valid AccessKey and make sure that it has the required permissions.
UnknownTopicOrPartitionException	Indicates that the topic or partition does not exist. The error can be caused by either of the following conditions: · The corresponding project or Logstore does not exist. · The region where the project is located is different from the region indicated by the endpoint that you entered.	Check if you have created the corresponding project and Logstore before you collect data. Check if the region where the project is located is the same as the region indicated by the endpoint that you entered.
KafkaStorageException	Indicates a server exception.	Try again one second later.

Configuration examples

In this example, data is collected to Log Service. The project in Log Service is named `test-project-1`, and the Logstore is named `test-logstore-1`. The region where the project is located is `cn-hangzhou`. The AccessKey ID with the corresponding write permission is `<yourAccessKeyId>`, and the AccessKey Secret is `<yourAccessKeySecret>`.

- **Example 1: Use Beats software to collect data to Log Service**

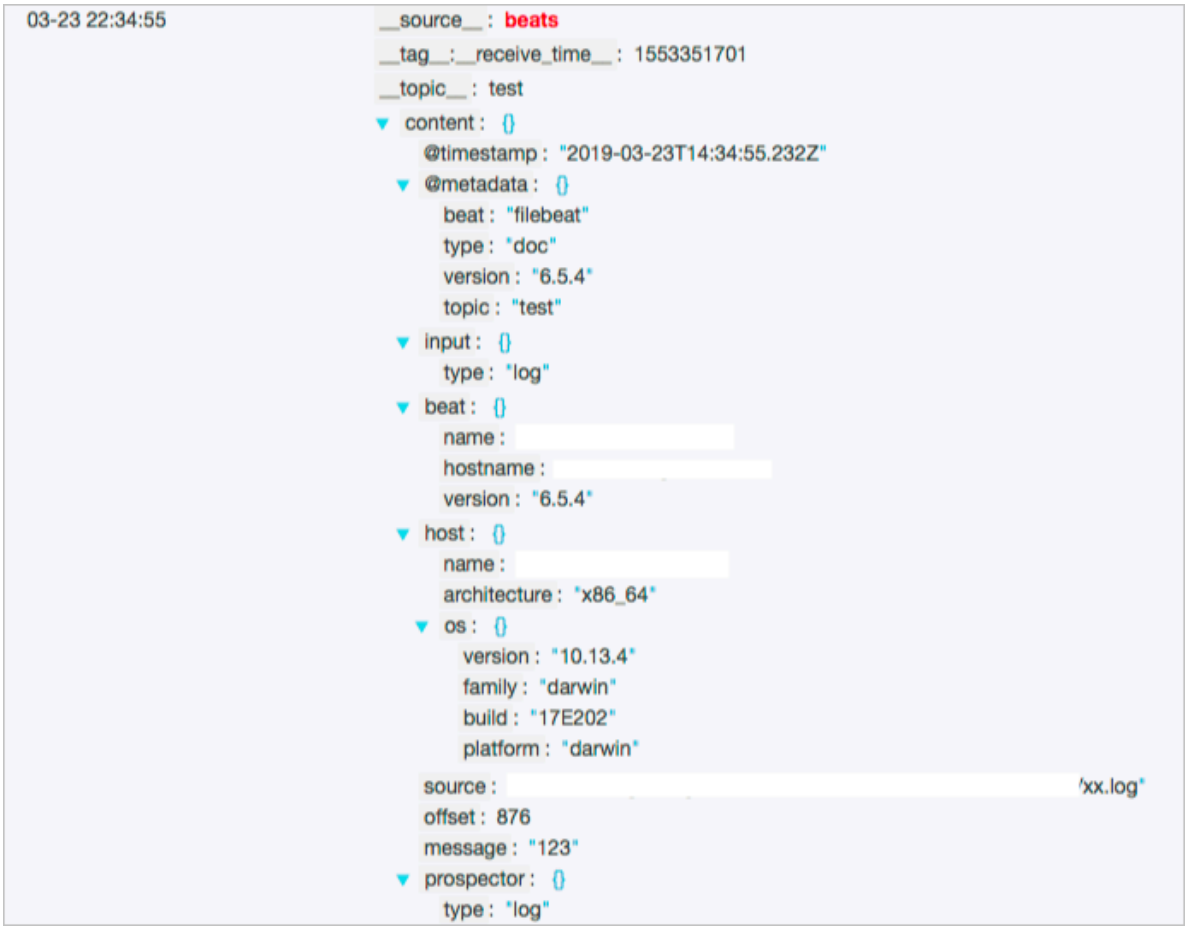
  You can output the collected data to Kafka by using Beats software such as MetricBeat, PacketBeat, Winlogbeat,Auditbeat, Filebeat, and Heartbeat. For more information, see *Beats-Kafka-Output*.

```
output . kafka :
      #  initial   brokers   for   reading   cluster   metadata
       hosts : [" cn - hangzhou . log . aliyuncs . com : 10012 "]
       username : "< youruserna  me >"
       password : "< yourpasswo  rd >"
       ssl . certificat  e_authorit  ies :

      #  message   topic   selection  +  partitioni  ng
       topic : ' test - logstore - 1 '
       partition . round_robi  n :
       reachable_  only :  false

       required_a  cks :  1
       compressio  n :  gzip
       max_messag  e_bytes :  1000000
```

  By default, the Beats software outputs the JSON type logs to Kafka. You can also create a JSON type index for the content field. For more information, see *JSON type*. The following is a log example.



- Example 2: Use Collectd to collect data to Log Service

  Collectd is a daemon used to collect the performance metrics of system and application on regular basis. You can also use Collectd to output collected data to Kafka. For more information, see *Write Kafka plugin*.

  If you want to output the collected data to Kafka by using Collectd, you need to install the Kafka plugin and related dependency. In the Centos operating system, you can directly run the `sudo yum install collectd-write_kafka` command to install the plugin. For more information about the Red-Hat Package Manager (RPM) resources, see *Collectd-write_kafka*.

  The following is an example configuration template:

```
<Plugin write_kafka>
  Property "metadata.broker.list" "cn-hangzhou.log.aliyuncs.com:10012"
  Property "security.protocol" "sasl_ssl"
  Property "sasl.mechanism" "PLAIN"
  Property "sasl.username" "<yourusername>"
  Property "sasl.password" "<yourpassword>"
  Property "broker.address.family" "v4"
  <Topic "test-logstore-1">
    Format JSON
    Key "content"
  </Topic>
</Plugin>
```

  This example configuration template shows that the format of the data output to be sent to Kafka is set to JSON. In addition to JSON, Collectd also supports the

Command and Graphite formats. For more information, see *Collectd configuration document*.

If you use JSON format, you can create a JSON type index for the content field. For more information, see *JOSN index types*. The following is a log example.



- Example 3: Use Telegraf to collecte data to Log Service

  Telegraf is a sub-project of InfluxData. Telegraf is the agent for collecting, processing, and aggregating metrics. It is compiled by the Go language, and designed to use less memory resource. Telegraf can be used to build services and collect the metrics of a third party component through plugins. In addtion, Telegraf has the integration function, which can obtain metrics in the system where Telegraf runs, obtain metrics from a third party API, and even listen for metrics through StatsD and Kafka consumer services.

  Telegraf can output data to Kafka. Therefore, you only need to modify the configuration file to use Telegraf to collect data to Log Service.

```
[[ outputs . kafka ]]
       ##  URLs   of   kafka   brokers
        brokers  = [" cn - hangzhou . log . aliyuncs . com : 10012
 "]
       ##  Kafka   topic   for   producer   messages
        topic  = " test - logstore - 1 "
        routing_ke  y  = " content "

       ##  Compressio  nCodec   represents   the   various
 compressio  n   codecs   recognized   by
```

```
      ##  Kafka   in   messages .
      ##  0  :  No   compressio  n
      ##  1  :  Gzip   compressio  n
      ##  2  :  Snappy   compressio  n
      ##  3  :  LZ4   compressio  n
       compressio  n_codec  =  1

      ##  Optional   TLS   Config   tls_ca  = "/ etc / ssl / certs
/ ca - bundle . crt "
      #  tls_cert  = "/ etc / telegraf / cert . pem " #  tls_key  =
"/ etc / telegraf / key . pem "
      ##  Use   TLS   but   skip   chain  &  host   verificati  on

      #  insecure_s  kip_verify  =  false

      ##  Optional   SASL   Config
       sasl_usern  ame  = "< youruserna  me >"
       sasl_passw  ord  = "< yourpasswo  rd >"

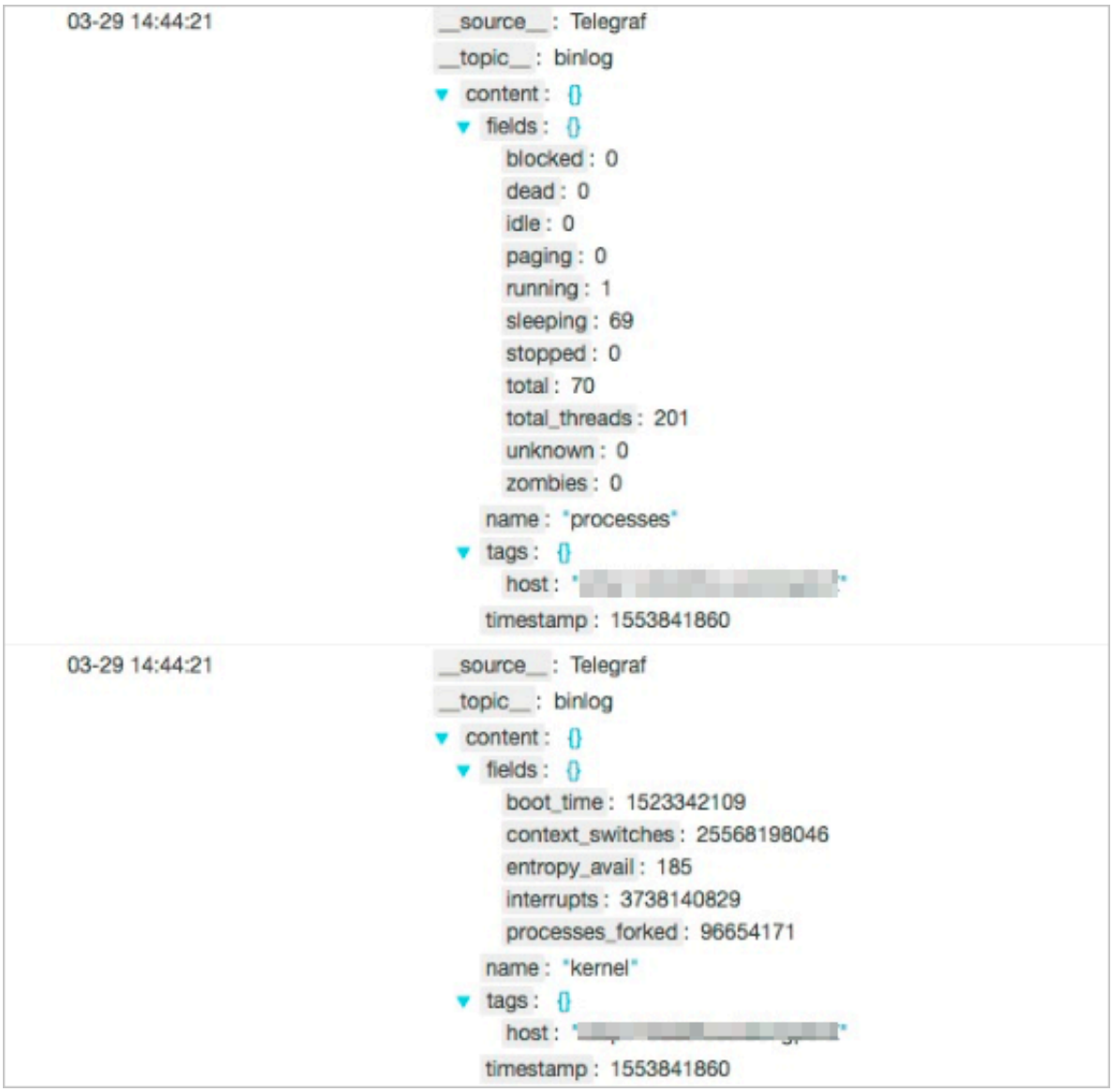
      ##  Data   format   to   output .
      ##  https :// github . com / influxdata / telegraf / blob /
master / docs / DATA_FORMA  TS_OUTPUT . md
       data_forma  t  = " json "
```

Note:

You must set a valid `tls_ca` directory for Telegraf. You can use the default root certificate. The typical root certificate directory in a Linux environment is `/etc/ssl/certs/ca-bundle.crt`.

This example configuration template indicates that the format of the data output to Kafka is set as JSON. Besides JSON, Graphite, Carbon2, and other formats are also supported. For more information, see *Telegraf output data format*.

If you use the JSON format, you can create a JSON type index for the content field. For more information, see *JOSN index types*. The following is a log example.



- Example 4: Use Fluentd to collect data to Log Service

  Fluentd is an open-source data collector for Unified Logging Layer. Fluentd allows you to unify data collection and usage for overall better optimization. Fluentd

a Cloud Native Computing Foundation (CNCF) member project, and follows the Apache 2 License protocol.

Many plugins for inputting, processing, and outputting data are available for Fluentd. Specifically, the *Kafka* plugin can help Fluentd output data to Kafka. You only need to install and configure this plugin. For more information, see *fluent-plugin-kafka*.

The following is an example configuration temple:

```
< match  **>
       @ type   kafka

       #  Brokers :  you   can   choose   either   brokers   or
 zookeeper .
        brokers        cn - hangzhou . log . aliyuncs . com : 10012

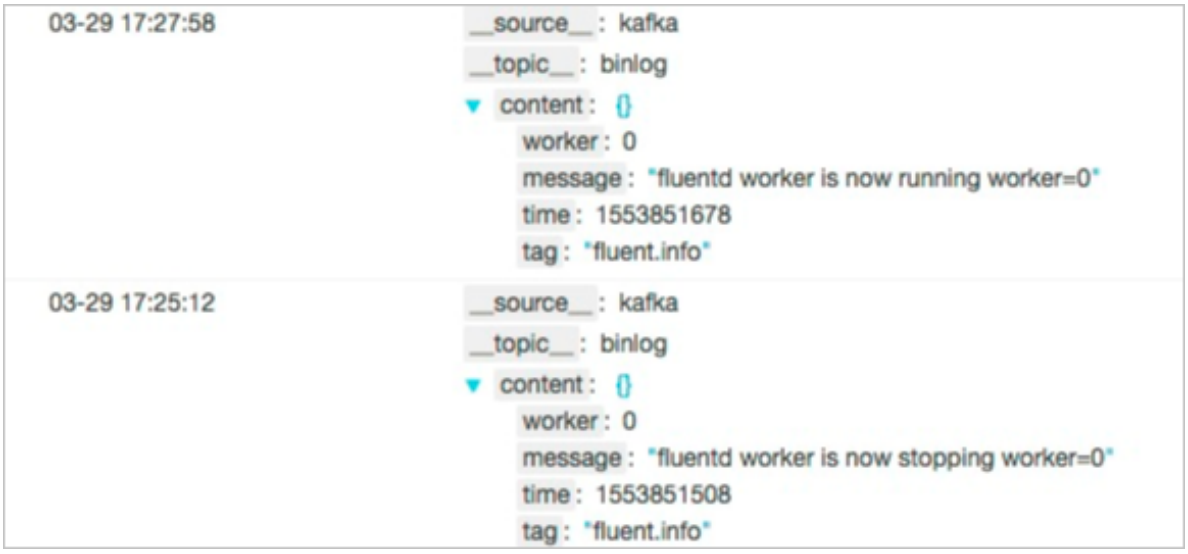

        default_to  pic   test - logstore - 1
        default_me  ssage_key   content
        output_dat  a_type   json
        output_inc  lude_tag   true
        output_inc  lude_time   true
        sasl_over_  ssl   true
        username  < youruserna  me >
        password  < yourpasswo  rd >
        ssl_ca_cer  ts_from_sy  stem   true

       #  ruby - kafka   producer   options
        max_send_r  etries   10000
        required_a  cks   1
        compressio  n_codec   gzip
```

```
        </ match >
```

This example configuration template shows that the format of the data output to be sent to Kafka is set to JSON. In addition to JSON, more formats are also supported. For more information, see *Fluentd formatter*.

If you use the JSON format, you can create a JSON type index for the content field. For more information, see *JOSN index types*. The following is a log example.



- Example 5: Use Logstash to collect data to Log Service

  Logstash is an open-source engine for collecting data in real time. With Logstash, you can dynamically collect data from different sources, process the data (for example, filter or convert the data), and output the result. You can analyze the data further based on the output result.

  The Kafka output plugin is built in Logstash, allowing you to directly configure Logstash to collect data to Log Service. However, you must configure the jass files for the SSL certificate and SASL because the Kafka protocol uses the SASL_SSL connection protocol.

  1. Create a jaas file, and then save it to a target directory, such as */ etc / kafka / kafka_clie nt_jaas . conf .*

     ```
     KafkaClien t {
             org . apache . kafka . common . security . plain .
     PlainLogin Module required
             username ="< youruserna me >"
             password ="< yourpasswo rd >";
     ```

```
        };
```

2. Set the SSL trusted certificate, and then save it to a target directory, such as `/etc/kafka/client-root.truststore.jks`.

   The domain names in Log Service all have trusted certificates. You only need to download the GlobalSign Root CA, and save the root certificate of base 64 code to any directory (for example, `/etc/kafka/ca-root`). Then run a keytool command to generate a JKS format file. When you generate a JKS format file for the first time, you need to set a password.

```
keytool -keystore client.truststore.jks -alias root
-import -file /etc/kafka/ca-root
```

3. Configure the Logstash as follows:

```
input { stdin { } }

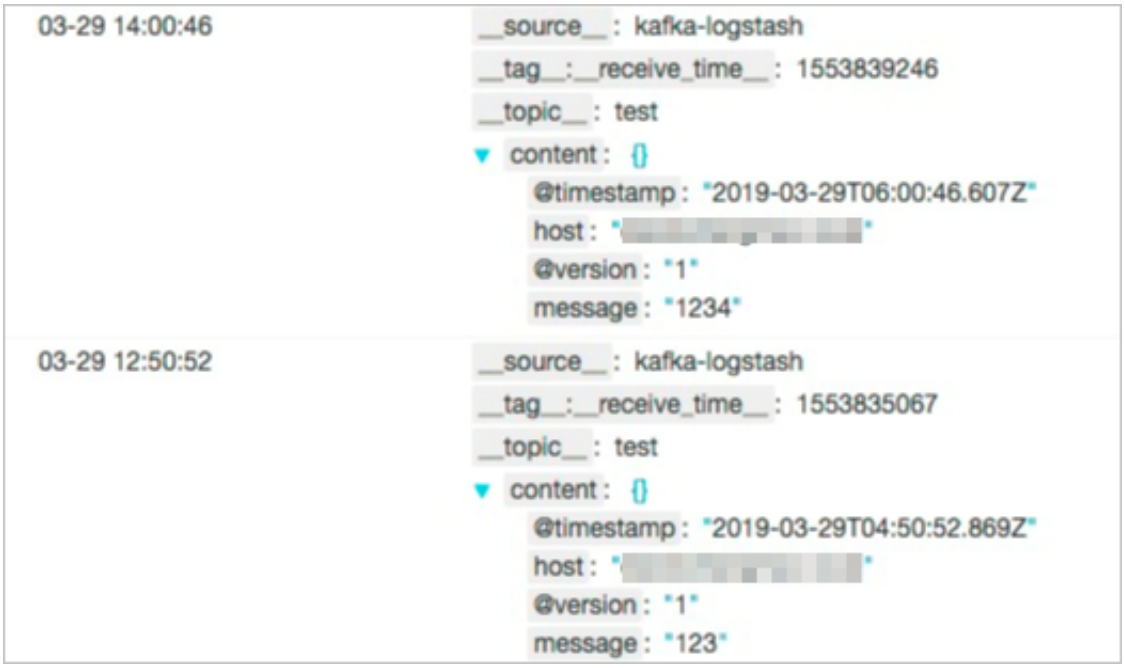
        output {
        stdout { codec => rubydebug }
        kafka {
        topic_id => "test-logstore-1"
        bootstrap_servers => "cn-hangzhou.log.aliyuncs
.com:10012"
        security_protocol => "SASL_SSL"
        ssl_truststore_location => "/etc/client-root.
truststore.jks"
        ssl_truststore_password => "123456"
        jaas_path => "/etc/kafka_client_jaas.conf"
        sasl_mechanism => "PLAIN"
        codec => "json"
        client_id => "kafka-logstash"
       }
       }
```

 **Note:**

This example configuration template is only for a connectivity test. In actual applications, we recommend that you remove the stdout output settings.

This example configuration template indicates that the format of the data output to Kafka is set as JSON. Other formats in addition to JSON are supported. For more information, see *Logstash Codec*.

If you use the JSON format, you can create a JSON type index for the content field. For more information, see *JOSN index types*. The following is a log example.



5.3 Use the Syslog protocol to collect data to Log Service

This topic describes how to use the Syslog protocol to collect data to Log Service without using the collection agent to forward collected data. The following sections contain information about the limits, required configurations, sample logs, and application examples detailing possible uses of the Syslog protocol.

Limits

- The Syslog protocol must be the standard *RFC5424* protocol.
- Each log can be up to 64 KB.
- You must use TLS 1.2 (based on TCP) to guarantee data transmission security.

Required configurations

- You must associate Syslog with Log Service.

  Specifically, the port number of Syslog (10009) and the service endpoint of the target Project are required. For example, `cn - hangzhou - intranet . log . aliyuncs . com : 10009` associates Syslog with a Project. The service endpoint of a Project in Log Service is a URL that is used to access the Project. For more information, see *Service endpoint*.

- You must set the `STRUCTURED - DATA` field (contains the parameters related to Log Service) of Syslog as follows.

Configuration	Description	Example
`STRUCTURED - DATA`	This field requires a name . This name must be Logservice.	Logservice
`Project`	The user name is mapped to a project name.	test-project-1
`Logstore`	The topic is mapped to a Logstore name. You must create a Logstore before you can collect data.	test-logstore-1
`access - key - id`	Indicates your AccessKey ID. We recommend that you use the AK of a RAM user. For more information, see *Grant a RAM user the permissions to access Log Service*.	<yourAccessKeyId>

Configuration	Description	Example
`access-key-secret`	Indicates your AccessKey Secret. We recommend that you use the AK of a RAM user. For more information, see *Grant a RAM user the permissions to access Log Service*.	<yourAccessKeySecret>

Sample logs

Log Service automatically resolves the received Syslog data. Furthermore, Log Service removes the Logservice field to avoid the AccessKey from being leaked. The following are the fields uploaded to Log Service (for more information about fields, see *RFC5424 protocol*).

Field	Description
__source__	Indicates the hostname field in Syslog.
__topic__	This field is fixed as syslog-forwarder.
__facility__	Indicates the facility (device or module) information in Syslog.
__program__	Indicates a process name.
__serverity__	Indicates the log severity.
__priority__	Indicate the log priority.
__unixtimestamp__	Indicates a log timestamp in seconds.
content	Indicates the msg field in Syslog.

Application examples

For each of the following application examples, the required parameters are set as follows to have the Syslog logs be directly collected on a host to Log Service:

- The project in Log Service is set as `test-project-1`.
- The Logstore in Log Service is set as `test-logstore-1`.

- The region where the project is located is set as `cn - hangzhou`.
- The AccessKey ID of a RAM user that has the write permissions is set as `< yourAccess KeyId >`, and the AccessKey Secret of the RAM user is set as `< yourAccess KeySecret >`.
- Application example 1: Use Rsyslog to forward system logs to Log Service

  Rsyslog is installed in a Linux host to process system logs. You can use Rsyslog to forward the system logs to Log Service. The Rsyslog configuration file format varies by Rsyslog versions. You can run the `man rsyslogd` command to view which Rsyslog version is being used.

  Note:

  Make sure that you have installed the gnutls module in Rsyslog. If the module is not installed in Rsyslog, you can run the `sudo apt - get install rsyslog - gnutls` command or the `sudo yum install rsyslog - gnutls` command to install the module.

  - Rsyslog V8 and later

    `$ DefaultNet streamDriv erCAFile` indicates the directory where the system root certificate is located.

```
#  Setup   disk   assisted   queues
         $ WorkDirect  ory  / var / spool / rsyslog  #  where   to
   place   spool   files
         $ ActionQueu  eFileName   fwdRule1      #  unique   name
   prefix   for   spool   files
         $ ActionQueu  eMaxDiskSp  ace   1g       #  1gb   space
   limit  ( use   as   much   as   possible )
         $ ActionQueu  eSaveOnShu  tdown   on      #  save
 messages   to   disk   on   shutdown
         $ ActionQueu  eType   LinkedList        #  run
 asynchrono  usly
         $ ActionResu  meRetryCou  nt  - 1        #  infinite
 retries   if   host   is   down
         $ ActionSend  TCPRebindI  nterval   100   #  close   and
   re - open   the   connection   to   the   remote   host   every
   100   of   messages   sent .
         # RsyslogGnu  TLS   set   to   default   ca   path
         $ DefaultNet  streamDriv  erCAFile  / etc / ssl / certs /
 ca - bundle . crt
          template ( name =" LogService  Format "  type =" string
 "
          string ="<% pri %> 1  % timestamp ::: date - rfc3339 %
 % HOSTNAME % % app - name % % procid % % msgid % [ logservice
 project =\" test - project - 1 \"  logstore =\" test - logstore -
 1 \"  access - key - id =\"< yourAccess  KeyId >\"  access - key
 - secret =\"< yourAccess  KeySecret >\"] % msg %\ n "
         )
```

```
        # Send messages to Loggly over TCP using
the template .
        action ( type =" omfwd " protocol =" tcp " target
=" cn - hangzhou . log . aliyuncs . com " port =" 10009 "
template =" LogService Format " StreamDriv er =" gtls "
StreamDriv erMode =" 1 " StreamDriv erAuthMode =" x509 / name
" StreamDriv erPermitte dPeers ="*. log . aliyuncs . com ")
```

- **Rsyslog V7 and earlier**

  `$ DefaultNet streamDriv erCAFile` **indicates the directory where the system root certificate is located.**

```
# Setup disk assisted queues
        $ WorkDirect ory / var / spool / rsyslog        #
where to place spool files
        $ ActionQueu eFileName fwdRule1            # unique
  name prefix for spool files $ ActionQueu eMaxDiskSp
ace 1g             # 1gb space limit ( use as much
  as possible ) $ ActionQueu eSaveOnShu tdown on
# save messages to disk on shutdown
        $ ActionQueu eType LinkedList              # run
asynchrono usly
        $ ActionResu meRetryCou nt - 1              #
infinite retries if host is down $ ActionSend
TCPRebindI nterval 100          # close and re - open
the connection to the remote host every 100 of
messages sent .
        # RsyslogGnu TLS set to default ca path
        $ DefaultNet streamDriv erCAFile / etc / ssl / certs /
ca - bundle . crt
        $ ActionSend StreamDriv er gtls
        $ ActionSend StreamDriv erMode 1
        $ ActionSend StreamDriv erAuthMode x509 / name
        $ ActionSend StreamDriv erPermitte dPeer cn -
hangzhou . log . aliyuncs . com
        template ( name =" LogService Format " type =" string
" string ="<% pri %> 1 % timestamp ::: date - rfc3339 % %
HOSTNAME % % app - name % % procid % % msgid % [ logservice
project =\" test - project - 1 \" logstore =\" test - logstore -
1 \" access - key - id =\"< yourAccess KeyId >\" access - key
- secret =\"< yourAccess KeySecret >\"] % msg %\ n ")
        *.* action ( type =" omfwd " protocol =" tcp " target
=" cn - hangzhou . log . aliyuncs . com " port =" 10009 "
template =" LogService Format ")
```

1. **Select the required version of Rsyslog, set either of the preceding configuration template according to your needs, and add the template to the end of the Rsyslog configuration file.**

   Note:

Generally, the directory to store the Rsyslog configuration file is `/etc/rsyslog.conf`.

2. Run the `sudo service rsyslog restart` and `sudo /etc/init.d/syslog-ng restart` commands or the `systemctl restart rsyslog` command to restart Rsyslog.
3. If no system logs are generated, run the `logger` command to generate a few logs for test. For example, `logger hello world!`.

- Example 2: Use Syslog-ng to forward system logs to Log Service

Syslog-ng is an open-source daemon for log management that helps Unix and Unix-like operating systems support the Syslog protocol. You can run the `sudo yum install syslog-ng` command or the `sudo apt-get install syslog-ng` command to install Syslog-ng.

```
### Syslog-ng Logging Config for LogService ###
        template LogService Format {
        template ("<${ PRI }> 1 ${ ISODATE } ${ HOST :--} ${
PROGRAM :--} ${ PID :--} ${ MSGID :--} [ logservice project =\"
test-project-1 \" logstore =\" test-logstore-1 \" access
-key-id =\"< yourAccess KeyId >\" access-key-secret =\"<
yourAccess KeySecret >\"] $ MSG \ n "); template_e scape ( no
);
       };
        destinatio n d_logservi ce {
        tcp (" cn-hangzhou . log . aliyuncs . com " port ( 10009
)
        tls ( peer-verify ( required-untrusted ))
        template ( LogService Format ));
       };
        log {
        source ( s_src ); # default use s_src
        destinatio n ( d_logservi ce );
       };
       ### END Syslog-ng Logging Config for LogService
 ###
```

Note:
By default, Rsyslog is installed in an ECS instance to process system logs. If you want to use Syslog-ng, you need to uninstall Syslog-ng first because Rsyslog and Syslog-ng cannot work together.

1. Set the preceding configuration template as needed, and add the template to the end of the Syslog-ng configuration file.

Note:

The typical directory to store the Syslog-ng configuration file is `/etc/syslog-ng/syslog-ng.conf`.

2. Run the `sudo /etc/init.d/syslog-ng restart`, `sudo service syslog-ng restart`, and `sudo systemctl restart syslog-ng` commands to restart Syslog-ng.
3. If no system logs are generated, run the `logger` command to generate a few logs for test. For example, `logger hello world!`.

Troubleshoot exceptions

- Manually upload a log to check the network connectivity

  You can use the ncat command to simulate uploading system logs. This method can check the network connectivity and whether the AccessKey has the permissions to upload logs. If ncat is not installed in your server, you can run the `sudo yum install nmap-ncat` command to install it. For example, the following command is run to send a system log to Log Service:

```
[root@iZbp145dd9fccuidd7g****~]# ncat --ssl cn-hangzhou.log.aliyuncs.com 10009
<34>1 2019-03-28T03:00:15.003Z mymachine.example.com su - ID47 [logservice project="test-project-1" logstore="test-logstore-1" access-key-id="<yourAccessKeyId>" access-key-secret="<yourAccessKeySecret>"] this is a test message
```

  The project in Log Service is set as `test-project-1` and the Logstore is set as `test-logstore-1`. The region where the project is located is set as cn-hangzhou. The AccessKey ID of a RAM user that has the write permissions is set as `<yourAccessKeyId>`, and the AccessKey Secret of the RAM user is set as `<yourAccessKeySecret>`.

  Note:

  - When you run the ncat command to upload a log, you must specify your current time for the log in ISO 8601 format. For example, if you upload a log at 2019-03-28T11:00:15.003, you must convert it to 2019-03-28T03:00:15.003Z before you add it to the ncat command.

- The ncat commands cannot identify network connection interrupt. Therefore, you need to enter messages and press Enter within 30 seconds after you run a ncat command.

After you run the command, you can preview logs in the Log Service console. For more information, see *Preview log data*.

- Diagnose log collection errors

  If you fail to manually upload logs, you can diagnose log collection errors to view error information. For more information, see *Diagnose collection errors*.

- Check Rsyslog error logs

  Rsyslog logs are saved in the `/ var / log / message` directory be default. You can run a vim command to view the directory.

  - Rsyslog error

    ```
    dlopen : / usr / lib64 / rsyslog / lmnsd_gtls . so :  cannot
      open   shared   object   file :  No   such   file   or
    directory
    ```

    This error occurred because the gnutls module is not installed. You can run the `sudo apt - get install rsyslog - gnutls` command or the `sudo yum install rsyslog - gnutls` command to install and restart Rsyslog.

  - Rsyslog error

    ```
    unexpected   GnuTLS   error   - 53   -   this   could   be   caused
      by   a   broken   connection .  GnuTLS   reports : Error   in
    the   push   function
    ```

    This error occurred because the TCP connection was idle for long time and then was forcibly shutdown. Rsyslog can automatically connect to Log Service.

- Check Syslog-ng error logs

  Syslog-ng logs are saved in the Journal logs by default. You can run the `systemctl status syslog - ng . service` and `journalctl - xe` command to check detailed logs.

  Syslog-ng initialization error

  ```
  Job   for   syslog - ng . service   failed   because   the
    control   process   exited   with   error   code .  See  "
  ```

```
systemctl status syslog-ng.service" and "journalctl -xe" for details
```

You must check that the Syslog-ng configuration file format is valid and that the settings in the file do not conflict (for example, multiple internal(); are not set).

5.4 Logstash

5.4.1 Install Logstash

This topic describes how to install Logstash to upload log data to Log Service.

Context

Logstash is an open-source software for collecting data. You can install the logstash-output-logservice plugin to upload data to Log Service. For more information, see *Logstash plugin* in GitHub.

Procedure

1. Install Java
   a. Download the installation package.

      Go to the *Java official website* to download JDK for installation.

   b. Sets the environment variable.

      Add or modify environment variables in advanced system settings.

      - PATH: *C:\Program Files\Java\jdk1.8.0_73\bin*
      - CLASSPATH: *C:\Program Files\Java\jdk1.8.0_73\lib;*
        C:\Program Files\Java\jdk1.8.0_73\lib\tools.jar
      - JAVA_HOME: *C:\Program Files\Java\jdk1.8.0_73*

   c. Perform verification.

      Run `PowerShell` or `cmd.exe` for verification.

```
PS C:\Users\Administrator> java -version
java version "1.8.0_73"
Java(TM) SE Runtime Environment (build 1.8.0_73-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.73-b02, mixed mode)
PS C:\Users\Administrator> javac -version
```

```
javac   1 . 8 . 0_73
```

2. Install Logstash

   a. Download the installation package from the official website *Logstash*.

   > Note:
   >
   > · We recommend that you download v 5.0 or any later versions of Logstash.
   > · Logstash v 6.4.3 can be installed can operates normally in the following operating systems: Mac OS 10.14.1, Windows 7, and Linux (CentOS 7).

   b. Install Logstash.

3. Install the plug-in used by Logstash to write logs to Log Service

   Install the plug-in online or offline based on the network environment where the machine resides.

   · Online installation

     The plug-in is hosted by RubyGems. For more information, see *here* .

     Run `PowerShell` or `cmd . exe` to go to the Logstash installation directory.

     ```
     PS   C :\ logstash - 6 . 4 . 3 > .\ bin \ plugin   install
     logstash - output - logservice
     ```

   · Offline installation

     Download from the official website. Go to the *logstash-output-logservice* page and click Download in the lower-right corner.

     If the machine from which logs are collected cannot access the Internet, copy the downloaded gem package to your local directory. Run *PowerShell* or *cmd . exe* to go to the Logstash installation directory. Perform the following command to install lLogstash:

     ```
     PS   C :\ logstash - 6 . 4 . 3 > .\ bin \ plugin   install   C :\
     logstash - 6 . 4 . 3 \ logstash - output - logservice - 0 . 4 . 0
     . gem
     ```

   · Perform verification.

     ```
     PS   C :\ logstash - 6 . 4 . 3 > .\ bin \ plugin   list
     ```

     Verify that logstash-output-logservice exists in the installed plug-in list of the machine.

5.4.2 Set Logstash as a Windows service

When logstash.bat is started in PowerShell, the Logstash process is working in the frontend. Logstash is generally used for testing configurations and debugging collections. Therefore, we recommend that you set Logstash as a Windows service after the debugging is passed so as to enable Logstash to work in the backend and start automatically when power-on.

Besides setting Logstash as a Windows service, you can also start, stop, modify, and delete the service by using command lines. For more information about how to use NSSM, see *NSSM official document*.

Add Logstash as a Windows service

This operation is generally performed when Logstash is deployed for the first time. If Logstash has been added, skip this step.

Run the following command to add Logstash as a Windows service.

- 32 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win32 \ nssm .
exe   install   logstash  " C :\ logstash - 2 . 2 . 2 - win \ bin
\ logstash . bat " " agent  - f   C :\ logstash - 2 . 2 . 2 - win \
conf "
```

- 64. -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win64 \ nssm .
exe   install   logstash  " C :\ logstash - 2 . 2 . 2 - win \ bin
\ logstash . bat " " agent  - f   C :\ logstash - 2 . 2 . 2 - win \
conf "
```

Start the service

If the configuration file in the Logstash `conf` directory is updated, stop the Logstash service and then start it again.

Run the following command to start the service.

- 32 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win32 \ nssm .
exe   start   logstash
```

- 64 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win64 \ nssm .
exe   start   logstash
```

Stop the service

Run the following command to stop the service.

- 32 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win32 \ nssm .
exe   stop   logstash
```

- 64 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win64 \ nssm .
exe   stop   logstash
```

Modify the service

Run the following command to modify the service.

- 32 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win32 \ nssm .
exe   edit   logstash
```

- 64 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win64 \ nssm .
exe   edit   logstash
```

Delete the service

Run the following command to delete the service.

- 32 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win32 \ nssm .
exe   remove   logstash
```

- 64 -bit system

```
C :\ logstash - 2 . 2 . 2 - win \ nssm - 2 . 24 \ win64 \ nssm .
exe   remove   logstash
```

5.4.3 Create Logstash collection configurations

Context

Related plug-ins

- logstash-input-file

  This plug-in is used to collect log files in tail mode. For more information, see *logstash-input-file*.

   Note:
  path indicates the file path, which must use UNIX separators, for example, `C :/ test / multiline /*. log` . Otherwise, fuzzy match is not supported.

- logstash-output-logservice

  This plug-in is used to output the logs collected by the logstash-input-file plug-in to Log Service.

Parameters	Description
endpoint	Log Service endpoint. Example: `http :// regionid . example . com` . For more information, see Log Service endpoint.
project	The project name of Log Service.
logstore	The Logstore name.
topic	The log topic name. The default value is null.
source	The log source. If this parameter is set to null, the IP address of the current machine is used as the log source . Otherwise, the log source is subject to the specified parameter value.
access_key_id	The AccessKey ID of the Alibaba Cloud account.
access_key_secret	The AccessKey Secret of the Alibaba Cloud account.

Parameters	Description
max_send_retry	The maximum number of retries performed when data packets cannot be sent to Log Service because of an exception. Data packets with retry failures are discarded. The retry interval is 200 ms.

Procedure

1. Create collection configurations

   Create a configuration file in the `C:\logstash-2.2.2-win\conf\` directory and then restart Logstash to apply the file.

   You can create a configuration file for each log type. The file name format is `*.conf`. For easier management, we recommend that you create all the configuration files in the `C:\logstash-2.2.2-win\conf\` directory.

    Note:
   The configuration file must be encoded in UTF-8 format without BOM. You can use Notepad++ to modify the file encoding format.

   - IIS logs

     For more information, see *Use Logstash to collect IIS logs*.

   - CSV logs

     Use the system time of log collection as the log uploaded time. For more information, see CSV log configuration.

   - Logs with built-in time

     Take CSV log format as an example. Use the time in the log content as the log uploaded time. For more information, see *Use Logstash to collect CSV logs*.

   - General logs

     By default, the system time of log collection is used as the log uploaded time. Log fields are not parsed. Single-line logs and multiline logs are supported. For more information, see *Use Logstash to collect other logs*.

2. Verify configuration syntax

   a. Run `PowerShell` or `cmd . exe` to go to the Logstash installation directory:

   ```
   PS C :\ logstash - 2 . 2 . 2 - win \ bin > .\ logstash . bat
   agent -- configtest -- config C :\ logstash - 2 . 2 . 2 - win
   \ conf \ iis_log . conf
   ```

   b. Modify the collection configuration file. Temporarily add a line of rubydebug configuration in the output phase to output the collection results to the console. Set the type field as per your needs.

   ```
   output {
   If [ type ] = "***"{
     stdout { codec => rubydebug }
     logservice {

    }
   ```

   c. Run `PowerShell` or `cmd . exe` to go to the Logstash installation directory and start the process:

   ```
   PS C :\ logstash - 2 . 2 . 2 - win \ bin > .\ logstash . bat
   agent - f C :\ logstash - 2 . 2 . 2 - win \ conf
   ```

   After the verification, end the *logstash . bat* process and delete the temporary configuration item rubydebug.

What's next

When *logstash . bat* is started in PowerShell, the Logstash process is working in the frontend. Logstash is generally used for testing configurations and debugging collections. Therefore, we recommend that you set Logstash as a Windows service after the debugging is passed so as to enable Logstash to work in the backend and start automatically when power-on. For how to set Logstash as a Windows service, see *Set Logstash as a Windows service*.

5.4.4 Advanced functions

Logstash provides *multiple plug-ins* to meet personalized requirements. For example:

- *grok*: Structurally parses logs into multiple fields by using regular expressions.
- *json_lines* and *json*: Structurally parses JSON logs.
- *date*: Parses and converts the date and time fields of logs.
- *multiline*: Customizes complex types of multiline logs.

- *kv*: Structurally parses logs of key-value pair type.

5.4.5 Logstash error processing

If you encounter the following collection errors when using Logstash to collect logs, follow the corresponding suggestions and process the errors.

If you encounter the following collection errors when using Logstash to collect logs, follow the corresponding suggestions and process the errors.

- Data with garbled characters in Log Service

  Logstash supports UTF-8 file encoding by default. Check whether input files are correctly encoded or not.

- Error message in the console

  The error `io / console not supported ; tty will not be manipulate d` is prompted in the console. However, the error does not affect the functions and can be ignored.

  If other errors occur, we recommend that you search Google or Logstash forums for help.

5.5 SDK collection

5.5.1 Producer Library

LogHub Producer Library is a LogHub class library written for high-concurrency Java applications. Producer Library and *Consumer Library* are the read and write packaging for LogHub to lower the threshold for data collection and consumption.

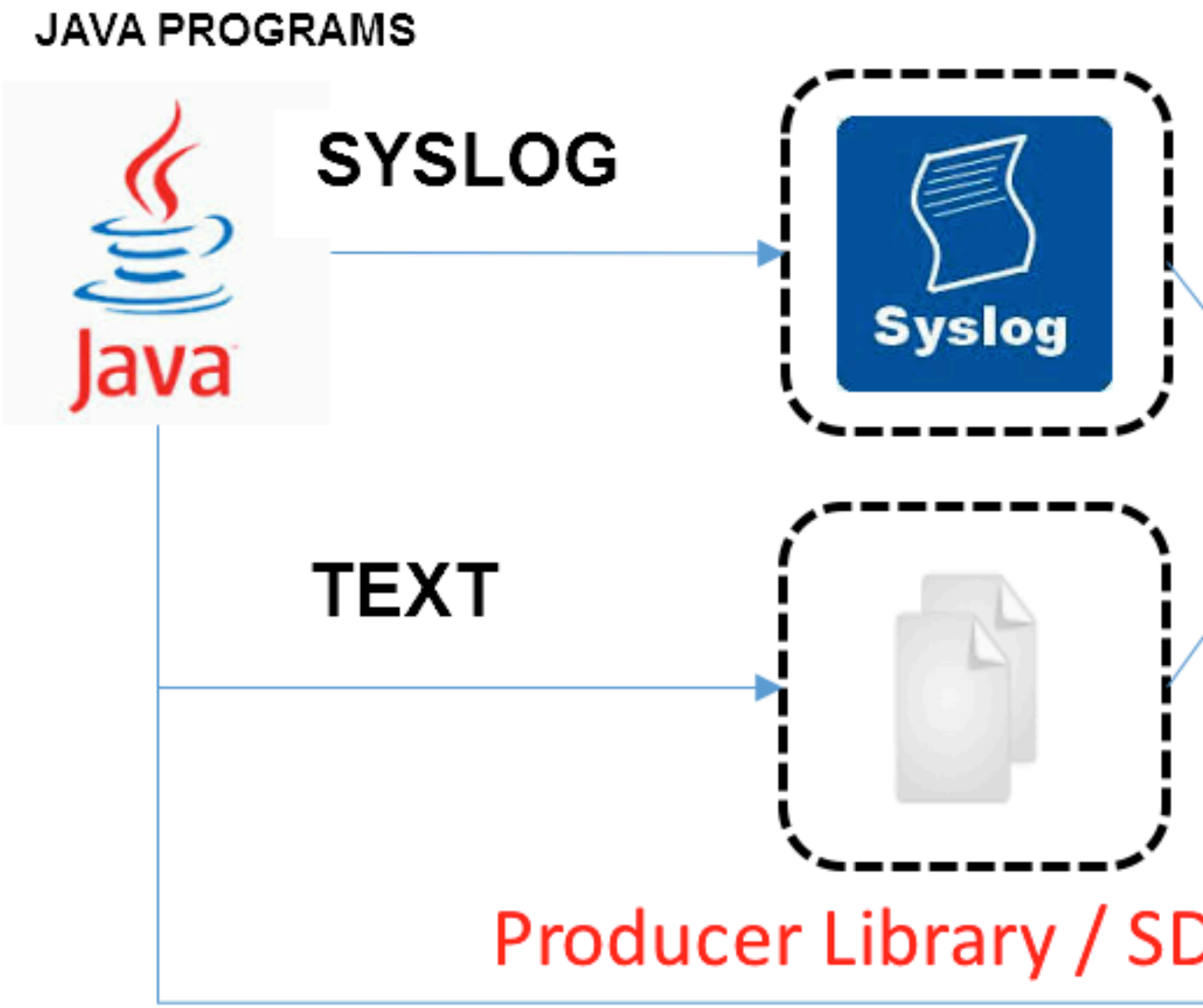
Function features

- Provides an asynchronous send interface to guarantee the thread security.
- Configurations of multiple projects can be added.
- The number of network I/O threads used for sending logs can be configured.
- The number and size of logs of a merged package can be configured.
- The memory usage is controllable. When the memory usage reaches your configured threshold value, the send interface of producer is blocked until idle memory is available.

Function advantages

- **Logs collected from the client are not flushed into the disk. Data is directly sent to Log Service by using the network after being generated.**
- **High concurrency write operations on the client. For example, more than one hundred write operations are performed in one second.**
- **Client computing logically separated from I/O. Printing logs does not affect the computing time used.**

In the preceding scenarios, Producer Library simplifies your program developmen t steps, aggregates write requests in batches, and sends the requests to the LogHub server asynchronously. During the process, you can configure the parameters for aggregation in batches and the logic to process server exception.



Compare the preceding access methods:

Access method	Advantages/disadvantages	Scenario
Log flushed into the disk + Logtail	Log collection decoupled from logging, no need to modify the code.	Common scenarios

Access method	Advantages/disadvantages	Scenario
Syslog + Logtail	Good performance (80 MB/s). Logs are not flushed into the disk. The syslog protocol must be supported.	Syslog scenarios.
SDK direct transmission	Not flushed into the disk, and directly sent to the server. Switching between the network I/O and program I/O must be properly processed.	Logs are not flushed into the disk.
Producer Library	Not flushed into the disk, asynchronously merged and sent to the server, with good throughput.	Logs are not flushed into the disk and the client QPS is high.

Procedure

- *Java Producer*
- *Log4J1. Log4J1.XAppender (based on Java Producer)*
- *Log4J2. XAppender (based on Java Producer)*
- *LogBack Appender (based on Java Producer)*
- *C Producer*
- *C Producer Lite*

5.5.2 Log4j Appender

Log4j is an open-source project of Apache, which allows you to set the log output destination to console, file, GUI component, socket server, NT event recorder, or UNIX Syslog daemon. You can also set the output format and level of each log to control log generation with a finer granularity. These configurations can be performed flexibly by using a configuration file without modifying application codes.

Alibaba Cloud Log4j Appender allows you to set the log output destination to Alibaba Cloud Log Service. For more information about download link and user guide, refer to *Github*.

5.5.3 C Producer Library

Besides the Producer Library of Java version, LogHub also supports the Producer Library and Producer Lite Library of the C version, which provides you with a simple

and high-performance one-stop log collection solution across platforms and with low consumption of resources.

For the GitHub project address, see:

- *C Producer Library (recommended for servers)*
- *C Producer Lite Library (recommended for IOT and smart devices)*

5.6 Common log formats

5.6.1 Apache log

The Apache log format and directory are generally in the */ etc / apache2 / httpd . conf* configuration file.

Log format

By default, the Apache log configuration file defines two print formats: combined format and common format. You can also create your own customized log print format as needed.

- Combined format:

```
LogFormat  "% h  % l  % u  % t  \"% r \" %> s  % b  \"%{ Referer }
i \" \"%{ User - Agent } i \""  combined
```

- Common format:

```
LogFormat  "% h  % l  % u  % t  \"% r \" %> s  % b "
```

- Customized format:

```
LogFormat  "% h  % l  % u  % t  \"% r \" %> s  % b  \"%{ Referer }
i \" \"%{ User - Agent } i \" % D  % f  % k  % p  % q  % R  % T  %
I  % O "  customized
```

You need to specify the print format, log file path, and log name of the current log in the Apache log configuration file. For example, the following log configuration file indicates that combined print format is used, and the log path and name is displayed as /var/log/apache2/access_log.

```
CustomLog  "/ var / log / apache2 / access_log "  combined
```

Field description

Format	Key name	Description
%a	client_addr	Client IP address.

%A	local_addr	Local private IP address.
%b	response_size_bytes	Size of response in bytes. When the size of response is null, this field is a hyphen (-).
%B	response_bytes	Size of response in bytes. When the size of response is null, this field is a hyphen (-).
%D	request_time_msec	Request time, in microseconds.
%h	remote_addr	Remote hostname.
%H	request_protocol_supple	Request protocol.
%l	remote_ident	Client log name from identd.
%m	request_method_supple	Request method.
%p	remote_port	Server port.
%P	child_process	Child process ID.
%q	request_query	Query string. If no query string exists, this field is an empty string.
"%r"	request	Request content, including the request method name, address, and HTTP protocol.
%s	status	HTTP status code.
%>s	status	Final HTTP status code.
%f	filename	Filename.
%k	keep_alive	Number of keepalive requests.
%R	response_handler	Handler on the server.
%t	time_local	Server time.
%T	request_time_sec	Request time, in seconds.
%u	remote_user	Client username.
%U	request_uri_supple	Requested URL path. No query is included in the path.
%v	server_name	Server name.
%V	server_name_canonical	Server name conforming to the UseCanonicalName setting.

%I	bytes_received	Number of bytes received by the server. You must enable the mod_logio module.
%O	bytes_sent	Number of bytes sent by the server . You must enable the mod_logio module.
"%{User-Agent}i"	http_user_agent	Client information.
"%{Rererer}i"	http_referer	Source page.

Sample log

```
192 . 168 . 1 . 2  - - [ 02 / Feb / 2016 : 17 : 44 : 13  + 0800
] " GET  / favicon . ico  HTTP / 1 . 1 "  404  209  " http ://
localhost / x1 . html " " Mozilla / 5 . 0  ( Macintosh ;  Intel
Mac  OS  X  10_11_3 )  AppleWebKi  t / 537 . 36  ( KHTML ,  like
Gecko )  Chrome / 48 . 0 . 2564 . 97  Safari / 537 . 36 "
```

Configure a Logtail client to collect Apache logs

1. On the Logstores page, click the Data Import Wizard icon.
2. Select a data type.

   Select APACHE Access Log.
3. Configure data source.

   a. Enter the Configuration Name and Log Path.

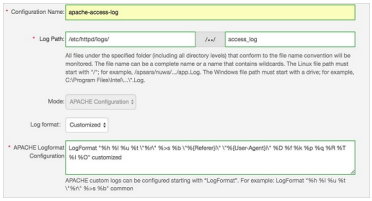
   b. Select a Log format.

   c. Enter APACHE Logformat Configuration if you select the customized log format.

   Enter the log format configuration fields of the standard APACHE configuration file. Generally, the configuration file starts with LogFormat.

    **Note:**
   If you select common or combined from the Log format drop-down list, the configuration fields of the corresponding log format are automatically added

here. You need to confirm whether the added configuration fields are consistent with the format defined in the local Apache configuration file.



d. Confirm APACHE Key Name.

Log Service automatically reads your Apache keys. Confirm the Apache key names on the current page.



e. (Optional) Configure Advanced options.

Configuration item	Desceiption
Upload Original Log	Select whether or not to upload the original log. If enabled , the new field is added by default to upload the original log.
Topic Generation Mode	· Null - Do not generate topic: The default option, which indicates to set the topic as a null string and you can query logs without entering the topic. · Machine Group Topic Attributes: Used to clearly differentiate log data generated in different frontend servers. · File Path Regular: With this option selected, you must enter the Custom RegEx to use the regular expression to extract contents from the path as the topic. Used to differentiate log data generated by users and instances. Used to differentiate log data generated by users and instances.
Custom RegEx	After selecting File Path Regular as Topic Generation Mode, you must enter your custom regular expression.

Configuration item	Desceiption
Log File Encoding	· utf8: Use UTF-8 encoding. · gbk: Use GBK encoding.
Maximum Monitor Directory Depth	Specify the maximum depth of the monitored directory when logs are collected from the log source, that is, at most how many levels of logs can be monitored. The range is 0–1000, and 0 indicates to only monitor the current directory level.
Timeout	A log file has timed out if it does not have any update within a specified time. You can configure the following settings for Timeout. · Never Time out: Specify to monitor all log files persistently and the log files never time out. · 30 minute timeout: A log file has timed out and is not monitored if it does not have any update within 30 minutes.
Filter Configuration	Only logs that completely conform to the filter conditions can be collected. For example: · collect logs that conform to a condition : Key:level Regex:WARNING|ERROR indicates to only collect logs whose level is WARNING or ERROR. · *filter logs that do not conform to a condition* : - `Key : level Regex :^(?!. *( INFO | DEBUG ))`, indicates to not collect logs whose level is INFO or DEBUG. - `Key : url Regex :. *^(?!.*( healthchec k )). *`, indicates to filter logs with healthcheck in the url. Such as logs in which key is url and value is `/ inner / healthchec k / jiankong . html` will not be collected. For similar examples, see*regex-exclude-word* and *regex-exclude-pattern*.

4. Click Next.
5. Select a machine group and then click Apply to Machine Group.

   If you have not created any machine group, click +Create Machine Group to create one.

   After you apply the Logtail configuration to the machine group, Log Service collects Apache logs according to the configuration. You can configure indexes and log shippers by following the steps of the Data Import Wizard.

5.6.2 Nginx logs

The Nginx log format and directory are generally in the configuration file `/ etc / nginx / nginx . conf .`

Nginx log format

The log configuration file defines the print format of Nginx logs, that is, the main format:

```
log_format   main  '$ remote_add  r  - $ remote_use  r  [$ time_local
] "$ request " '
                '$ request_ti  me  $ request_le  ngth  '
                '$ status  $ body_bytes  _sent  "$ http_refer  er " '
                '"$ http_user_  agent "';
```

The declaration uses the main log format and the written file name.

```
access_log  / var / logs / nginx / access . log   main
```

Field Description

Field name	Definition
remoteaddr	The IP address of the client.
remote_user	The username of the client.
request	The requested URL and HTTP protocol.
status	The request status.
bodybytessent	The number of bytes (not including the size of the response header) sent to the client. The total number of bytes for this variable is the same as that sent to the client by bytes_sent in modlogconfig of the Apache module.
connection	The connection serial number.

Field name	Definition
connection_requests	The number of requests received by using a connection.
msec	The log write time, which is which is measured in seconds and precise to milliseconds.
pipe	Whether or not requests are sent by using the HTTP pipeline. p indicates requests are sent by using the HTTP pipeline. Otherwise, the value is . .
httpreferer	Web page link from which the access is directed.
“http_user_agent”	Information about the browser on the client. http_user_agent must be enclosed in double quotation marks.
requestlength	The length of a request, including the request line, request header, and request body.
Request_time	The request processing time, which is measured in seconds and precise to milliseconds. The time starts when the first byte is sent to the client and ends when the logs are written after the last character is sent to the client.
[$time_local]	he local time in the general log format . This variable must be enclosed in brackets.

Log sample

```
192 . 168 . 1 . 2  - - [ 10 / Jul / 2015 : 15 : 51 : 09  + 0800 ] "
GET  / ubuntu . iso   HTTP / 1 . 0 "  0 . 000   129   404   168  "-"
" Wget / 1 . 11 . 4   Red   Hat   modified "
```

Configure Logtail to collect Nginx logs

1. Click the Data Import Wizard chart in the Logstore list page to enter the data import wizard.
2. Select a data source.

   Select the text file and click Next.

3. Select the data source.

   a. Enter the Configuration Name, and Log Path.

   b. Enter the nNginx log format.

      Complete the standard Nginx profile log configuration section, typically beginning with the `log_format`. Log Service automatically reads your Nginx key.

   c. Set Advanced Options according to your requirements. Click Next after completing the configurations.

      For more information about advanced options, see Advanced options.

   After configuring Logtail, apply the configuration to the machine group to start collecting Nginx logs standardly.

5.6.3 Python logs

The logging module of Python provides a general logging system, which can be used by third-party modules or applications. The logging module provides different log levels and logging methods such as files, HTTP GET/POST, SMTP, and Socket. You can customize a logging method as needed. The logging module is the same as Log4j except that they have different implementation details. The logging module provides the logger, handler, filter, and formatter features.

To collect Python logs, we recommend you to use logging handler directly:

- *Automatically upload Python logs by using log handler*
- *Log handler automatically parses logs in K-V format*
- *Log handler automatically parses logs in JSON format*

Python log format

The log format specifies the output format of log records in formatter. The constructi on method of formatter needs two parameters: message format string and message date string. Both of the parameters are optional.

Python log format:

```
import   logging
import   logging . handlers
LOG_FILE  = ' tst . log '
handler  =  logging . handlers . RotatingFi  leHandler ( LOG_FILE ,
maxBytes  =  1024 * 1024 ,  backupCoun  t  =  5 ) #  Instantiat  e
the   handler
```

```
fmt  = '%( asctime ) s  - %( filename ) s :%( lineno ) s  - %( name )
s  - %( message ) s '
formatter  =  logging . Formatter ( fmt )   #  Instantiat  e   the
formatter
handler . setFormatt  er ( formatter )      #  Add   the   formatter
  to   the   handler
logger  =  logging . getLogger (' tst ')    #  Obtain   the   logger
  named   tst
logger . addHandler ( handler )           #  Add   the   handler   to
  the   logger
logger . setLevel ( logging . DEBUG )
logger . info (' first   info   message ')
logger . debug (' first   debug   message ')
```

Field description

The formatter is configured in the `%( key ) s` format, that is, replacing the dictionary keywords. The following keywords are provided:

Format	Meaning
%(name)s	The logger name of the generated log.
%(levelno)s	The log level in numeric format, including DEBUG, INFO, WARNING, ERROR, and CRITICAL.
%(levelname)s	The log level in text format, including DEBUG, INFO, WARNING, ERROR, and CRITICAL.
%(pathname)s	The full path of the source file where the statement that outputs the log resides (if available).
%(filename)s	The file name.
%(module)s	The name of the module where the statement that outputs the log resides.
%(funcName)s	The name of the function that calls the log output.
%(lineno)d	The code line where the function statement that calls the log output resides (if available).
%(created)f	The time (in the UNIX standard time format) when the log is created, which indicates the number of seconds since 1970-1-1 00:00:00 UTC.

Format	Meaning
%(relativeCreated)d	The interval (in milliseconds) between the log created time and the time that the logging module is loaded.
%(asctime)s	The log creation time, which is in the format of “2003-07-08 16:49:45,896” by default (the number after the comma (,) is the number of milliseconds).
%(msecs)d	The log creation time in the milliseconds.
%(thread)d	The thread ID (if available).
%(threadName)s	The thread name (if available).
%(process)d	The process ID (if available).
%(message)s	The log message.

Log sample

Log sample

```
2015 - 03 - 04   23 : 21 : 59 , 682  -  log_test . py : 16  -  tst
-  first   info   message
2015 - 03 - 04   23 : 21 : 59 , 682  -  log_test . py : 17  -  tst
-  first   debug   message
```

Common Python logs and the corresponding regular expressions:

- Log format

```
2016 - 02 - 19   11 : 03 : 13 , 410  -  test . py : 19  -  tst  -
first   debug   message
```

Regular expression:

```
(\ d +-\ d +-\ d +\ s \ S +)\ s +-\ s +([^:]+):(\ d +)\ s +-\ s +(\
 w +)\ s +-\ s +(. *)
```

- Log format

```
%( asctime ) s  - %( filename ) s :%( lineno ) s  - %( levelno ) s
 %( levelname ) s  %( pathname ) s  %( module ) s  %( funcName ) s
 %( created ) f  %( thread ) d  %( threadName ) s  %( process ) d
 %( name ) s  - %( message ) s
```

Log sample

```
2016 - 02 - 19   11 : 06 : 52 , 514  -  test . py : 19  -  10
  DEBUG   test . py   test  < module >  1455851212 . 514271
```

```
1398659966  87072   MainThread   20193   tst  -  first   debug
message
```

Regular expression:

```
(\ d +-\ d +-\ d +\ s \ S +)\ s -\ s ([^:]+):(\ d +)\ s +-\ s +(\ d
 +)\ s +(\ w +)\ s +(\ S +)\ s +(\ w +)\ s +(\ S +)\ s +(\ S +)\ s
 +(\ d +)\ s +(\ w +)\ s +(\ d +)\ s +(\ w +)\ s +-\ s +(. *)
```

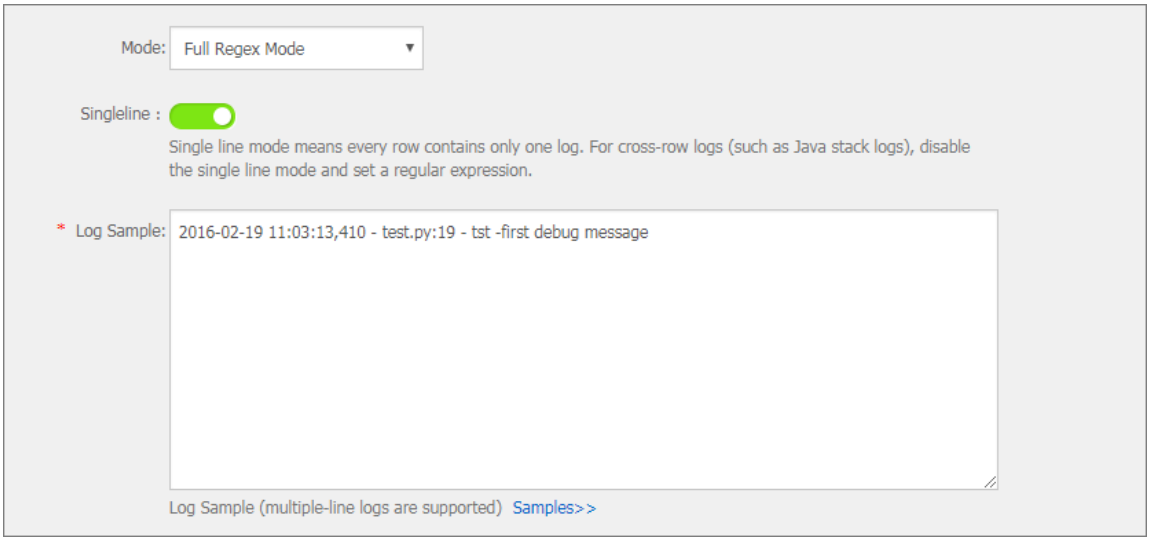
Configure Logtail to collect Python logs

For the detailed procedures of collecting Python logs by using Logtail, see *5-minute quick start*. Select the corresponding configuration based on your network deployment and actual situation.

1. Create a project and a Logstore. For detailed procedures, see *Preparation*.
2. On the Logstores page, click the Data Import Wizard icon.
3. Select a data source.

   Select the Text File.

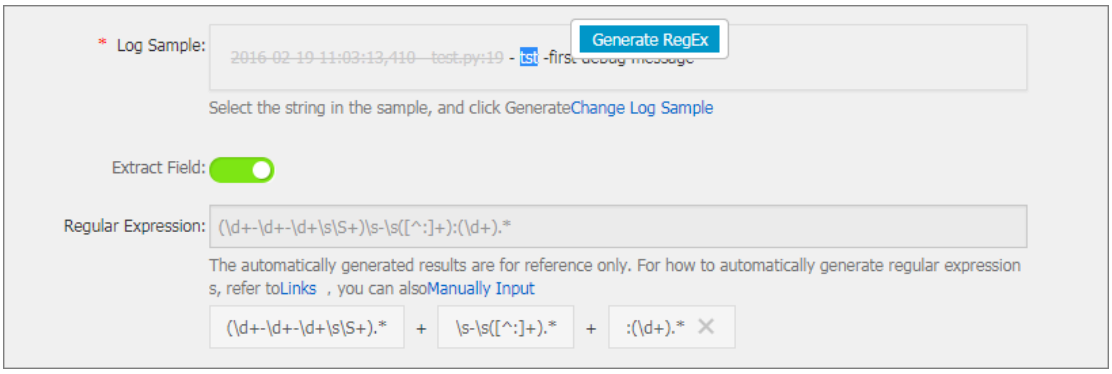
4. Configure the data source.
   a. Enter the Configuration Name and Log Path, and then select the Full Regex Mode from the mode drop-down list.
   b. Turn on the Singleline switch.
   c. Enter Log Sample.



   d. Turn on the Extract Field switch.
   e. Configure Regular Expression.
      A. Generates a regular expression by selecting strings of the log sample.

         If the automatically generated regular expression does not match your log sample, you can generate a regular expression by selecting strings of the log sample. Log Service supports selecting strings to automatically parse the log sample, that is, to automatically generates a regular expression for each selected field. In Log Sample, select log fields and click Generate RegEx. A regular expression of each selected field is displayed in the Regular

Expression column. You can generate a full regular expression for the log sample through multiple selections.



B. Modify the regular expression.

Considering the format of the actual log data may have minor changes, click Manually Input to adjust the automatically generated regular expression

according to the actual situations to conform to all log formats that may occur in the collection process.

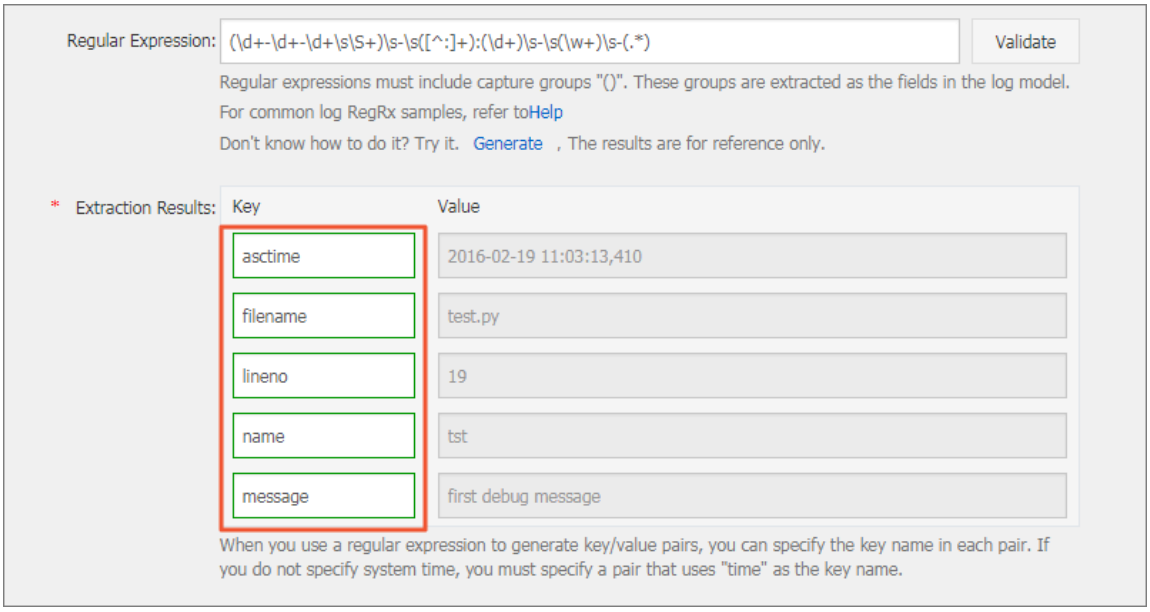
C. Validate the regular expression.

Click Validate after modifying the regular expression If the regular expression is correct, extracted results are displayed. Modify the regular expression if any errors exist.

f. Confirm Extraction Results.

View the parsing results of the log fields and enter corresponding keys for the log extraction results.

Assign a descriptive field name for each log field extraction result. For example , assign time for the time field. If you do not use the system time, you must specify a field where value is time, and name its key as time.



g. Turn on the System Time switch.

If you use the system time, the time of each log is the time when the Logtail client parses the log.

h. (Optional) Configure Advanced options.

i. Click Next.

After completing Logtail configuration, apply the configuration to the machine group to collect Python logs.

5.6.4 Log4j logs

Access Mode

Log Service supports collecting Log4j logs by using:

- LogHub Log4j Appender
- Logtail

Collect Log4j logs by using LogHub Log4j Appender

For more information, see *Log4j Appender*.

Collect Log4j logs by using Logtail

The log4j log consists of the first and second generations, and this document takes the default configuration of the first generation as an example, describes how to configure regular, if log4j is used 2. You need to modify the default configuration to print the date completely.

```
< Configurat  ion   status =" WARN ">
  < Appenders >
    < Console   name =" Console "  target =" SYSTEM_OUT ">
      < PatternLay  out   pattern ="% d { yyyy - MM - dd   HH : mm :
 ss : SSS   zzz } [% t ] %- 5level  % logger { 36 } - % msg % n "/>
    </ Console >
  </ Appenders >
  < Loggers >
    < Logger   name =" com . foo . Bar "  level =" trace ">
      < AppenderRe  f   ref =" Console "/>
    </ Logger >
    < Root   level =" error ">
      < AppenderRe  f   ref =" Console "/>
    </ Root >
  </ Loggers >
</ Configurat  ion >
```

For how to configure Logtail to collect Log4j logs, see *Python logs*. Select the corresponding configuration based on your network deployment and actual situation.

The automatically generated regular expression is only based on the log sample and does not cover all the situations of logs. Therefore, you must adjust the regular expression slightly after it is automatically generated.

Log4j e log sample of Log4j default log format printed to a file is as follows:

```
 2013 - 12 - 25   19 : 57 : 06 , 954  [ 10 . 207 . 37 . 161 ]  WARN
   impl . PermanentT  airDaoImpl  -  Fail   to   Read   Permanent
 Tair , key : e : 4702173193  19741_1 , result : com . example . tair
```

```
. Result @ 172e3ebc [ rc = code =- 1 ,  msg = connection   error   or
  timeout , value =, flag = 0 ]
```

Matching of the beginning of a line in multiline logs (use IP to indicate the beginning of a line):

```
\ d +-\ d +-\ d +\ s .
```

The regular expression used to extract log information:

```
(\ d +-\ d +-\ d +\ s \ d +:\ d +:\ d +,\ d +)\ s \[([^\]]*)\]\ s (\ S
 +)\ s +(\ S +)\ s -\ s (.
```

Time conversion format:

```
% Y -% m -% d  % H :% M :% S
```

Extraction results of the log sample:

Key	value
time	2013-12-25 19:57:06,954
ip	10.207.37.161
level	WARN
class	impl.PermanentTairDaoImpl
message	Fail to Read Permanent Tair,key:e:470217319319741_1,result :com.example.tair.Result@172e3ebc[rc=code=-1, msg= connection error or timeout,value=,flag=0]

5.6.5 Node.js logs

By default, Node.js logs are printed to the console, which makes the data collection and troubleshooting inconvenient. By using Log4js, logs can be printed to files and log format can be customized, which is convenient for data collection and coordination.

```
 var   log4js  =  require (' log4js ');
 log4js . configure ({
   appenders : [
    {
       type : ' file ', // file   output
       filename : ' logs / access . log ',
       maxLogSize :  1024 ,
       backups : 3 ,
       category : ' normal '
    }
  ]
});
 var   logger  =  log4js . getLogger (' normal ');
 logger . setLevel (' INFO ');
```

```
logger . info (" this   is   a   info   msg ");
logger . error (" this   is   a   err   msg ");
```

Log format

After the log data is stored in the text file format by using Log4js, the log is displayed in the following format in the file:

```
[ 2016 - 02 - 24   17 : 42 : 38 . 946 ] [ INFO ]  normal  -  this   is
   a   info   msg
[ 2016 - 02 - 24   17 : 42 : 38 . 951 ] [ ERROR ]  normal  -  this
 is   a   err   msg
```

Log4js has six output levels, including trace, debug, info, warn, error, and fatal in ascending order.

Collect Node.js logs by using Logtail

For how to configure Logtail to collect Log4j logs, see *Python logs*. Select the corresponding configuration based on your network deployment and actual situation.

The automatically generated regular expression is only based on the log sample and does not cover all the situations of logs. Therefore, you must adjust the regular expression slightly after it is automatically generated. Therefore, you must adjust the regular expression slightly after it is automatically generated. See the following Node.js log samples for reference and write a correct and comprehensive regular expression for your log.

See the following common Node.js logs and the corresponding regular expressions:

- Log sample 1:
  - Log sample:

    ```
    [ 2016 - 02 - 24   17 : 42 : 38 . 946 ] [ INFO ]  normal  -  this
       is   a   info   msg
    ```

  - Regular expression type

    ```
    \[([^]]+)]\ s \[([^\]]+)]\ s (\ w +)\ s -(. *)
    ```

  - Extracted fields:

    `time` , `level` , `loggerName` and `message` .

- Log sample 2:
  - Log sample:

```
[ 2016 - 01 - 31   12 : 02 : 25 . 844 ] [ INFO ]  access  -  42 .
 120 . 73 . 203  - - " GET  / user / projects / ali_sls_lo  g ?
 ignoreErro  r = true   HTTP / 1 . 1 "  304  - " http ://
 aliyun . com /" " Mozilla / 5 . 0  ( Macintosh ;  Intel   Mac
 OS   X   10_10_3 )  AppleWebKi  t / 537 . 36  ( KHTML ,  like
 Gecko )  Chrome / 48 . 0 . 2564 . 97   Safari / 537 . 36 "
```

  - Regular expression type

```
\[([^]]+)]\ s \[(\ w +)]\ s (\ w +)\ s -\ s (\ S +)\ s -\ s -\ s
 "([^"]+)"\ s (\ d +)[^"]+("[^"]+)"\ s "([^"]+). *
```

  - Extracted fields:

`time` , `level` 、, `loggerName` , `ip` , `request` , `status` , `referer` and `user_agent` .

5.6.6 WordPress logs

Default WordPress log format

Raw sample log:

```
172 . 64 . 0 . 2  - - [ 07 / Jan / 2016 : 21 : 06 : 39  + 0800 ]
" GET  / wp - admin / js / password - strength - meter . min . js
?  ver = 4 . 4   HTTP / 1 . 0 "  200   776  " http :// wordpress .
c4a1a0aecd  b194316955  5231dcc4ad  fb7 . cn - hangzhou . alicontain
er . com / wp - admin / install . php " " Mozilla / 5 . 0  (
Macintosh ;  Intel   Mac   OS   X   10_10_5 )  AppleWebKi  t / 537 .
36  ( KHTML ,  like   Gecko )  Chrome / 47 . 0 . 2526 . 106   Safari
/ 537 . 36 "
```

tching of the beginning of a line in multiline logs (use IP to indicate the beginning of a line):

```
\ d +\.\ d +\.\ d +\.\ d +\ s -\ s . *
```

The regular expression used to extract log information:

```
(\ S +) - - \[([^\]]*)] "(\ S +) ([^"]+)" (\ S +) (\ S +) "([^"]+)"
 "([^"]+)"
```

Time conversion format:

```
% d /% b /% Y :% H :% M :% S
```

Extraction results of the log sample:

Key	Value
ip	10.10.10.1
time	07/Jan/2016:21:06:39 +0800
method	GET
url	/wp-admin/js/password-strength-meter.min.js? ver=4.4 HTTP/1.0
status	200
length	776
ref	http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36

5.6.7 Delimiter logs

Log introduction

Delimiter logs use line breaks as the boundary. Each line is a log. The fields of each log are connected by fixed delimiters, including tabs, spaces, vertical lines (|), commas (,), semicolons (;), and other single characters. Fields containing delimiters are enclosed in double quotation marks (“).

Comma-separated values (CSV) logs and tab-separated values (TSV) logs are common delimiter logs.

Log format

A delimiter log is divided into several fields by delimiters, and supports two modes: single character and multiple characters.

- Single character mode

  Single character mode divides logs by matching single characters, such as tabs (`\ t `), spaces, vertical lines (`|`), commas (`,`), and semicolons (`;`).

   Note:

The double quotation mark (") cannot be a delimiter, but is used as the quote of the default single character delimiter.

Single character delimiters are often contained in log fields. To prevent log fields from being divided incorrectly, a double quotation mark (") is used as the quote to isolate the log field. If double quotation marks (“”) are found in the log content but not used as the quote, they must be escaped and processed as `""`. You can either use a double quotation mark (") in field border as the quote, or use double quotation marks ("") as field data. For other situations, use other modes, such as simple mode and full mode, to parse fields because the other conditions do not meet the format definition of delimiter logs.

- Double quotation mark (“) used as the quote

  When the double quotation mark (") is used as the quote, fields containing delimiters must be enclosed in a pair of quotes. The quote must be located adjacent to the delimiter. Modify the format if any spaces, tabs, and other characters exist between them.

  For example, comma (,) is the delimiter and double quotation mark (“) is the quote. The log format is `1997 , Ford , E350 ," ac , abs , moon ", 3000 . 00` . This log can be parsed into five fields, `1997` , `Ford` , `E350 ` , `ac , abs , moon` , and `3000 . 00` , which is enclosed in quotes, is considered as a complete field. `ac , abs , moon` , which is enclosed in quotes, is considered as a complete field.
- Double quotation marks (“”) used as a part of log field

  When double quotation marks (“”) are used as a part of the log field instead of the quote, they must be escaped and processed as `""`. Restore it when parsing fields, that is, restoring `""` to `"`.

  For example, a comma acts as a separator, double quotes, and comma as part of a log field, you must wrap the log field containing the comma in the quote, the double quotation marks in the log field are also escaped as the correct double quotation marks `""`. The log format after the processing is `1999 , Chevy ," Venture "" Extended Edition , Very Large ""","", 5000 . 00 ` . This log can be parsed into five fields, `1999` 、 `Chevy` 、 `Venture " Extended Edition , Very Large "`, a blank field, and `5000 . 00` .

- Multiple character mode

  In multiple character mode, a delimiter can contain two or three characters, such as `||`, `&&&`, `^ _ ^`. In this mode, logs are parsed completely by matching delimiters and you do not need to use the quote to enclose the logs.

  Note:

  Make sure that the full match of the delimiter does not appear in the log field. Otherwise, the field will be divided incorrectly.

  For example, if `&&` is the delimiter, the log `1997 && Ford && E350 && ac & abs & moon && 3000 . 00` can be parsed into 5 fields, `1997` , `Ford` , `E350` , `ac & abs & moon` , and `3000 . 00` .

Log sample

- Single character delimiter logs

```
05 / May / 2016 : 13 : 30 : 28 , 10 . 10 .*.*," POST  / PutData ?
Category = YunOsAccou  ntOpLog & AccessKeyI  d =****************&
Date = Fri % 2C % 2028 % 20Jun % 202013 % 2006 % 3A53 % 3A30 %
20GMT & Topic = raw & Signature =********************************
HTTP / 1 . 1 ", 200 , 18204 , aliyun - sdk - java
05 / May / 2016 : 13 : 31 : 23 , 10 . 10 .*.*," POST  / PutData ?
Category = YunOsAccou  ntOpLog & AccessKeyI  d =****************&
Date = Fri % 2C % 2028 % 20Jun % 202013 % 2006 % 3A53 % 3A30 %
20GMT & Topic = raw & Signature =********************************
HTTP / 1 . 1 ", 401 , 23472 , aliyun - sdk - java
```

- Multiple character delimiter logs

```
05 / May / 2016 : 13 : 30 : 28 && 10 . 200 .**.**&& POST  / PutData
? Category = YunOsAccou  ntOpLog & AccessKeyI  d =****************&
Date = Fri % 2C % 2028 % 20Jun % 202013 % 2006 % 3A53 % 3A30 %
20GMT & Topic = raw & Signature = pD12XYLmGx  KQ % 2Bmkd6x7hA
gQ7b1c % 3D  HTTP / 1 . 1 && 200 && 18204 && aliyun - sdk - java
05 / May / 2016 : 13 : 31 : 23 && 10 . 200 .**.**&& POST  / PutData
? Category = YunOsAccou  ntOpLog & AccessKeyI  d =****************&
Date = Fri % 2C % 2028 % 20Jun % 202013 % 2006 % 3A53 % 3A30 %
20GMT & Topic = raw & Signature =********************************
HTTP / 1 . 1 && 401 && 23472 && aliyun - sdk - java
```

Configure Logtail to collect delimiter logs

For the complete process of collecting logs by using Logtail, see *Python logs*. Select the corresponding configuration based on your network deployment and actual situation.

1. On the Logstore List page, click the Data Import Wizard.

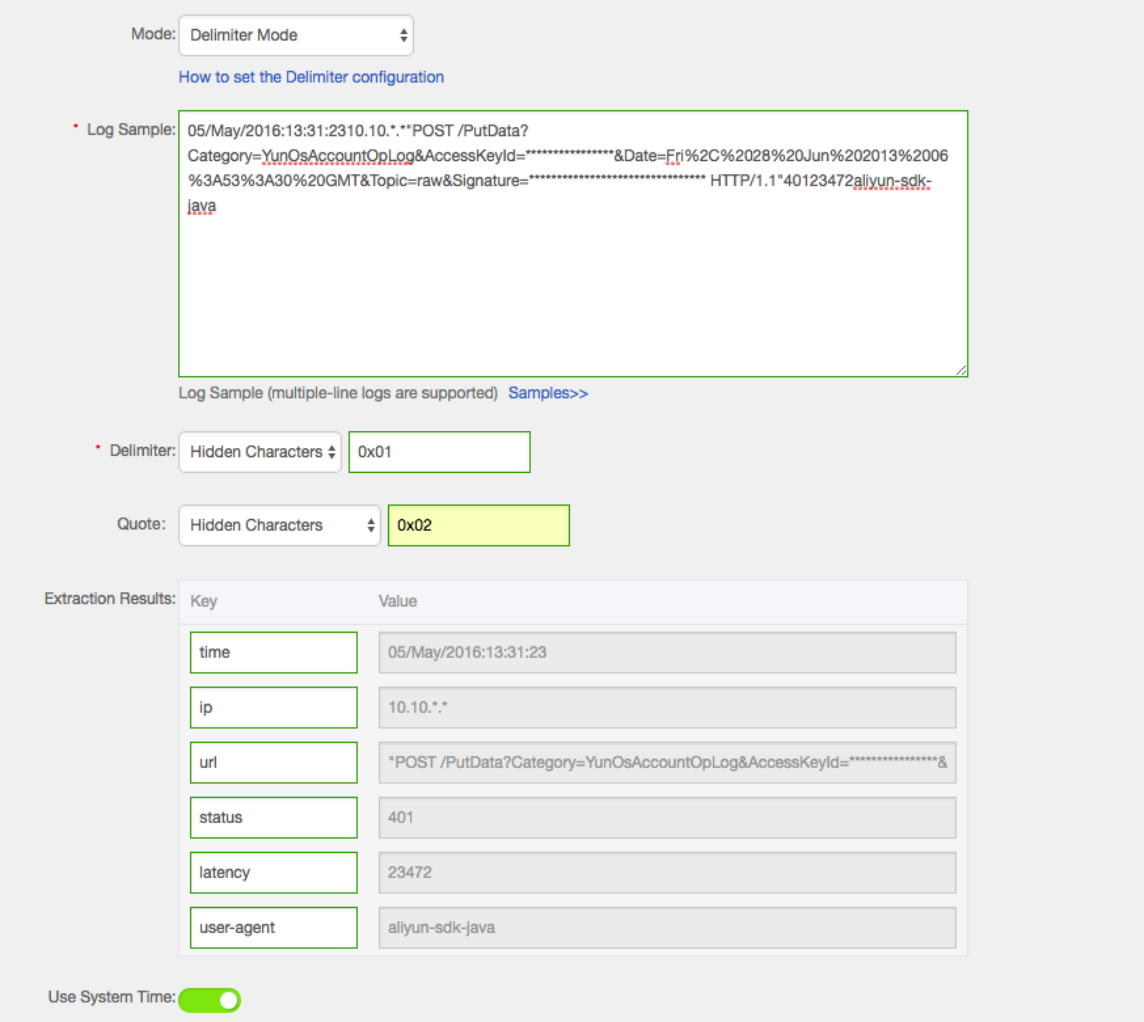
2. **Select the data source.**

   Select the text file and click Next.

3. Configure the data source.
   a. Enter the Configuration Name and Log Path. Then, select Delimiter Mode as the log collection mode.
   b. Enter the log sample and select the delimiter.

      Select the correct delimiter based on your log format. Otherwise, the log data will fail to be parsed.

      Figure 5-1: Select the data source.



   c. Specify the key in the log extraction results.

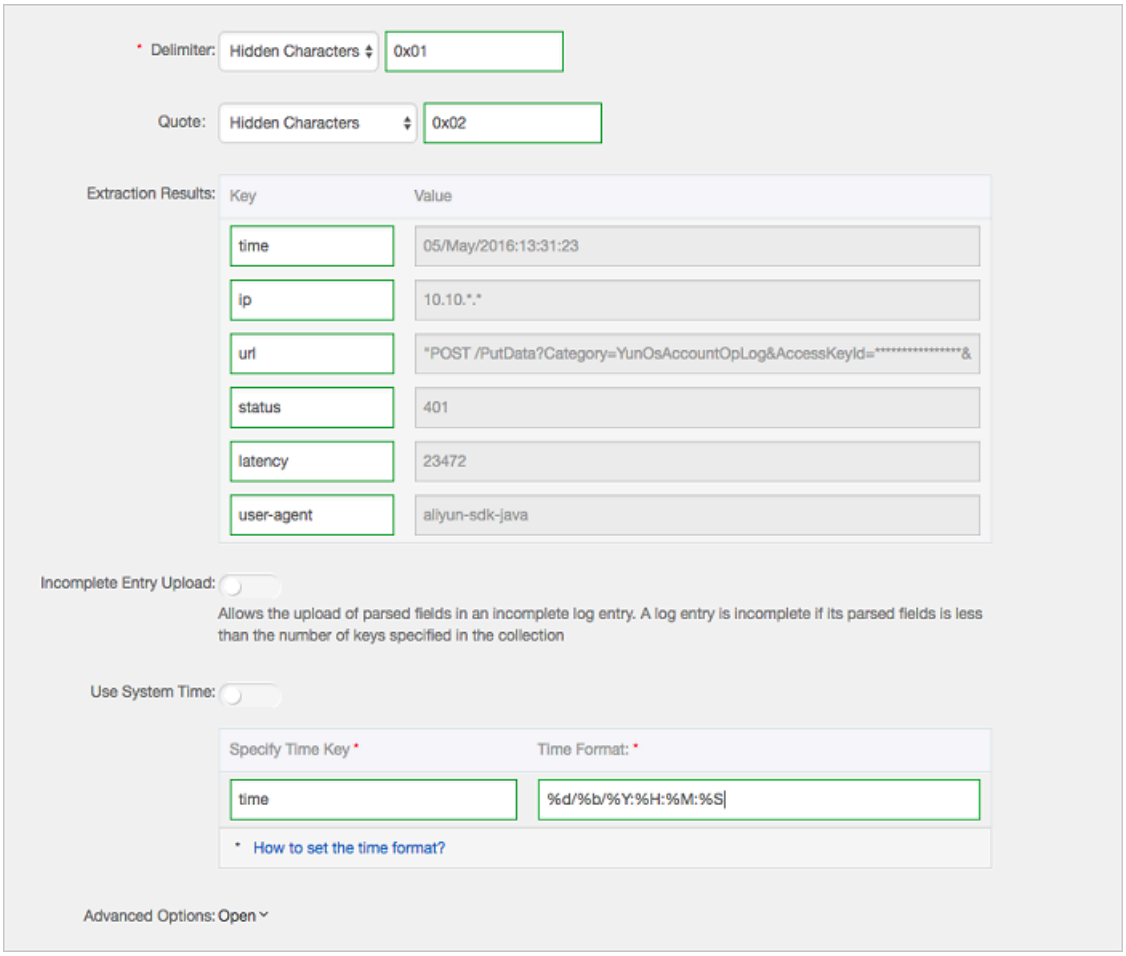
      After you enter the log sample and select the delimiter, Log Service extracts log fields according to your selected delimiter, and defines them as Value. You must specify the corresponding Key for the Value.

      For the preceding log sample, use a comma (,) as the delimiter, and six fields are in the log. Set the keys as time, ip, url, status, latency, and user-agent.

d. Specify the log time.

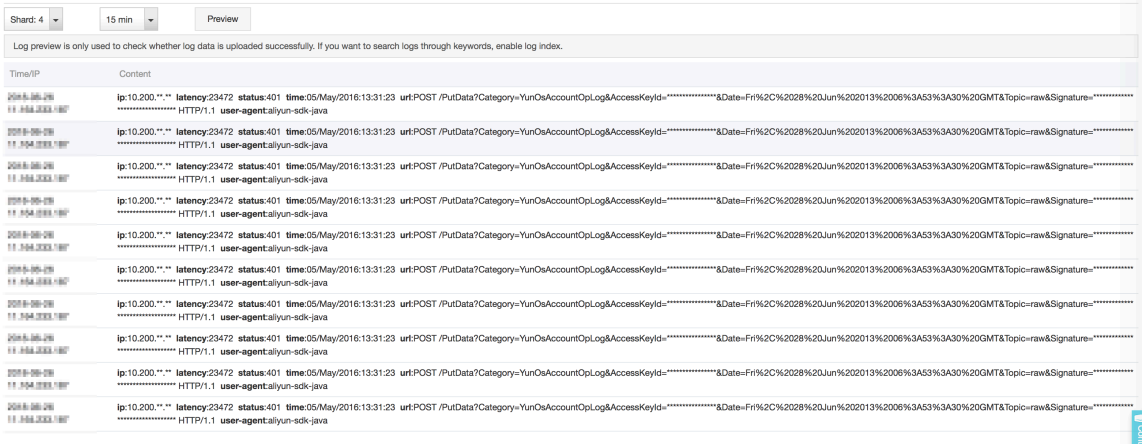
You can select to use the system time or a log field (such as the time field, 05/May/2016:13:30:29) as the log time. For how to configure the date format, see *Text logs - Configure time format*.

Figure 5-2: Specify log time



e. Preview logs in the console, and confirm whether logs are successfully collected.

Figure 5-3: Previewing logs



5.6.8 JSON logs

JSON logs are constructed in two structures:

- Object: A collection of key/value pairs.
- Array: An ordered list of values.

Logtail supports JSON logs of the object type. Logtail automatically extracts the keys and values from the first layer of an object as the names and values of fields respectively. The field value can be the object, array, or basic type, for example, a string or number. `\ n` is used to separate the lines of JSON logs. Each line is extracted as a single log.

Logtail does not support automatic parsing of non-object data (for example, JSON arrays). You can use regular expressions for field extraction or use the simple mode for log collection by line.

Log sample

```
{" url ": " POST  / PutData ?  Category = YunOsAccou  ntOpLog &
 AccessKeyI  d = U0Ujpek ********& Date = Fri % 2C % 2028 % 20Jun
 % 202013 % 2006 % 3A53 % 3A30 % 20GMT & Topic = raw & Signature =
 pD12XYLmGx  KQ % 2Bmkd6x7hA  gQ7b1c % 3D   HTTP / 1 . 1 ", " ip ": "
 10 . 200 . 98 . 220 ", " user - agent ": " aliyun - sdk - java ", "
 request ": {" status ": " 200 ", " latency ": " 18204 "}, " time ": "
 05 / May / 2016 : 13 : 30 : 28 "}
{" url ": " POST  / PutData ?  Category = YunOsAccou  ntOpLog &
 AccessKeyI  d = U0Ujpek ********& Date = Fri % 2C % 2028 % 20Jun
 % 202013 % 2006 % 3A53 % 3A30 % 20GMT & Topic = raw & Signature =
 pD12XYLmGx  KQ % 2Bmkd6x7hA  gQ7b1c % 3D   HTTP / 1 . 1 ", " ip ": "
 10 . 200 . 98 . 210 ", " user - agent ": " aliyun - sdk - java ", "
```

```
request ": {" status ": " 200 ", " latency ": " 10204 "}, " time ": "
05 / May / 2016 : 13 : 30 : 29 "}
```

Configure Logtail to collect JSON logs

For the complete process of collecting JSON logs by using Logtail, see *5-minute quick start*. This document shows the detailed configuration Log Collection Mode of Logtail.

1. On the Logstore List, click the Data Import Wizard.
2. Select the data type.

   Select the text file and click Next.
3. Configure the data source.

   a. Enter the configuration name, Log Path, and select log collection mode as JSON mode.

   b. Select whether to use the system time as the log time according to your requirements. You can enable or disable the Use System Time function.

   - Enable Use System Time function

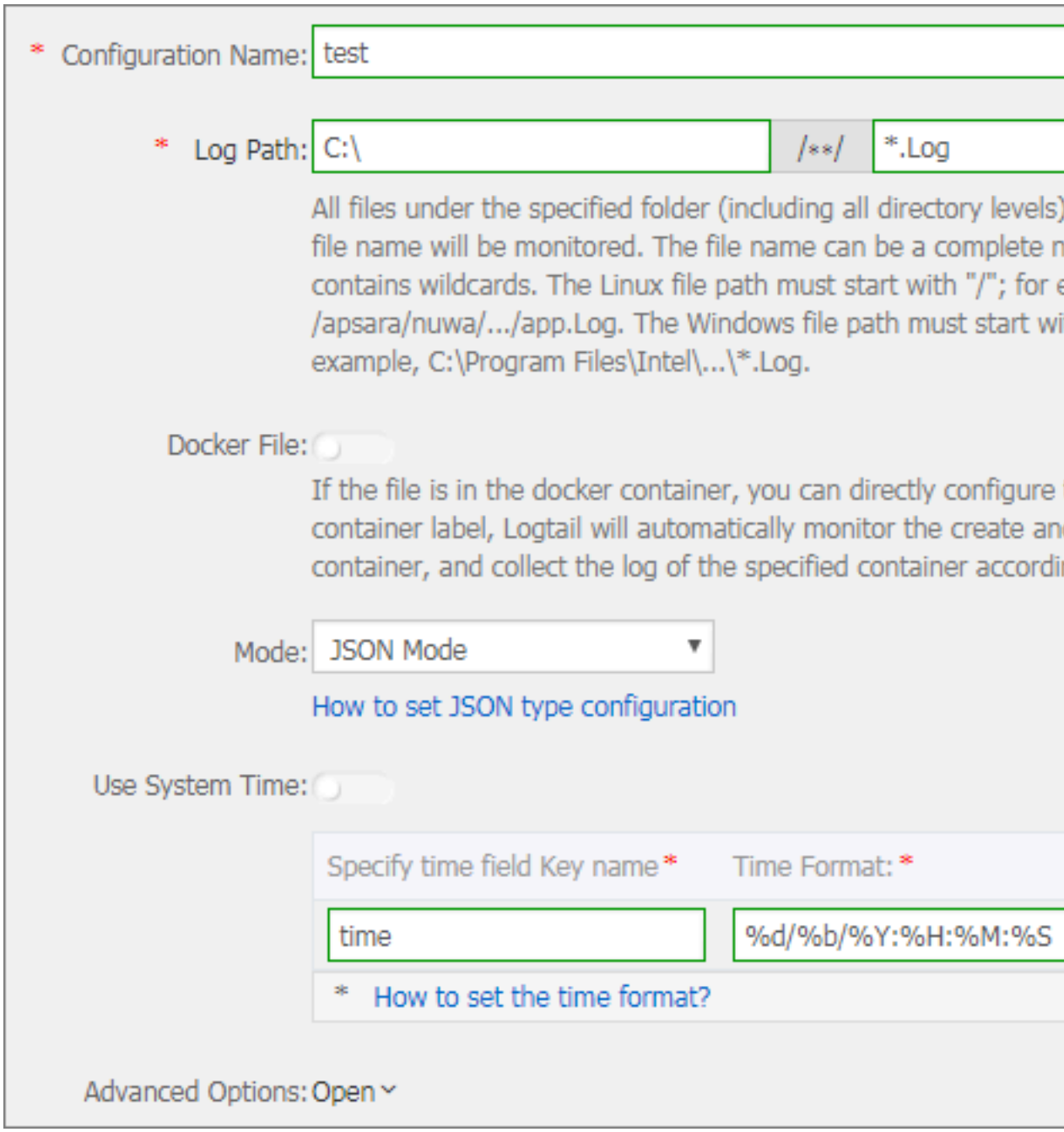
     Enabling this function means to use the time when Log Service collects the log as the log time, instead of extracting the time fields in the log.
   - Disable Use System Time function

     Disabling this function means to extract the time fields from the log as the log time.

     If you select to disable the Use System Time function, you must define the key of the extracted time field, and the time conversion format. For example, the

`time` field (05/May/2016:13:30:29) in JSON Object can be extracted as log time. For how to configure the date format, see *Text logs - Configure time format*.

Figure 5-4: JSON logs



5.6.9 ThinkPHP logs

ThinkPHP is a Web application development framework based on the PHP language.

Log format

Logs are printed in the following format in ThinkPHP:

```
<?  php
 Think \ Log :: record (' D   method   instantiat  ion   does   not
 find   the   model   class ' );
```

Log example

```
[  2016 - 05 - 11T21 : 03 : 05 + 08 : 00  ]  10 . 10 . 10 . 1  / index
 . php
 INFO : [  app_init  ] -- START --
 INFO :  Run   Behavior \ BuildLiteB  ehavior  [  RunTime : 0 .
 000014s  ]
 INFO : [  app_init  ] -- END -- [  RunTime : 0 . 000091s  ]
 Info : [ app_begin ] --  start  --
 INFO :  Run   Behavior \ ReadHtmlCa  cheBehavio  r  [  RunTime : 0 .
 000038s  ]
 INFO : [  app_begin  ] -- END -- [  RunTime : 0 . 000076s  ]
 INFO : [  view_parse  ] -- START --
 INFO :  Run   Behavior \ ParseTempl  ateBehavio  r  [  RunTime : 0 .
 000068s  ]
 INFO : [  view_parse  ] -- END -- [  RunTime : 0 . 000104s  ]
 INFO : [  view_filte  r  ] -- START --
 INFO :  Run   Behavior \ WriteHtmlC  acheBehavi  or  [  RunTime : 0 .
 000032s  ]
 INFO : [  view_filte  r  ] -- END -- [  RunTime : 0 . 000062s  ]
 INFO : [  app_end  ] -- START --
 INFO :  Run   Behavior \ ShowPageTr  aceBehavio  r  [  RunTime : 0 .
 000032s  ]
 INFO : [  app_end  ] -- END -- [  RunTime : 0 . 000070s  ]
 ERR :  D   method   instantiat  ion   does   not   find   the   model
   class
```

Configure Logtail to collect ThinkPHP logs

For the complete process of collecting ThinkPHP logs by using Logtail, see *Python logs*. Select the corresponding configuration based on your network deployment and actual situation.

The automatically generated regular expression is only based on the log sample and does not cover all the situations of logs. Therefore, you must adjust the regular expression slightly after it is automatically generated.

ThinkPHP logs are multiline logs whose mode is not fixed. The following fields can be extracted from the ThinkPHP logs: time, access IP, accessed URL, and printed message. The message field contains multiple lines of information and can only be packaged to one field because the mode is not fixed.

Logtail collects configuration parameters of ThinkPHP logs

Regular expression at the beginning of the line:

```
\[\ s \ d +-\ d +-\ w +:\ d +:\ d +\+\ d +:\ d +\ s .
```

Regular expression:

```
\[\ s (\ d +-\ d +-\ w +:\ d +:\ d +)[^:]+:\ d +\ s ]\ s +(\ S +)\ s
 (\ S +)\ s +(.
```

Time expression:

```
% Y -% m -% dT % H :% M :% S
```

5.6.10 Use Logstash to collect IIS logs

You need to modify the configuration file to parse the IIS log fields before you use logsturg to capture the IIS log.

Log sample

View IIS log configurations, select the W3C format (default field setting), and save the format to put it into effect.

```
 2016 - 02 - 25   01 : 27 : 04   112 . 74 . 74 . 124   GET  / goods /
 list / 0 / 1 . html  -  80  -  66 . 249 . 65 . 102   Mozilla / 5 . 0
 +( compatible ;+ Googlebot / 2 . 1 ;++ http :// www . google . com /
 bot . html )  404   0   2   703
```

Collection configuration

```
 input  {
   file  {
     type  => " iis_log_1 "
     path  => [" C :/ inetpub / logs / LogFiles / W3SVC1 /*. log "]
     start_posi  tion  => " beginning "
  }
}
 filter  {
   if  [ type ] == " iis_log_1 " {
  # ignore   log   comments
   if  [ message ] =~ "^#" {
     drop  {}
  }
   grok  {
    #  check   that   fields   match   your   IIS   log   settings
     match  => [" message ", "%{ TIMESTAMP_  ISO8601 : log_timest
 amp } %{ IPORHOST : site } %{ WORD : method } %{ URIPATH : page } %{
 NOTSPACE : querystrin  g } %{ NUMBER : port } %{ NOTSPACE : username
 } %{ IPORHOST : clienthost } %{ NOTSPACE : useragent } %{ NUMBER
 : response } %{ NUMBER : subrespons  e } %{ NUMBER : scstatus } %{
 NUMBER : time_taken }"]
  }
     date  {
```

```
      match  => [ " log_timest  amp ", " YYYY - MM - dd   HH : mm : ss
 " ]
        timezone  => " Etc / UTC "
  }
   useragent  {
     source => " useragent "
     prefix => " browser "
  }
   mutate  {
     remove_fie  ld  => [ " log_timest  amp "]
  }
  }
}
 output  {
   if  [ type ] == " iis_log_1 " {
   logservice  {
         codec  => " json "
         endpoint  => "***"
         project  => "***"
         logstore  => "***"
         topic  => ""
         source  => ""
         access_key  _id  => "***"
         access_key  _secret  => "***"
         max_send_r  etry  =>  10
    }
    }
}
```



Note:

- The configuration file must be encoded in UTF-8 format without BOM. You can use Notepad++ to modify the file encoding format.
- `path` indicates the file path, which must use delimiters in the UNIX format, for example, `C :/ test / multiline /*. log` . Otherwise, fuzzy match is not supported.
- The `type` field must be modified unitedly and kept consistent in the file. If a machine has multiple Logstash configuration files, the type field in each configuration file must be unique. Otherwise, data cannot be processed properly.

Related plug-ins: *file* and *grok*.

Restart Logstash to apply configurations

Create a configuration file in the `conf` directory and restart Logstash to apply the file. See *Set Logstash as a Windows service* for more information.

5.6.11 Use Logstash to collect CSV logs

You need to modify the configuration file to parse the CSV log fields before you use logsturg to capture the CSV log. The acquisition of the CSV log can use the system

time of the acquisition log as the upload log time, you can also use the time in the contents of the log as the upload log time. For different definitions of log time, there are two ways to configure logstroudsburg to collect CSV logs.

Use the system time as the uploaded log time

- Log sample

```
10 . 116 . 14 . 201 ,-, 2 / 25 / 2016 , 11 : 53 : 17 , W3SVC7
, 2132 , 200 , 0 , GET , project / shenzhen - test / logstore /
logstash / detail , C :\ test \ csv \ test_csv . log
```

- Collection configuration

```
 input  {
   file  {
     type  => " csv_log_1 "
     path  => [" C :/ test / csv /*. log "]
     start_posi  tion  => " beginning "
  }
}
 filter  {
   if  [ type ] == " csv_log_1 " {
   csv  {
     separator  => ","
     columns  => [" ip ", " a ", " date ", " time ", " b ", "
 latency ", " status ", " size ", " method ", " url ", " file "]
  }
  }
}
 output  {
   if  [ type ] == " csv_log_1 " {
   logservice  {
         codec  => " json "
         endpoint  => "***"
         project  => "***"
         logstore  => "***"
         topic  => ""
         source  => ""
         access_key  _id  => "***"
         access_key  _secret  => "***"
         max_send_r  etry  =>  10
    }
    }
}
```



Note:

- The configuration file must be encoded in UTF-8 format without BOM. You can use Notepad++ to modify the file encoding format.
- *path* indicates the file path, which must use delimiters in the UNIX format, for example, *C :/ test / multiline /*. log* . Otherwise, fuzzy match is not supported.

- `type` field must be modified unitedly and kept consistent in the file. If a machine has multiple Logstash configuration files, `type` field in each configuration file must be unique. Otherwise, data cannot be processed properly.

Related plug-ins: *file* and *csv*.

- Restart Logstash to apply configurations

  Create a configuration file in the `conf` directory and restart Logstash to apply the file. For more information, see Set *Set Logstash as a Windows service* as a Windows service.

Upload the log field content as the log time

- Log sample

```
10 . 116 . 14 . 201 ,-, Feb   25   2016   14 : 03 : 44 , W3SVC7
, 1332 , 200 , 0 , GET , project / shenzhen - test / logstore /
logstash / detail , C :\ test \ csv \ test_csv_w  ithtime . log
```

- Collection configuration

```
 input  {
   file  {
     type  => " csv_log_2 "
     path  => [" C :/ test / csv_withti  me /*. log "]
     start_posi  tion  => " beginning "
  }
}
 filter  {
   if  [ type ] == " csv_log_2 " {
   csv  {
     separator  => ","
     columns  => [" ip ", " a ", " datetime ", " b ", " latency ", "
 status ", " size ", " method ", " url ", " file "]
  }
   date  {
     match  => [ " datetime " , " MMM   dd   YYYY   HH : mm : ss " ]
  }
  }
}
 output  {
   if  [ type ] == " csv_log_2 " {
   logservice  {
         codec  => " json "
         endpoint  => "***"
         project  => "***"
         logstore  => "***"
         topic  => ""
         source  => ""
         access_key  _id  => "***"
         access_key  _secret  => "***"
         max_send_r  etry  =>  10
    }
    }
```

```
}
```

Note:

- The configuration file must be encoded in UTF-8 format without BOM. You can use Notepad++ to modify the file encoding format.
- `path` indicates the file path, which must use delimiters in the UNIX format, for example, `C :/ test / multiline /*. log` . Otherwise, fuzzy match is not supported.
- `type` field must be modified unitedly and kept consistent in the file. If a machine has multiple Logstash configuration files, `type` field in each configuration file must be unique. Otherwise, data cannot be processed properly.

Related plug-ins: *file* and *csv*.

- Restart Logstash to apply configurations

  Create a configuration file in the `conf` directory and restart Logstash to apply the file. For more information, see Set *Set Logstash as a Windows service* as a Windows service.

5.6.12 Use Logstash to collect other logs

You can modify the configuration file to parse log fields before you use logsturg to capture logs.

Upload using system time as log time

- Log sample

```
2016 - 02 - 25   15 : 37 : 01  [ main ]  INFO   com . aliyun . sls
. test_log4j  -  single   line   log
2016 - 02 - 25   15 : 37 : 11  [ main ]  ERROR   com . aliyun . sls
. test_log4j  -  catch   exception  !
 java . lang . Arithmetic  Exception : /  by   zero
    at   com . aliyun . sls . test_log4j . divide ( test_log4j .
java : 23 ) ~[ bin /:?]
    at   com . aliyun . sls . test_log4j . main ( test_log4j . java
: 13 ) [ bin /:?]
2016 - 02 - 25   15 : 38 : 02  [ main ]  INFO   com . aliyun . sls
. test_log4j  -  normal   log
```

- Collection configuration

```
input  {
  file  {
    type  => " common_log  _1 "
    path  => [" C :/ test / multiline /*. log "]
```

```
      start_posi  tion  => " beginning "
      codec  =>  multiline  {
        pattern  => "^\ d { 4 }-\ d { 2 }-\ d { 2 } \ d { 2 }:\ d { 2
 }:\ d { 2 }"
        negate  =>  true
        auto_flush  _interval  =>  3
        what  =>  previous
    }
  }
}
 output  {
   if  [ type ] == " common_log  _1 " {
   logservice  {
         codec  => " json "
         endpoint  => "***"
         project  => "***"
         logstore  => "***"
         topic  => ""
         source  => ""
         access_key  _id  => "***"
         access_key  _secret  => "***"
         max_send_r  etry  =>  10
    }
    }
}
```



Note:

- The configuration file must be encoded in UTF-8 format without BOM. You can use Notepad++ to modify the file encoding format.
- *path* indicates the file path, which must use delimiters in the UNIX format, for example, *C :/ test / multiline /*. log* . Otherwise, fuzzy match is not supported.
- *type* field must be modified unitedly and kept consistent in the file. If a machine has multiple Logstash configuration files, the *type* field in each configuration file must be unique. Otherwise, data cannot be processed properly.

Related plug-ins: *file* and multiline(for a single-line log file, remove the codec => multiline configuration).

- Restart Logstash to apply configurations

  Create a configuration file in the *conf* directory and restart Logstash to apply the file. For more information, see *Set Logstash as a Windows service*.

5.6.13 Unity3D logs

Context

Unity3D is an integrated game development tool compatible with multiple platforms . Developed by Unity Technologies, this tool allows a player to easily create various interactive contents such as 3D video game, architectural visualization, and real-time 3D animation. Unity3D is a fully integrated and professional game engine.

You can use the Web Tracking function of Log ServiceWeb Tracking to collect Unity3D logs conveniently. This document introduces how to use the Web Tracking function to collect Unity logs to Log Service by collecting the `Unity Debug . Log` .

Procedure

1. Activate the Web Tracking function

   For more information, see Web Tracking .

2. Register Unity3D LogHandler

   Create a C# file `LogOutputH andler . cs` in the Unity editor. Enter the following codes and modify three member variables in the codes, which are:

   - project, indicating the name of the log project.
   - logstore, indicating the name of the Logstore.
   - serviceAddr, indicating the address of the log project.

     For more information, see Service endpoint.

```
using UnityEngin e ;
using System . Collection s ;
public class LogOutputH andler : MonoBehavi our
{
    // Register the HandleLog function on scene start
to fire on debug . log events
    public void OnEnable ()
    {
        Applicatio n . logMessage Received += HandleLog ;
    }
    // Remove callback when object goes out of scope
    public void OnDisable ()
    {
        Applicatio n . logMessage Received -= HandleLog ;
    }
    string project = " your project name ";
    string logstore = " your logstore name ";
    string serviceAdd r = " http address of your log
service project ";
    // Capture debug . log output , send logs to Loggly
    public void HandleLog ( string logString , string
stackTrace , LogType type )
    {
        string parameters = "";
        parameters += " Level =" + WWW . EscapeURL ( type .
ToString ());
        parameters += "&";
```

```
        parameters  += " Message =" +  WWW . EscapeURL ( logString
 );
        parameters  += "&";
        parameters  += " Stack_Trac  e =" +  WWW . EscapeURL (
 stackTrace );
        parameters  += "&";
       // Add   any   User ,  Game ,  or   Device   MetaData   that
   would   be   useful   to   finding   issues   later
        parameters  += " Device_Mod  el =" +  WWW . EscapeURL (
 SystemInfo . deviceMode  l );
        string   url  = " http ://" +  project  + "." +
 serviceAdd  r  + "/ logstores /" +  logstore  + "/ track ?
 APIVersion = 0 . 6 . 0 &" +  parameters ;
        StartCorou  tine ( SendData ( url ));
    }
     public   IEnumerato  r   SendData ( string   url )
    {
        WWW   sendLog  =  new   WWW ( url );
        yield   return   sendLog ;
    }
}
```

The preceding codes can asynchronously send logs to Alibaba Cloud Log Service. You can add more fields that you want to collect in the example.

3. Generate Unity logs

   In the project, create the *LogglyTest . cs* file and add the following codes:

```
 using   UnityEngin  e ;
 using   System . Collection  s ;
 public   class   LogglyTest  :  MonoBehavi  our  {
     void   Start  () {
        Debug . Log  (" Hello   world ");
    }
}
```

4. Preview the log in the console.

   After completing the preceding steps, run the Unity program. Then, you can preview your sent logs in the Log Service console.

   The preceding example provides the methods for collecting logs such as *Debug . Log* , *Debug . LogError* , and *Debug . LogExcepti on* . The component object model of Unity, its program crash API, and other types of Log APIs can be used to conveniently collect the device information on the client.

6 Index and query

6.1 Overview

Log Service enables you to query and analyze massive amounts of logs in real time by using the LogSearch and Analytics functions. If the index function is disabled, raw data can be used in the order that is defined by Kafka based on Shards. If the index function is enabled, data statistics and query are also supported.

Functional advantages

- Real-time: Logs can be analyzed immediately after they are written.
- Fast:
  - Query: Billions of data can be processed and queried within one second (with five conditions).
  - Analysis: Hundreds of millions of data can be aggregated and analyzed within one second (with aggregation by five dimensions and the GroupBy condition).
- Flexible: Query and analysis conditions can be changed as required to obtain results in real time.
- Extensive: Besides functions such as reports, dashboards, and quick analysis provided in the console, Log Service seamlessly interconnects with products such as Grafana, DataV, and Jaeger, and supports protocols such as RESTful API and JDBC.

Indexes

The index function is designed to sort a specific column or multiple columns in logs. By using indexes, you can quickly access the collected logs. However, before using the LogSearch and Analytics functions, you must collect logs and *enable the index function and configure indexes* for the logs.

Log Service provides full text indexes and field indexes.

- Full text indexes: In this mode, the entire log is configured with indexes. The default index is used to query all keys in the log. The log can be queried even if only one key is matched.

- field indexes: In this mode, indexes are configured for specific keys. This allows you to query a specific key to narrow down the query range.

The data type of fields must be specified when you use field indexes. Log Service supports *text*, *json*. *long*, and *double*. For more information, see *Index data type overview*.

Query methods

- Query logs in the console:

  You can log on to the Log Service console and specify a query time range and enter a query statement on the query and analysis page. For more information, see *Query logs* and *Query syntax*.

- Query logs through API calls:

  You can use the *GetLogs* and *GetHistograms* APIs to query logs.



Note:

Before querying logs, you must collect logs and *enable the index function and configure indexes* for the logs.

Query and analysis statement format

To query and analyze logs in real time, you need to enter a query and analysis statement. The statement consists of a query statement and an analysis statement, and the two statements are separated by a vertical bar (|). The following shows an example:

```
$ Search  |$ Analytics
```

Statement type	Required?	Description
Query statement	No	The query condition, which can contain keywords, blur values, numbers, ranges, and combined conditions If the query statement is empty or "*", no filter condition is set for the current data. That is, all data will be returned. For more information, see *Query syntax*.

Statement type	Required?	Description
Analysis statement	No	The analysis statement, which is used to calculate and collect query results or full data. If the analysis statement is empty, only query results will be returned but no statistical analysis will be performed. For more information, see *Syntax description*.

Other information

If you query a large amount of log data (such as a long query time span, where the data volume is over 10 billion), one request cannot query all the data. In this case, Log Service returns the existing data and notifies you that the query result is incomplete.

At the same time, the server caches the results of the query within 15 minutes. When the query result is partially cached, the server continues to scan log data that has not been cached. To reduce the workload of merging multiple query results, Log Service merges the result of the cache hit with the result of the new query and returns it to you.

Therefore, Log Service enables you to get the final result by calling the interface repeatedly with the same parameters.

6.2 Syntax description

Log Service provides a function similar to the SQL aggregate computing. This function integrates with the *query* function and the SQL computing function to compute the query results.

Syntax example:

```
status > 200 | select avg ( latency ), max ( latency ) , count ( 1 ) as c GROUP BY method ORDER BY c DESC LIMIT 20
```

Basic syntax:

```
[ search query ] | [ sql query ]
```

The SEARCH condition and computing condition are separated by a vertical bar (`|`). This syntax Indicates that the required logs are filtered from the log by the search

query, and SQL queries are computed for these logs. The search query syntax is specific to Log Service. For details, see *Query syntax*.

Prerequisites

To use the analysis function, you must click Enable of the SQL related fields in Search and Analysis config. For more information, see *Overview*.

- If you do not enable analysis function, computing function of up to 10 thousand lines of data per shard is provided, and the delay is relatively high.
- With the Enable Analytics turned on, Log Service provides the quick analysis in seconds.
- Only works for new data when function is enabled.
- No additional charges are incurred after the Enable Analytics is turned on.

Supported SQL syntax

Log Service supports the following SQL syntaxes. For details, click the specific links.

- **SELECT aggregate computing functions:**
  - *General aggregate functions*
  - *Security detection functions*
  - *Mapping function*
  - *Estimating functions*
  - *Mathematical statistics functions*
  - *Mathematical calculation functions*
  - *String functions*
  - *Date and time functions*
  - *URL functions*
  - *Regular expression functions*
  - *JSON functions*
  - *Type conversion functions*
  - *IP functions*
  - *Arrays*
  - *Binary string functions*
  - *Bit operation*
  - *Interval-valued comparison and periodicity-valued comparison functions*
  - *Comparison functions and operators*
  - *Lambda functions*
  - *Logical functions*
  - *Geospatial functions*
  - *Geo functions*
- *GROUP BY syntax*
- *Window functions*
- *HAVING syntax*
- *ORDER BY syntax*
- *LIMIT syntax*
- *Case when and if branch syntax*
- *UNNEST function*
- *Column alias*

- *Nested subquery*

Syntax structure

The SQL syntax structure is as follows:

- The FROM clause and WHERE clause are not required in the SQL statement. By default, FROM indicates to query the data of the current Logstore, and the WHERE condition is search query.
- The supported clauses include SELECT, GROUP BY, ORDER BY [ASC,DESC], LIMIT, and HAVING.



Note:

By default, only the first 10 results are returned. To return more results, add limit n. For example, `* | select count ( 1 ) as c , ip group by ip order by c desc limit 100 .`

Built-in fields

Log Service has built-in fields for statistics. These built-in fields are automatically added when you configure any valid column.

Field name	Type	Meaning
`__time__`	bigint	The log time.
`__source__`	varchar	The source IP of the log. This field is source when you query. The underscores (__) are added before and after source only in SQL.
`__topic__`	varchar	The log topic.

Limits

1. The highest concurrency of each project is 15.
2. A single column varchar has the maximum length of 2048 and is truncated if the length exceeds 2048.
3. By default, 100 lines of data are returned, and page turning is not supported. If you want more data to be returned, use *LIMIT syntax*.

Examples

Count the hourly PV, UV, and maximum delay corresponding to a user request, with the highest delay of 10:

```
*| select   date_trunc (' hour ', from_unixt  ime ( __time__ ))  as
 time ,
      count ( 1 )  as   pv ,
      approx_dis  tinct ( userid )  as   uv ,
      max_by ( url , latency )  as   top_latenc  y_url ,
      max ( latency , 10 )  as   top_10_lat  ency
      group   by   1
      order   by   time
```

6.3 Enable and set indexes

Before using the LogSearch/Analytics function of Log Service, you need to enable and set indexes for the logs.

Context

You can query the collected logs only after you enable and set indexes for the logs. Set indexes based on the log fields and your query requirements.



Note:

- After the LogSearch/Analytics function is enabled, data is indexed on the backend server. Therefore, index traffic is incurred and index storage space is required.
- Index settings take effect only on the data recorded after the settings are enabled or modified.
- At least one of the following indexes must be enabled for a log: full text index and key/value index.
- To use SQL statements to *analyze* the query result of a field, enable the Analytics function of the field.
- If you want to set an index for a *Tags* field, such as an Internet IP address or a Unix timestamp, set the Key Name to a value in the `__tag__ : key` format, for example, `_tag__ : __receive_  time__` . A Tags field does not support indexes of the numeric type. Instead, set the Type of all Tags fields to text. For example, to query a field with the key name `__tag__ : __receive_  time__` , you can use a fuzzy value, such as `__tag__ : __receive_  time__ :  1537928 *`, or the full value of the field, such as `__tag__ : __receive_  time__ :  1537928404` as the keyword.

When a log is collected, information about the log, such as the source and time, is automatically added to the log as key/value pairs. These fields are reserved in Log Service. When you enable and set indexes for logs, the indexes and the Analytics function are automatically enabled for these fields.

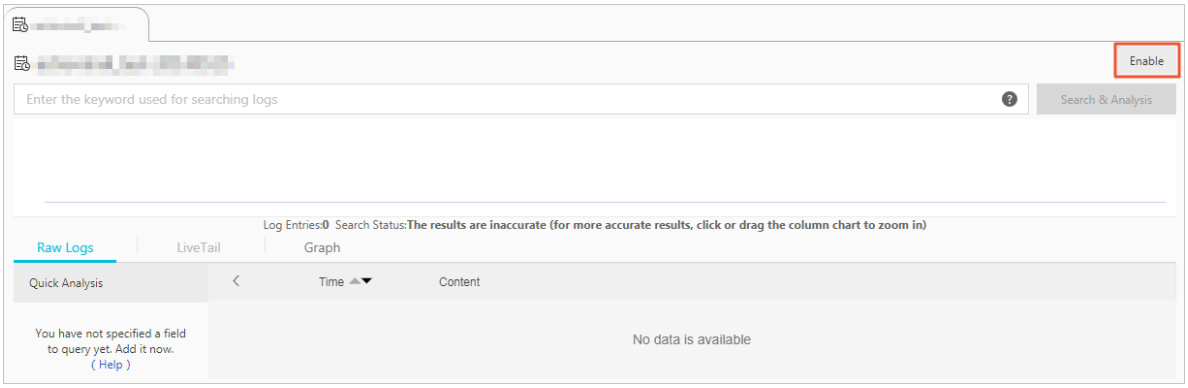
Note:
The delimiters of the `__topic__` and `__source__` fields are null. It means that the keywords used to query the two fields must match the field values.

Table 6-1: Reserved fields in Log Service

Name	Description
`__topic__`	Indicates the log topic. If you set a *topic* for a log, Log Service automatically adds a topic field to the log. The key of the field is `__topic__`, and the value of the field is the log topic.
`__source__`	Indicates the source equipment that generates the log.
`__time__`	Indicates the time that is specified when the log is recorded by the SDK.

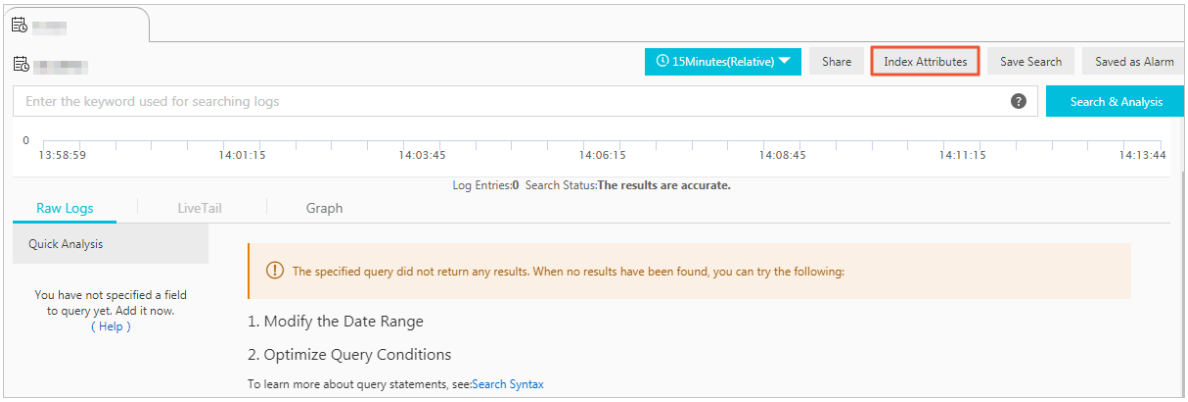
Procedure

1. Log on to the *Log Service console*, and then click the target project name.
2. In the LogSearch column, click Search.
3. Click Enable in the upper right corner.



Note:

If you have created an index, click Index Attributes > Modify.



4. Set indexes for logs.

   Log Service supports two indexes: full text index and key/value index. At least one of the two indexes must be set for a log.

   Note:
   If both a full text index and a key/value index are set for a log, the key/value index prevails.

Index type	Description
Full text index	Indicates that all fields in the log are queried as text with a key/value index. The key and value of the index are text and can both be queried. You do not need to specify the key name in queries.

Index type	Description
Key/Value index	After setting a key/value index for a field, you must specify the key name to query the field. If a full text index is set for a log and a key/value index is set for a field in the log, the full text index does not take effect on the field. You can set multiple data types for a field, including: · *Text* · *JSON* · *Numeric (Long and Double)*

a) Set a full text index for a log.

You can set an index for the full content of a log. The values of all keys in the log are queried by default when you query the log.

Parameter	Description	Example
Full Text Index	If this option is enabled, an index is enabled for the full content of the log . The values of all keys in the log are queried by default. The log can be queried if any one of the keys matches the keyword.	-
Case Sensitive	Specifies whether the queries are case-sensitive. · If this option is disabled, the queries are not case-sensitive, that is, an internal error log can be queried by both of the keywords "INTERNALERROR" and "internalerror". · If this option is enabled, the queries are case-sensitive, that is, a log that includes "internalError" can be queried only by the keyword "internalError".	-

Parameter	Description	Example
Chinese character	Sets whether to distinguish between English and Chinese. · After opening, if the log contains Chinese, the Chinese word segmentation is carried out according to the Chinese grammar , word Segmentation is carried out in English according to the word segmentation characters. · When closed, word all the content according to the word segmentation .	-
Delimiter	Specifies single-byte characters used to separate a log into multiple keywords. For example, if the content of a log is `a , b ; c ; D - F` , you can specify the comma (,), semi-colon (;), and hyphen (-) as delimiters to separate the log into five keywords: ” a” , “b” , “c” , “D” , and “F” .	`, '";=()[]{}? @&<>/:\ n \ t`

b) Set key/value indexes for a log.

You can set indexes for specified keys. After setting key/value indexes for a log, you can query specified keys to narrow down the query scope.



Note:

· Log Service automatically creates indexes for the *reserved fields* and enables the Analytics function of the fields. The reserved fields include `__topic__` , `__source__` , and `__time__` .

· The settings in the Customize tab page are described as an example in this topic. The Nginx Template and MNS Template are used only to collect Nginx logs and MNS logs and do not support customized index settings.

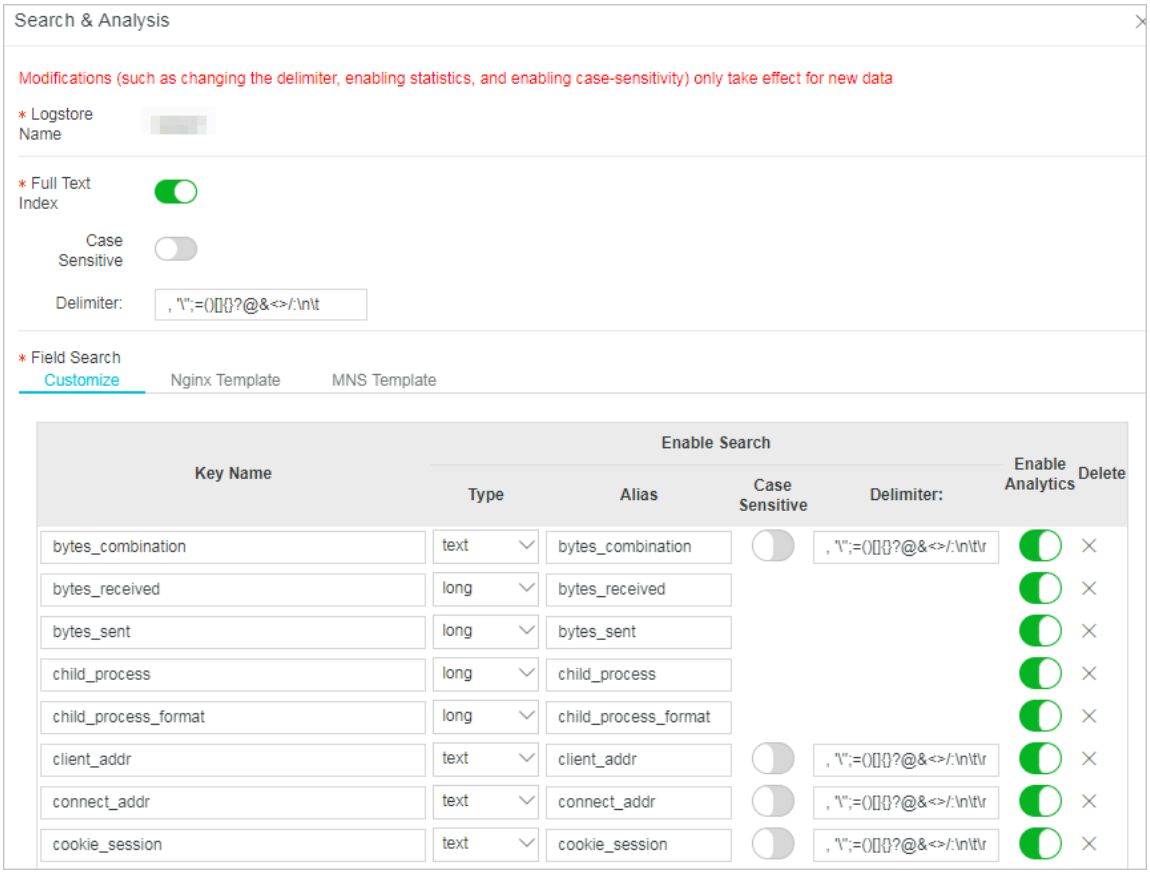
· If you want to set an index for a *Tags* field, such as an Internet IP address or a Unix timestamp, set the Key Name to a value in the `__tag__ : key` format, for example, `_tag__ : __receive_ time__` . A Tags field does not support indexes of the numeric type. Set the Type of all Tags fields to text. For example, to query a field with the key name `__tag__ : __receive_`

`time__`, you can use a fuzzy value, such as `__tag__ : __receive_ time__ : 1537928 *`, or the full value of the field, such as `__tag__ : __receive_ time__ : 1537928404` as the keyword.

Parameter	Description	Example
Key Name	Specifies the name of a field in the log.	`_address_`
Type	Specifies the data type of a field in the log, including: · text: Indicates that the content of the field is text. · long: Indicates that the content of the field is an integer. This field must be queried by a value range. · double: Indicates that the content of the field is a floating-point number. This field must be queried by a value range. · json: Indicates that the content of the field is in JSON format. Note: Numeric types (Long and Double) do not support Case Sensitive or Delimiter.	-
Alias	Indicates the alias of a column. An alias is used only for SQL statistics. A field is still identified by its original name in the underlying storage. Therefore, you must use the original name of a field to query the field. For more information, see *Column alias*.	`address`

Parameter	Description	Example
Case Sensitive	Specifies whether the queries are case-sensitive. This parameter has two values: · false: The queries are not case-sensitive, that is, the sample log can be queried by both of the keywords "INTERNALERROR" and "internalerror". · true: The queries are case-sensitive, that is, the sample log can be queried only by the keyword "internalError".	-
Delimiter	Specifies single-byte characters used to separate a log into multiple keywords. For example, if the content of a log is `a , b ; c ; D - F` , you can specify the comma (,), semi-colon (;), and hyphen (-) as delimiters to separate the log into five keywords: ” a” , “b” , “c” , “D” , and “F” .	`, '";=()[]{}? @&<>/:\ n \ t`

Parameter	Description	Example
Enable Analytics	Specifies whether the Analytics function is enabled. This function is enabled by default. After enabling the Analytics function, you can use query and analysis statements to analyze the query results.	-



5. Click OK.

Note:

- The index settings take effect within one minute.
- Index settings take effect only on data recorded after the settings are enabled or modified.

6.4 Query logs

After enabling the index function and setting indexes, you can query and analyze the collected logs in the console.

Prerequisites

- Logs have been collected.
- You have *enabled the index function and set indexes*.

Procedure

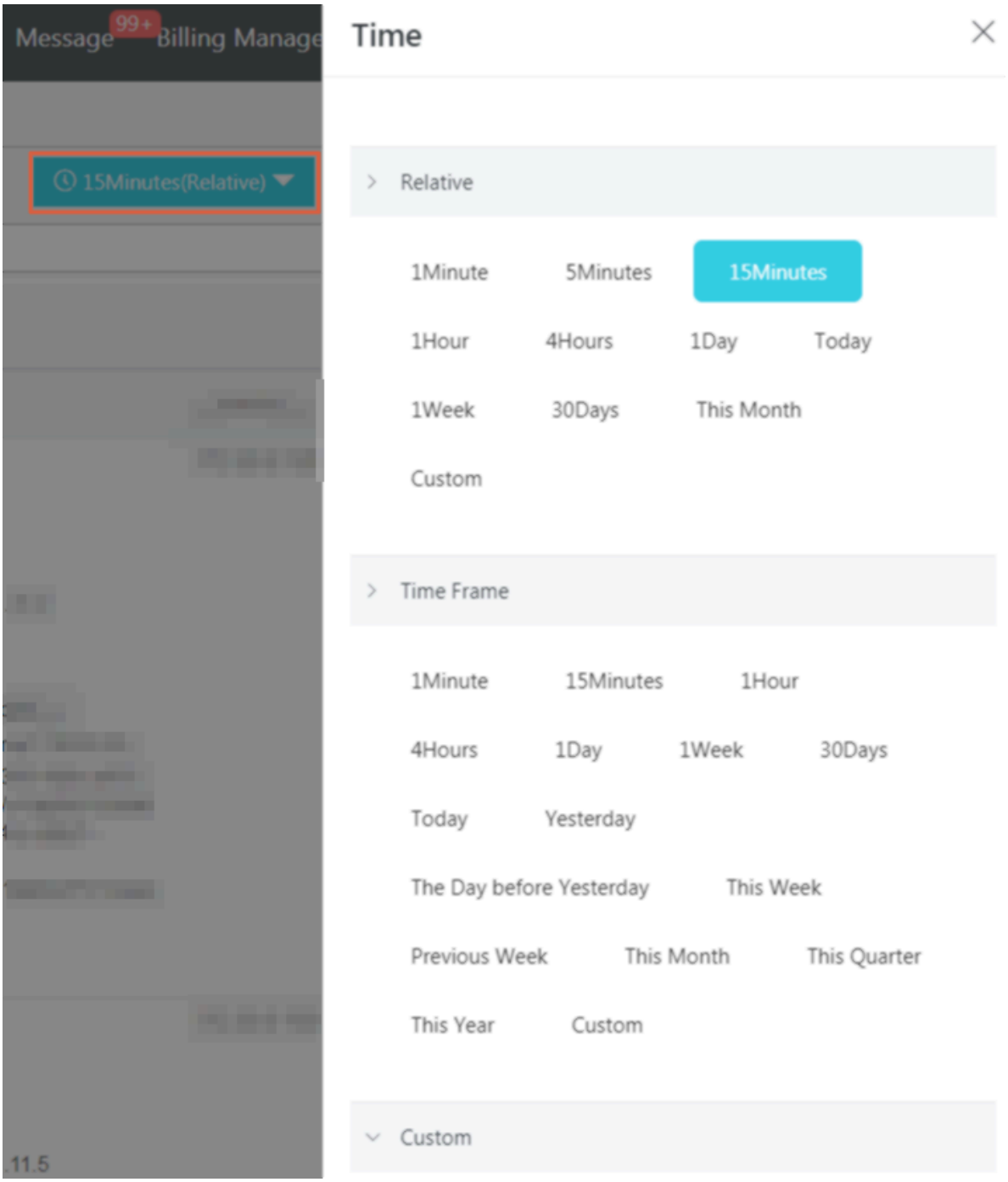
1. Log on to the *Log Service console*, and then click the target project name.
2. In the LogSearch column, click Search.
3. In the search box, enter a query analysis statement.

   A query analysis statement consists of a query statement and an analysis statement in the format of `query statement | analysis statement`. For more information, see *Overview*.
4. In the upper-right corner, click 15 Minutes (Relative)to set the time range for queries.

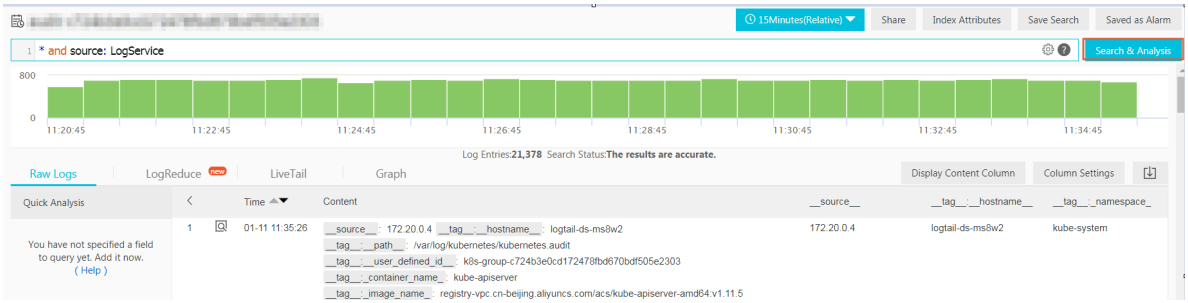
   You can choose between a relative time period and a time frame or customize a time range.

    Note:

The query results may contain logs obtained one minute earlier or later than the specified time range.



5. Click Search & Analysis to view the query results.



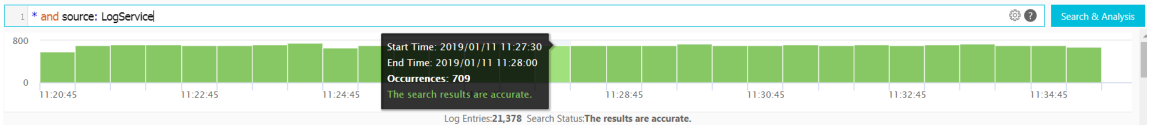
You can view the query results through a log distribution histogram, raw logs, or various graphs.

> **Note:**
> By default, 100 query results are returned. If you need to view more results, see *LIMIT syntax*.

- Log distribution histogram:

  The log distribution histogram shows the log distribution in the time dimension.

  - Rest the pointer on a green data block to view the time range indicated by the data block and the number of log query results within the time range.
  - Click a data block to view finer-grained log distribution. Additionally, you can view the log query results on the Raw logs tab page.



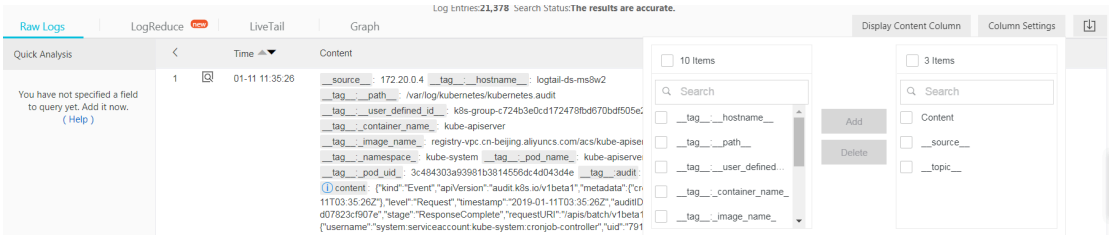
- Raw logs:

  On the Raw logs tab page, you can view the logs that match your search conditions.

  - Use the quick analysis function to receive a quick analysis of the distribution of a field over a period of time. For more information, see *Quick analysis*.
  - Click the download icon in the upper-right corner to specify a download range, and then click OK.
  - In the upper-right corner, click Column Settings. In the displayed dialog box, select the target fields from the left area and click Add to add the fields to the right area. Then, columns indicated by the added fields appear on the tab

page. The field names are also column names, and the columns list the field values.

> **Note:**
> To view the log content on the tab page, you must select Content.



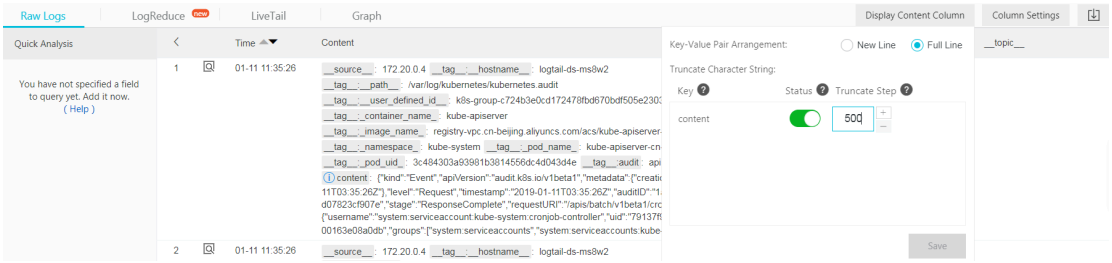
- Set the style of the content column. If the field contains more than 3,000 characters, then some content will be hidden and a message indicating this will be displayed before the field Key. Specifically, click Display Content Column. In the displayed dialog box, set Key-Value Pair Arrangement and Truncate Character String.

> **Note:**
> If the content limit is set to 10,000 characters, any character past this number will be downgraded. Further, none of these characters will be displayed, and no delimiter will be specified for these characters.

Parameter		Description
Key-Value Pair Arrangement		You can set this parameter to New Line or Full Line as needed.
Truncate Character String	Key	When a field value contains more than 3,000 characters, the value is truncated by default. However, this parameter remains empty if this value is not reached. The value of this parameter is the key of the truncated value.

Parameter		Description
	Status	You can determine whether to enable value truncation. It is enabled by default. ■ Enable: When a value length exceeds the preset value of Truncate Step, the value is automatically truncated. You can click the button at the end of a value to perform incremental expansion. The number of incremental characters is the value of Truncate Step. ■ Disable: A value will not be truncated even if its length exceeds the preset value of Truncate Step.

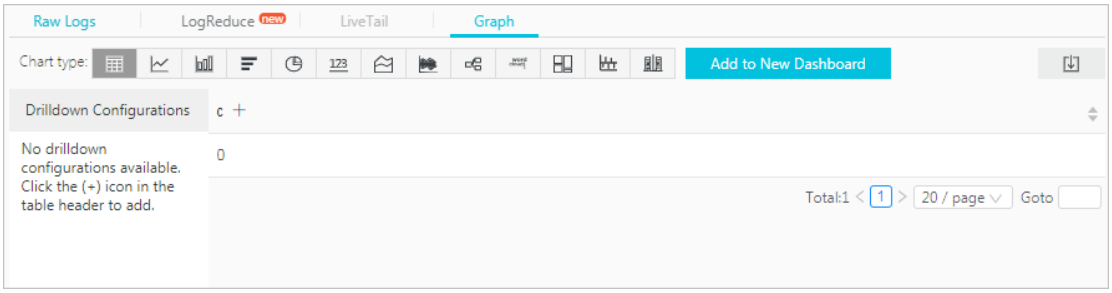
Parameter		Description
	Truncate Step	This parameter indicates the maximum number of a value as well as the number of incremental characters per time. The parameter value ranges from 500 to 10, 000 characters. The default value is 3,000.



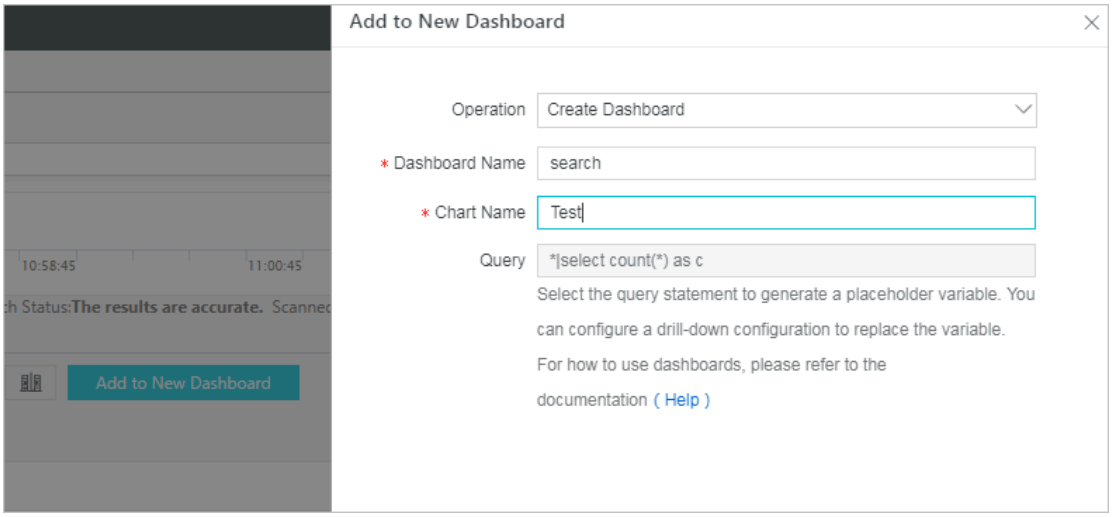
- Graph:

  If you have enabled the statistics function and used a query analysis statement for query, you can view the analysis results on the Graph tab page.

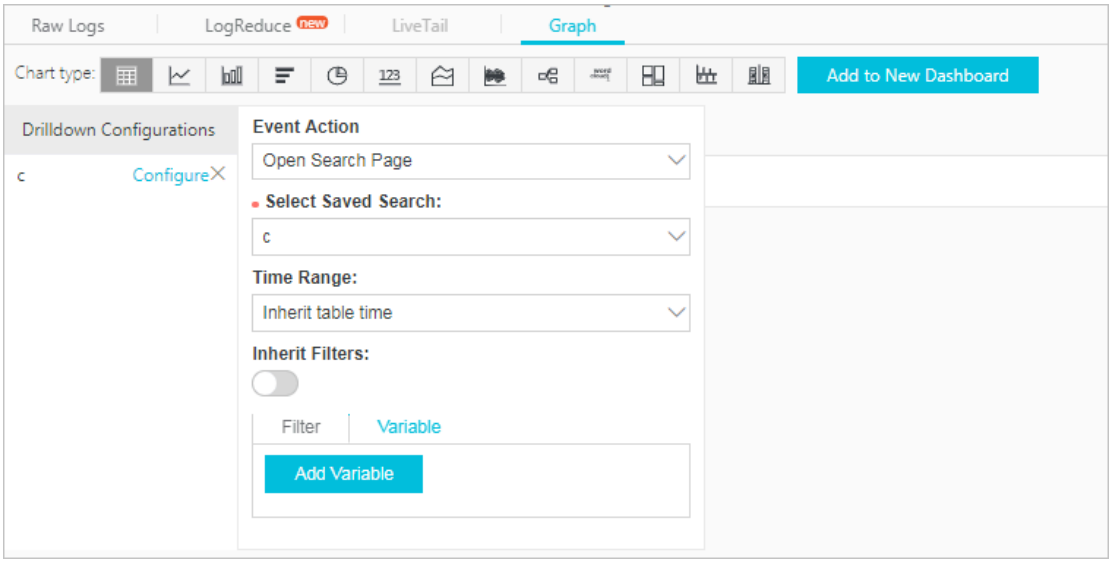
  - Select an appropriate *graph* type to show analysis results based on your requirements. Several chart types are provided in Log Serve including tables, line charts, and bar charts.



  - Add the graph to the *dashboard* for real-time data analysis results to be displayed. Click Add to dashboard to save common query statements as a graph saved on the dashboard.



- Set drill-down configurations to gain deeper insight into the analysis results. Then, click the values in the graph to view the analysis results from more dimensions. For more information, see *Drill-down analysis*.



Additionally, you can click Save Search and Save as Alarm in the upper-right corner. Then, you can use the *saved search* and *alarm* functions.

6.5 Download logs

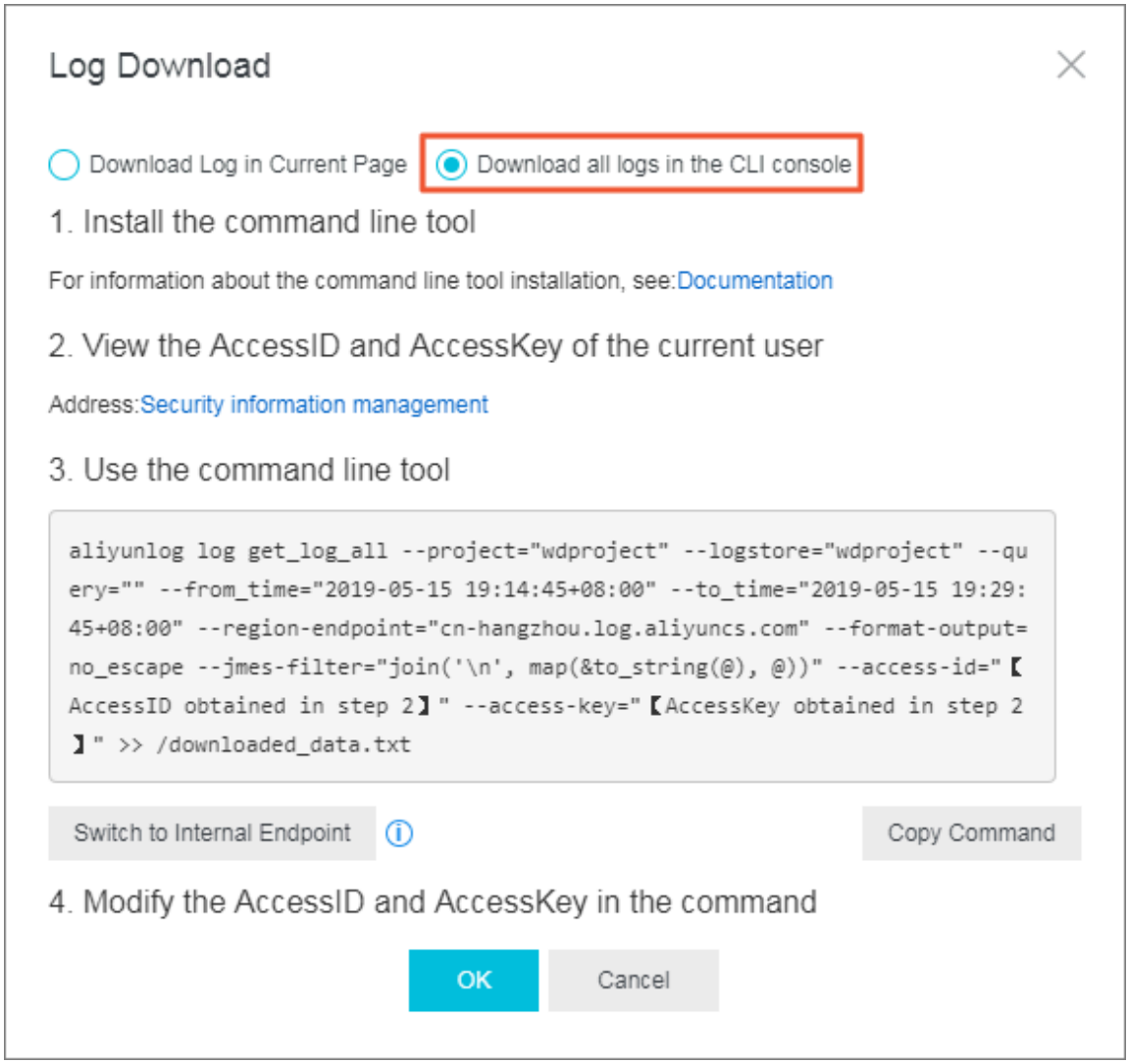
This topic describes how to download a specific page of logs in CSV format, or all logs in TXT format, to a local host.

Procedure

1. Log on to the *Log Service console*, and then click the target project name.
2. In the LogSearch column, click Search.

3. On the Raw Logs tab page, click the icon  .

4. Select the method to download logs.
   - Click Download Log in Current Page to download logs of the current page in the CSV format to your local host.
   - Click Download all logs in the CLI console to download all logs by using the CLI tool.



   a. Install the command line tool. For more information, see *Log Service CLI User Guide*.
   b. Select *Security information management* to copy your AccessID and AccessKey.
   c. Select Copy Command, and use the AccessID and AccessKey copied in the preceding step to replace `【AccessID obtained in step 2】` and `【AccessKey obtained in step 2】` in the command.
   d. Run the command in the CLI tool to download logs.

Note:
The logs are downloaded to the `download_data.txt` file. This file is located in the directory where the command was run.

6.6 Data type of index

6.6.1 Overview

Log Service allows you to set indexes for the full text or some fields of the collected logs. If you set an index for the full text of a log, the value used to query this log is the content of the entire log. If you set indexes for some fields of a log, you can set the data type of each key used in queries.

Data type

The following table describes the supported index types.

Query type	Index type	Description	Example
Basic	*text*	Indicates the text type that supports keywords and fuzzy matching in queries.	`uri :" login *" method :" post "`
	long	Indicates the numeric type that supports interval queries.	`status > 200 , status in [ 200 , 500 ]`
	double	Indicates the numeric type that supports floating-point numbers.	`price > 28 . 95 , t in [ 20 . 0 , 37 ]`
Combination	*json*	Indicates that the index is a JSON field that supports nested queries. The field type is Text by default. You can set an index of the Text , Long, or Double type for element 'b' under element 'a' by using a path format such as 'a.b'. The field type is determined by the index type you set.	`level0 . key > 29 . 95 level0 . key2 :" action "`

Query type	Index type	Description	Example
	Full text	Indicates that the full content of the log is queried as text.	`error and " login fail "`

Example

The following log includes time and other four keys.

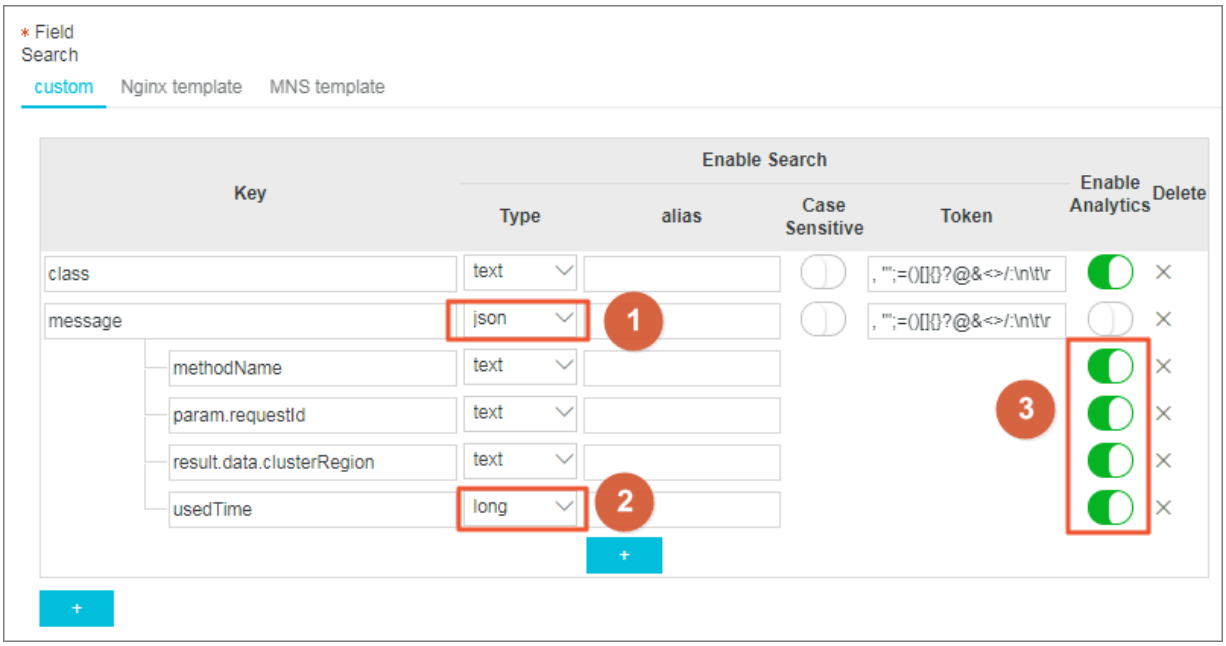
No.	Key	Type
0	time	-
1	class	text
2	status	long
3	latency	double
4	message	json

```
0 .  time : 2018 - 01 - 01   12 : 00 : 00
  1 .  class : central - log
  2 .  status : 200
  3 .  latency : 68 . 75
  4 .  message :
 {
     " methodName ": " getProject  Info ",
     " success ":  true ,
     " remoteAddr  ess ": " 1 . 1 . 1 . 1 : 11111 ",
     " usedTime ":  48 ,
     " param ": {
           " projectNam  e ": " ali - log - test - project ",
           " requestId ": " d3f0c96a - 51b0 - 4166 - a850 -
f4175dde73  23 "
     },
     " result ": {
         " message ": " successful ",
         " code ": " 200 ",
         " data ": {
             " clusterReg  ion ": " ap - southeast - 1 ",
             " ProjectNam  e ": " ali - log - test - project ",
             " CreateTime ": " 2017 - 06 - 08   20 : 22 : 41 "
         },
         " success ":  true
     }
```

```
}
```

You can set indexes for a log as follows:

Figure 6-1: Index setting



In the preceding figure:

- Mark ① indicates that the index type for this field is json and all data of the string type and bool type in the field can be queried.
- Mark ② indicates that the index type for this field is long and data of the long type in the field can be queried.
- Mark ③ indicates that the fields can be analyzed by using SQL statements.

Example:

1. Query data of the string type and bool type.
   - You do not need to configure the fields in the json field.
   - JSON maps and arrays are automatically expanded. You can query fields that are multi-level nested with each level separated by a period (.).

```
class : cental *
message . traceInfo . requestId : 92 . 137_151813 9699935_55
99
message . param . projectNam e : ali - log - test - project
```

```
message . success  :  true
```

2. Query data of the Double type and Long type.

   The fields in a JSON field must be configured separately and must not be contained in an array.

```
latency > 40
message . usedTime  >  40
```

3. Query data with combined data types.

```
class  :  cental *  and   message . usedTime  >  40   not   message
. param . projectNam  e : ali - log - test - project
```

6.6.2 Text type

Similar to search engines, text data is queried based on terms. Therefore, you must configure word segmentation, case sensitivity, including options.

Instructions

Case sensitivity

Determine whether to support case sensitivity when querying raw logs. For example, the raw log is internalError .

- After turning off the Case Sensitive switch, the sample log can be queried based on the keyword INTERNALERROR or internalerror .
- After turning on the Case Sensitive switch, the sample log can only be queried based on the keyword internalError .

Token

You can separate the contents of a raw log into several keywords by using a token.

For example, the raw log is

```
/ url / pic / abc . gif
```

- If no token is set, the string is considered as an individual word `/ url / pic / abc . gif` . You can only query this log by using the complete string or fuzzy match such as `/ url / pic /*` .
- If `/` is set as the token, the raw log is separated into three words: `url` , `pic` , and `abc . gif` . You can query this log by using any of the three words or fuzzy match, for example, `url` , `abc . gif` , or `pi *` . You can also use `/ url /`

`pic / abc . gif` to query this log (`url and pic and abc . gif` is separated into the following three conditions during the query: url , pic , and abc.gif).

- If `/.` is set as the token, the raw log is separated into four words: `url` , `pic` , `abc` , and `gif` .



Note:

You can broaden the query range by setting appropriate tokens.

Full text index

By default, full text query (index) considers all the fields and keys of a log, except the time field, as text data, and does not need to specify keys. For example, the following log is composed of four fields (time/status/level/message):

```
[ 20180102 12 : 00 : 00 ] 200 , error , some thing is error
 in this field
```

- time:2018-01-02 12:00:00
- level:” error”
- status:200
- message:” some thing is error in this field”

After enabling full text index, the following text data is assembled in the “key:value + space” mode.

```
status : 200 level : error message :" some thing is error
in this field "
```



Note:

- Prefix is not required for full text query. Enter error as the keyword, both level field and message field meet the query condition.
- You must set a token for the full text query. If a space is set as the token, status:200 is considered as a phrase. If : is set as the token, status and 200 are considered as two independent phrases.
- Numbers are processed as texts. For example, you can use the keyword 200 to query this log. The time field is not processed as a text.
- You can query this log if you enter a key such as ” status" .

6.6.3 JSON type

JSON contains multiple data types, including text, boolean, value, array, and map.

Instructions

Text type

For JSON fields, fields of text type and boolean type are automatically recognized.

For example, the following jsonkey can be queried by using the conditions such as `jsonkey . key1 :" text_value "`.

```
 jsonkey : {
    key1 : text_value ,
    key2 : true ,
    key3 : 3 . 14
}
```

Value type

You can query the double or long type data that is not in the JSON array by setting the type and specifying the path.

For example, the type of the jsonkey.key3 field is double.Then, the query statement is as follows:

```
  jsonkey . key3  >  3
```

JSON field including invalid content

Log Service attempts to parse the valid contents until the invalid content appears.

For example:

```
" json_strin  g ":
{
     " key_1 " : " value_1 ",
     " key_map " :
      {
             " key_2 " : " value_2 ",
             " key_3 " : " valu
```

Data after key_3 is truncated and lost. The field `json_strin  g . key_map . key_2` and contents before this field are successfully parsed.

Instructions

- JSON object type and JSON array type are not supported.
- The field cannot be in a JSON array.
- Boolean fields can be converted to the text type.

Query syntax

To query a specific key, you must add the parent path prefix of JSON in the query statement. The text type and value type of JSON have the same query syntax as those of non-JSON. For more information, see *Query syntax*.

6.6.4 Value type

When configuring indexes, you can configure a field as the value type and query the key by using a value range.

Instructions

Supported types: `long` (long integer) and `double` (decimal). After configuring a field as the value type, you can only query the key by using a value range.

Example

To query the longkey whose key range is (1000 2000], use the following methods.

- Use values to query the longkey:

```
longKey > 1000 and longKey <= 2000
```

- Use an interval to query the longkey:

```
longKey in ( 1000 2000 ]
```

For more syntaxes, see *Query syntax*.

6.7 Query

6.7.1 Query syntax

To help you query logs more effectively, Log Service provides a set of query syntax to express query conditions.

Query methods

After *enabling the index function and configuring indexes*, you can enter a query and analysis statement on the log query page to query logs.

The query statement is the first part of a query analysis statement, and is used to specify rules for filtering logs and logs that conform to the query condition. Both full text query and key/value query are supported.

- Full text query

  In a full text query, the entire log is regarded as a special key/value pair, in which the log content is regarded as the value. You can specify keywords for a full text query. Specifically, you can specify the keywords which must be included in or excluded from the query condition. The log that meets the specified query condition will be returned as a query result.

  Log Service also supports phrase query and fuzzy query.

  - Common full text query: You need to specify a keyword and rule. Logs that contain the keyword and conform to the rule will be returned as query results.

    For example, `a and b` indicates that the query results must contain both the keywords `a` and `b`.
  - Phrase query: If the target phrase contains a space, you can enclose the phrase with double quotation marks (""). In this case, the phrase will be regarded as a complete keyword for log query.

    For example. `" http error "` indicates that the query results must contain `http error`.
  - Fuzzy query: You can specify a partial word up to 64 characters in length, and add a fuzzy query keyword (`*` or `?`) at the middle or end of the word. By doing so, up to 100 words that meet the query condition among all logs will be queried, and the logs corresponding to the 100 words will be returned as query results.

    For example, `addr ?`indicates that Log Service needs to query up to 100 words starting with `addr`, and return the corresponding logs.

- Key/value query

  After configuring indexes for fields, you can query the name or content of a specific field. For fields of the double or longtype, you can also specify the value range for query. For example, the key/value query statement `Latency > 5000 and`

`Method : Get * and not Status : 200` indicates that the query results must meet the following conditions:

- The value of `Latency` must be greater than 5000.
- The `Method` field must start with `Get` .
- The value of the `Status` field is not 200.

You can perform various types of basic query and combined query according to the data types set for field indexes. For more information about key/value query examples, see *Index data type overview*.

Precautions

- When both full text query and key/value query are performed, if the delimiters set for the two query methods are different, the delimiter set for key/value query is used, and the query results of full text query become invalid.
- You can query fields with a specified value range only after setting the data type of the fields to double or long. If the field data type is unspecified or the syntax for querying value ranges is incorrect, Log Service determines that the query condition is for full text query. This may return unexpected query results.
- If the date type of a field is changed from text to numeric, the data collected before the change only support `=` query.

Operators

The following operators can be used in query statements:

Operator	Description
and	Binary operator. Format: query1 and query2. Indicates the intersection of the query results of query1 and query2 . With no syntax keyword among multiple words, the relation is and by default.
or	Binary operator. Format: `query1 or query2` . Indicates the union of the query results of `query1` and `query2` .
not	Binary operator. Format: `query1 not query2` . Indicates a result that matches `query1` and does not match `query2` , which is equivalent to `query1 - query2` . If only `not query1` exists, it indicates to select the results excluding `query1` from all the logs.

Operator	Description
(,)	Parentheses () are used to merge one or more sub-queries into one query condition to increase the priority of the query in the parentheses ().
:	Used to query the key-value pairs. `term1 : term2` makes up a key-value pair. If the key or value contains reserved characters such as spaces and colons (:), use quotation marks (“) to enclose the entire key or value.
“	Converts a keyword to a common query character. Each term enclosed in quotation marks (“) can be queried and is not be considered as a syntax keyword. Or all the terms enclosed in quotation marks (“) are regarded as a whole in the key-value query.
\	Escape character. Used to escape quotation marks. The escaped quotation marks indicate the symbols themselves, and they cannot be used as escape characters, such as `"\""`.
|	The pipeline operator indicates more calculations based on the previous calculation, such as query1 | timeslice 1h | count.
timeslice	The time-slice operator indicates how long the data is calculated as a whole. Timeslice 1h, 1m, 1s indicates 1 hour, 1 minute, and 1 second respectively. For example, query1 | timeslice 1h | count represents the query query condition, and returns to the total number of hours divided by 1 hour.
count	The count operator indicates the number of log lines.
*	Fuzzy query keyword. Used to replace zero or multiple characters. For example, the query results of `que *` start with `que` .  Note: At most 100 query results can be returned.
?	Fuzzy query keyword. Used to replace one character. For example, the query results of `qu ? ry` start with `qu` , end with `ry` , and have a character in the middle.
`__topic__`	Topic data query. You can query the data of zero or multiple topics in the query. For example, `__topic__ : mytopicnam e` .

Operator	Description
`__tag__`	Query a tag value in a tag key. For example, `__tag__ : tagkey : tagvalue`.
Source	Query the data of an IP. For example, `source : 127 . 0 . 0 . 1`.
>	Query the logs with a field value greater than a specific number. For example, `latency > 100`.
>=	Query the logs with a field value greater than or equal to a specific number. For example, `latency >= 100`.
<	Query the logs with a field value less than a specific number. For example, `latency < 100`.
<=	Query the logs with a field value less than or equal to a specific number. For example, `latency <= 100`.
=	Query the logs with a field value equal to a specific number. For example, `latency = 100`.
in	Query the logs with a field staying within a specific range. Braces ([]) are used to indicate closed intervals and parentheses (()) are used to indicate open intervals. Enclose two numbers in braces ([]) or parentheses (()) and separate the numbers with several spaces. For example, `latency in [ 100 200 ]` or `latency in ( 100 200 ]]`.

Note:

- Operators are case-insensitive.
- Priorities of operators are sorted in descending order as follows: `:>">( )> and > not > or`.
- Log Service reserves the right to use the following operators: `sort`, `asc`, `desc`, `group by`, `avg`, `sum`, `min`, `max`, and `limit`. To use these keywords, enclose them in quotation marks ("").

Query examples

Query demand	Example
Logs that contain a and b at the same time	`a and b` or `a b`

Query demand	Example
Logs that contain a or b	`a or b`
Logs that contain a but do not contain b	`a not b`
All the logs that do not contain a	`not a`
Logs that contain a and b, but do not contain c	`a and b not c`
Logs that contain a or b and must contain c	`( a or b ) and c`
Logs that contain a or b, but do not contain c	`( a or b ) not c`
Logs that contain a and b and may contain c	`a and b or c`
Logs whose FILE field contains apsara	`FILE : apsara`
Logs whose FILE field contains apsara and shennong	`FILE :" apsara shennong ", FILE : apsara FILE : shennong` , or `FILE : apsara and FILE : shennong`
Logs that contain and	`and`
Logs with the FILE field containing apsara or shennong	`FILE : apsara or FILE : shennong`
Logs with the file info field containing apsara	`" file info ": apsara`
Logs that contain quotation marks ("")	`\"`
All logs starting with shen	`shen *`
All logs starting with shen in the FILE field	`FILE : shen *`
All logs with the FILE field of shen*	`FILE : " shen *"`
Logs starting with shen, ending with ong , and having a character in the middle	`shen ? ong`
Logs starting with shen and aps	`shen * and aps *`
Logs starting with shen every 20 minutes	`shen *| timeslice 20m | count`
All data in the topic1 and topic2	`__topic__ : topic1 or __topic__ : topic2`

Query demand	Example
All data of the tagvalue2 in the tagkey1	`__tag__ : tagkey1 : tagvalue2`
All data with a latency greater than or equal to 100 and less than 200	`latency >= 100 and latency < 200` or `latency in [ 100 200 )`
All requests with a latency greater than 100	`latency > 100`
Logs that do not contain spider and do not contain opx in http_referer	`not spider not bot not http_refer er : opx`
Logs with the empty cdnIP field	`not cdnIP :""`
Logs without cdnIP field	`not cdnIP :*`
Logs with the cdnIP field	`cdnIP :*`

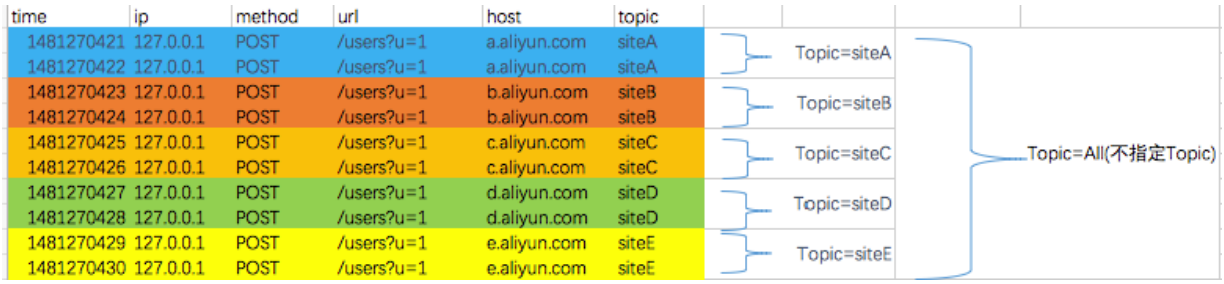
Specified or cross-topic query

Each LogStore can be divided into one or more subspaces by the topic. During therfhfrg query, specifying topics can limit the query range so as to increase the speed. Therefore, we recommend that you use topic to divide the LogStore if you have a secondary classification requirement for the LogStore.

With one or more topics specified, the query is only performed in the topics that meet the conditions. However, if no topic is specified, data of all the topics is queried by default.

For example, use topic to classify logs with the different domain names:

Figure 6-2: Log topic



Topic query syntax:

- Data of all the topics can be queried. If no topic is specified in the query syntax and parameter, data of all the topics is queried.

- Supports query by topic. The query syntax is `__topic__ : topicName`. The old mode (specify the topic in the URL parameter) is still supported.
- Multiple topics can be queried. For example, `__topic__ : topic1 or __topic__ : topic2` indicates the union query of data from Topic1 and Topic2 .

Fuzzy search

Log Service support fuzzy search. Specify a word within 64 characters, and add fuzzy search operators such as `*` and `?` in the middle or in the end of the word. 100 eligible words will be searched out, in the meantime, all the logs eligible and contain the 100 words will be returned.

Limits:

- Prefix must be specified when query logs, that is, the word can not begin with `*` and `?`.
- Precise the specified word, you will get a more accurate result.
- Fuzzy search cannot be used to search for words that exceeds 64 characters. It is recommended that you specified a word under 64 characters.

6.7.2 LiveTail

LiveTail is an interactive function provided by Log Service in the console to help you monitor logs in real time and extract key log information.

Scenarios

In scenarios of online Operation & Maintenance (O&M), it is often necessary to monitor inbound data of the log queue in real time, and to extract key information from the latest log data to quickly find the cause of the exception. By using the traditional O&M method, you need to run the `tail -f` command on log files on the server to monitor the log files in real time. If the log information you require is not apparent enough, you can add `grep` or `grep -v` to the command to filter keywords. Log Service provides LiveTail in the console, an interactive function that monitors and analyzes online log data in real time, making O&M easier.

Benefits

- Monitors real-time log information, and marks and filters keywords.
- Distinguishes collected logs by using indexes through the collection configuration.

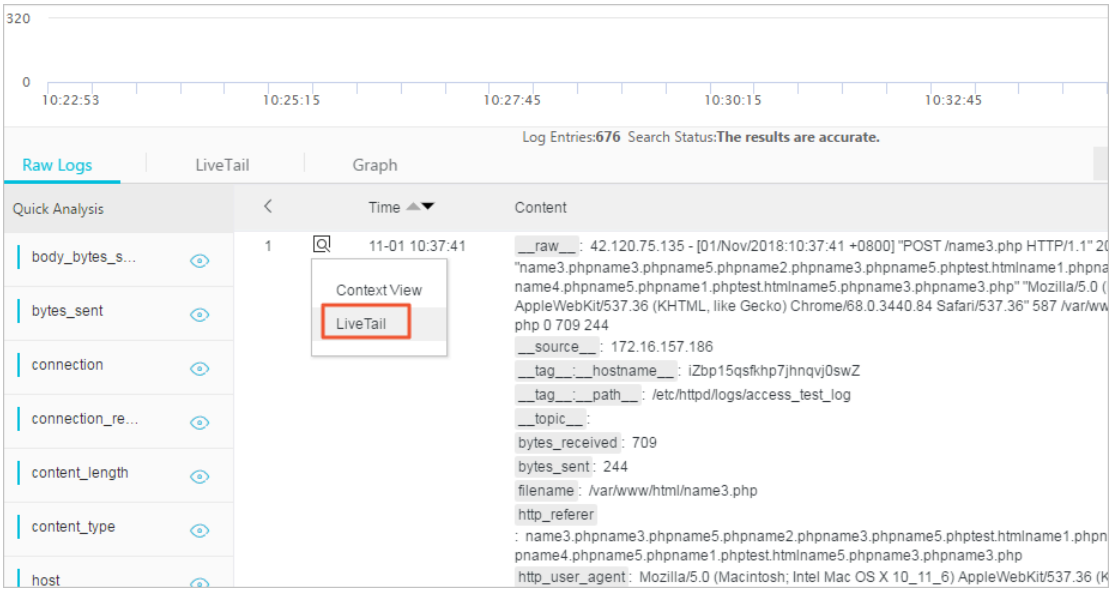
- Perform word segmentation for log fields to query the context logs that contain segmented words.
- Tracks the log file for real-time monitoring according to a single log entry without the need to connect to the server.

Limits

- LiveTail is only applicable to the logs collected by Logtail.
- LiveTail is available only when logs are collected.

Use LiveTail to monitor logs in real time

1. Click Search in the LogSearch column.
2. You can use LiveTail in one of the following two ways:
   - Quickly start LiveTail.
     a. On the Raw Logs tab, click the icon on the right of the sequence number of the raw log, and select LiveTail.

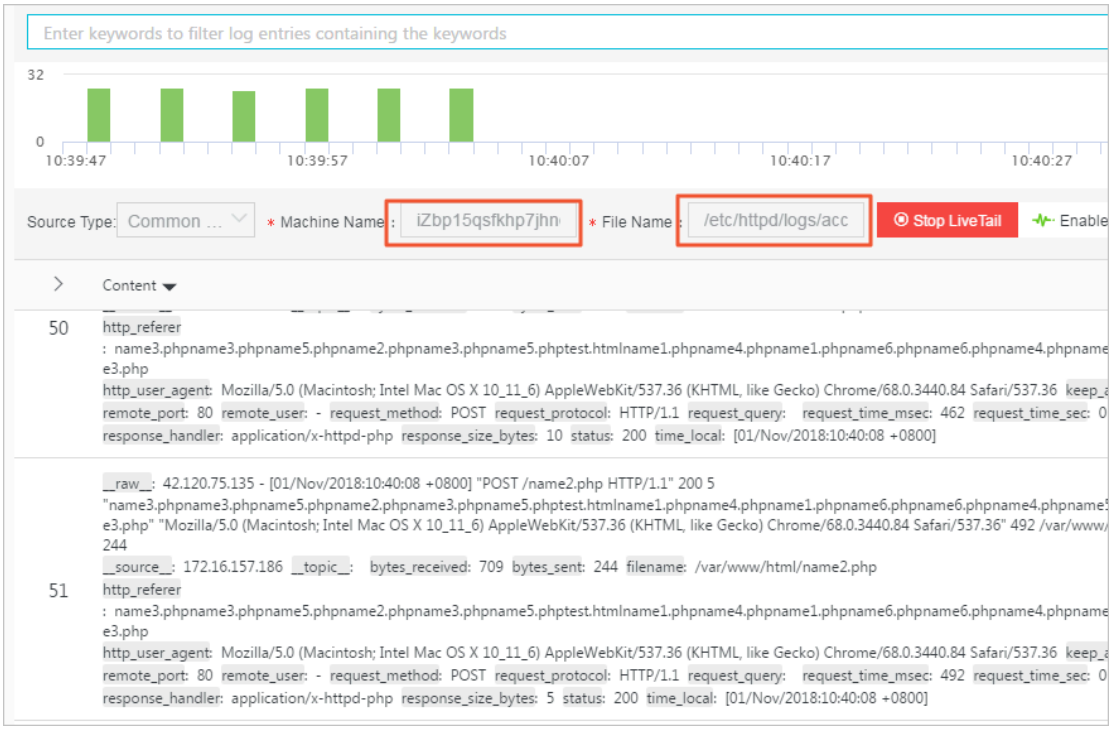


     b. The system automatically starts LiveTail and starts timing.

     Source Type, Machine Name, and File Name are pre-configured to specify the raw logs.

     After LiveTail is started, the log data collected by Logtail is displayed in order on the page. The latest log data is always displayed at the bottom of the page. The scrollbar is at the lowest position on the page by default so that you can immediately see the latest data. The page displays up to 1000 log entries

. When 1000 log entries are displayed, the page automatically refreshes to display the latest log entry at the bottom of the page.



c. (Optional) Enter keywords in the search box.

Only log entries that contain the keywords can be displayed in the monitoring list. By filtering logs that contain the keywords, you can monitor the content of the logs in real time.

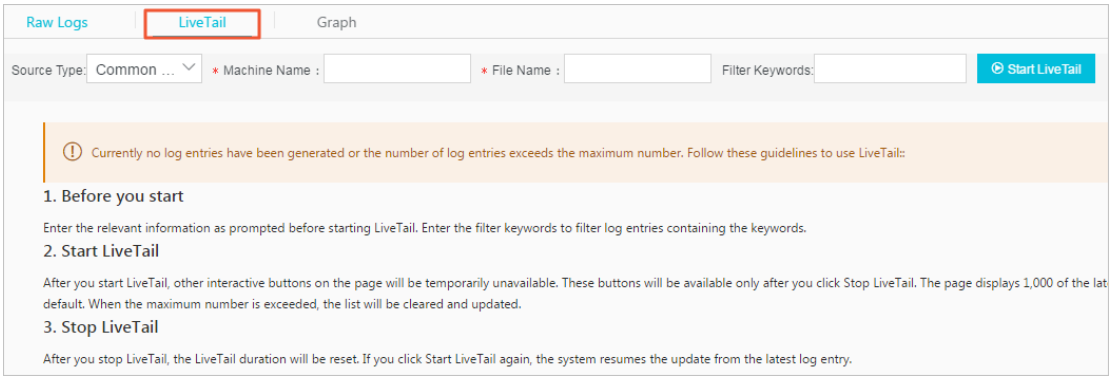
d. To analyze logs in which exceptions may exist during the real-time log monitoring process, click Stop LiveTail.

After you stop LiveTail, the LiveTail timing and the real-time log data update also stop.

For exceptions found in the process of log monitoring, Log Service provides multiple analysis methods. For more information, see *Use LiveTail to analyze logs*.

- Customize LiveTail settings.

a. Click the LiveTail tab.



b. Configure LiveTail.

Configuration	Required	Description
Source type	Yes	Log source, including: - Common log - Kubernetes - Docker
Machine name	Yes	Name of the log source server.
File name	Yes	Full path and file name of the log file.
Filter keywords	No	Keywords. After you configure a keyword, only the logs that contain the keyword can be displayed in the real-time monitoring window.

c. Click Start LiveTail.

After LiveTail is started, log data collected by Logtail are displayed in orders on the page. The latest log data is always displayed at the bottom of the page . The scrollbar resides at the lowest position of the page by default so that you can see the latest data. The page displays up to 1000 log entries. When 1000 log entries are displayed, the page automatically refreshes to display the latest log entry at the bottom of the page.

d. To analyze logs in which exceptions may exist during the real-time log monitoring process, click Stop LiveTail.

After you stop LiveTail, the LiveTail timing and the real-time log data update stop as well.

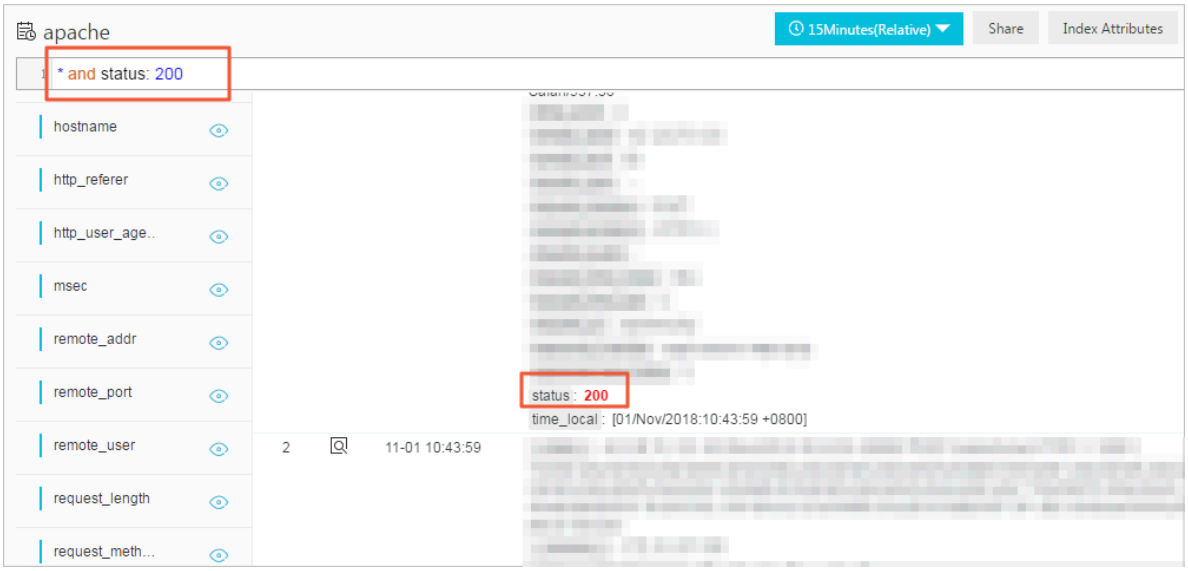
For exceptions found in the process of log monitoring, Log Service provides multiple analysis methods. For more information, see *Use LiveTail to analyze logs*.

Use LiveTail to analyze logs

After you stop LiveTail, the real-time monitoring window stops updating logs, and you can analyze and troubleshoot the exceptions found in the monitoring process.

- View the logs that contain the specified field.

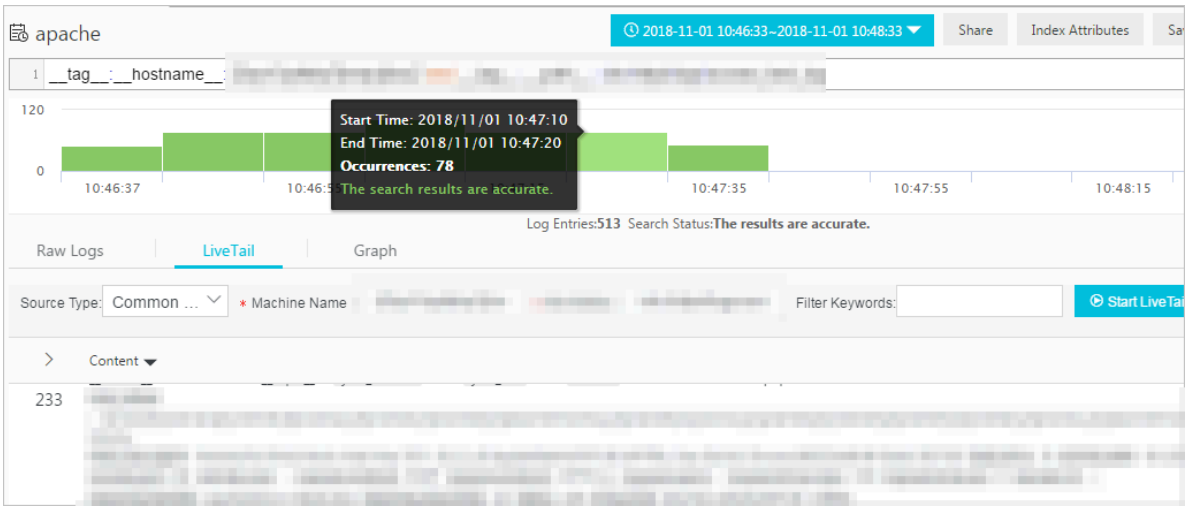
  Word segmentation has been conducted to all fields. When you click the exception field content, that is, a keyword, the page automatically jumps to the Raw Logs tab, and the system filters all logs to show the logs that contain the keyword. In addition, you can also analyze the logs that contain the keyword by using context view, statistical charts, and other analysis methods.



- Narrow the time range of a query according to the log distribution histogram.

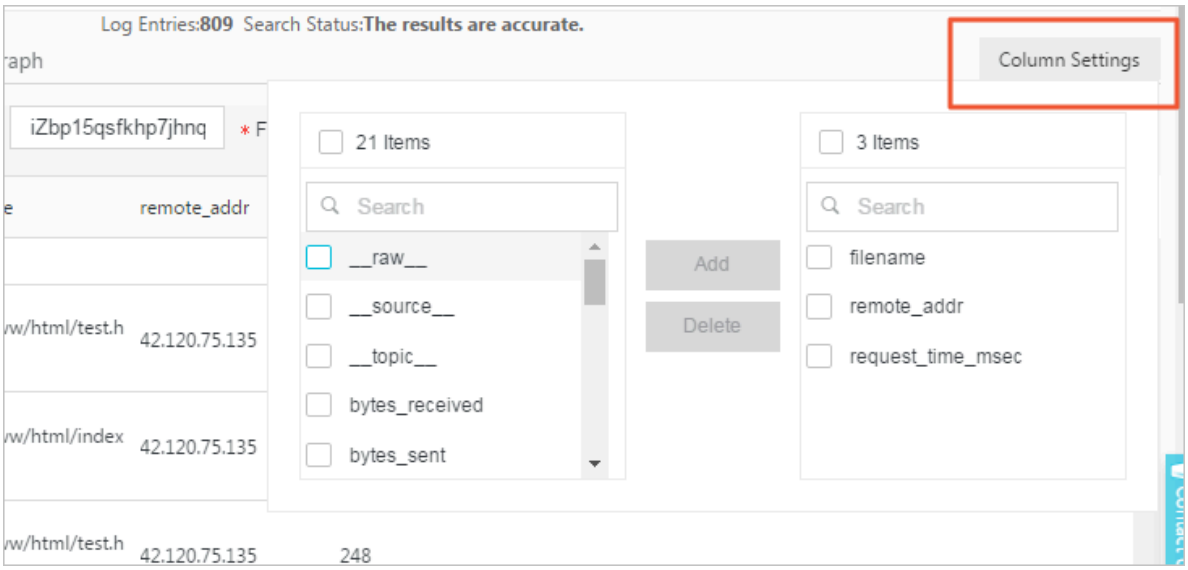
  When LiveTail is started, the log distribution histogram is also updated synchronously. If you find an exception of log distribution for a time period, for example a significant increase in the number of logs, you can click the green rectangle of the time period to narrow the time range of the query. The timeline of the raw logs redirected from the LiveTail page is associated with the timeline clicked in LiveTail

. You can view all the raw logs and the detailed log distribution over time during this time period.



- Highlight key information with column settings.

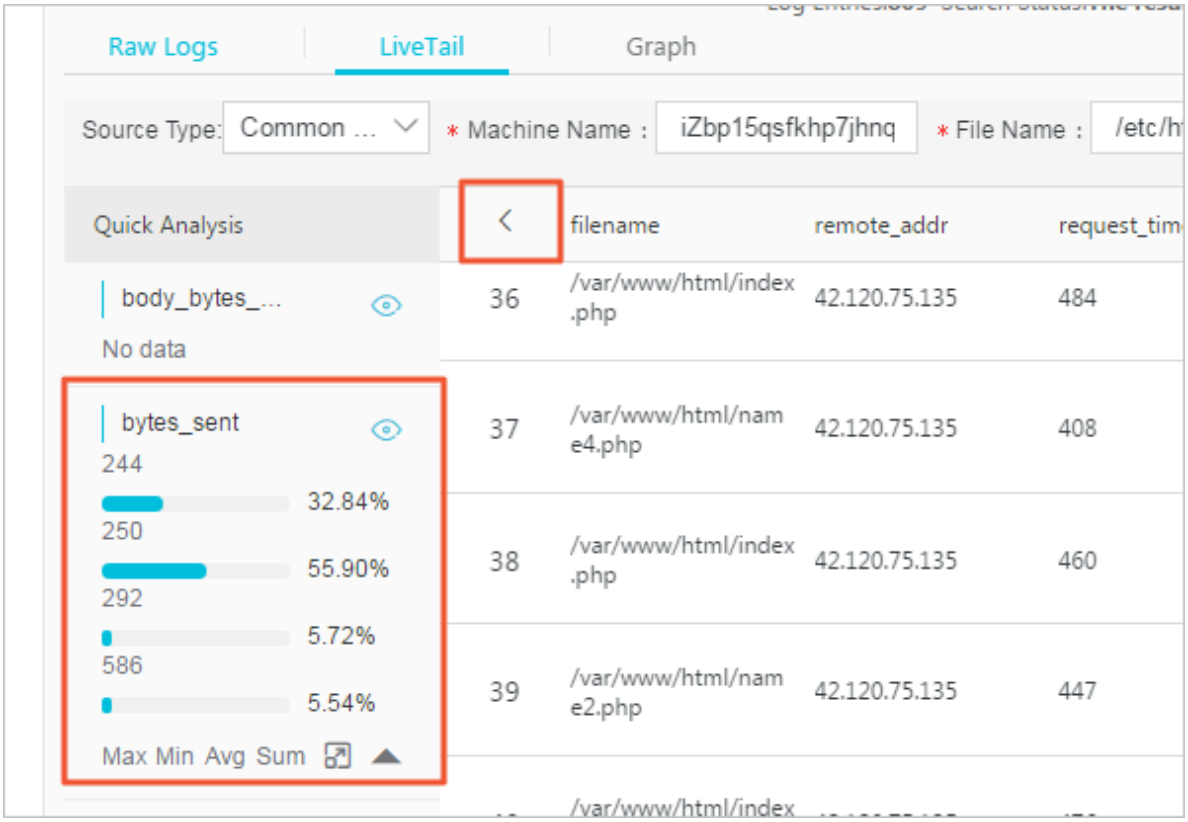
  On the LiveTail tab, click Column Settings in the upper-right corner of the log list, you can set a specified field as a separate column to make the data in this column more obvious. You can configure the data that requires high attention as one column to make it easier to view and recognize exceptions.



- Quickly analyze log data.

  On the LiveTail tab, by clicking the arrow in the upper-left corner of the log list, you can expand the quick analysis area. The time interval of the quick analysis is the period from when LiveTail starts to when it stops. The quick analysis provided

in LiveTail is the same as that provided in the raw logs. For more information, see *Quick analysis*.



6.7.3 Use LogReduce to group log data

This topic describes how to use LogReduce to group and analyze collected log entries that are extremely similar to detect frequently occurring log patterns such as conditions that trigger alarms.

Scenario

With LogReduce, you can locate problems, detect exceptions, and perform other O&M-related actions for DevOps, or detect network intrusions that may have compromised the security of your services. In addition, you can save the log grouping result as an analysis chart to a dashboard, and then view the grouped data in real time.

Benefits

- Log entries in three formats (Log4J, JSON, or Syslog) can be grouped by using the LogReduce function.
- Gigabytes of data can be grouped in seconds.

- You can view the log entries that are grouped for each log pattern, and you can display the number of grouped log entries during different time ranges.
- You can dynamically adjust the tolerance of log grouping.

Index size

> Note:
> After you enable the LogReduce function, the size of log indexes increases by 10%. For example, if the size of raw log data is 100 GB/day, the size of the log indexes increases by 10 GB after you enable the function.

Raw log size	Proportion of indexes in the raw log	Size of indexes generated by LogReduce	Index size
100 GB	20% (20 GB)	100 * 10%	30 GB
100 GB	40% (40 GB)	100 * 10%	50 GB
100 GB	100% (100 GB)	100 * 10%	110 GB

Enable LogReduce

> Note:
> By default, LogReduce is disabled.

1. Log on to the *Log Service console*, and then click the target project name.
2. On the Logstores page, click Search on the right of the target Logstore.

3. If you have enabled the index function, choose Index Attributes > Modify. If you have not enabled the index function, click Enable.

Figure 6-3: Enable the index function

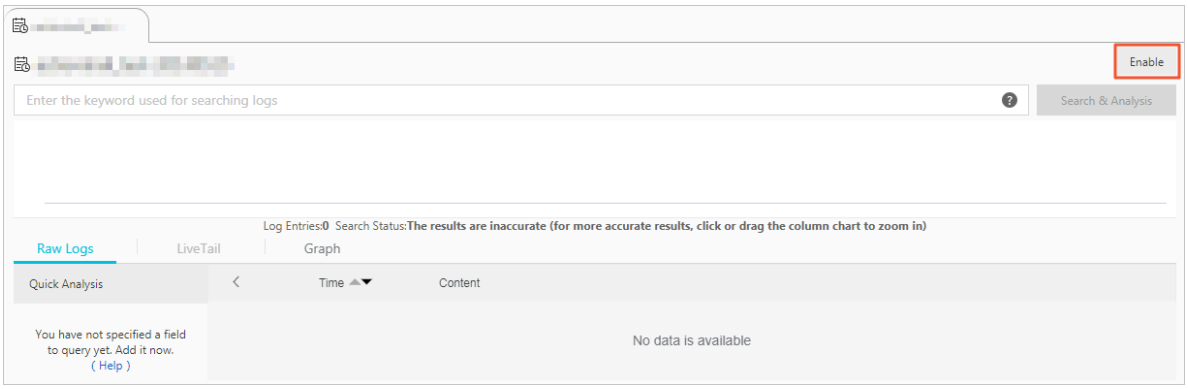
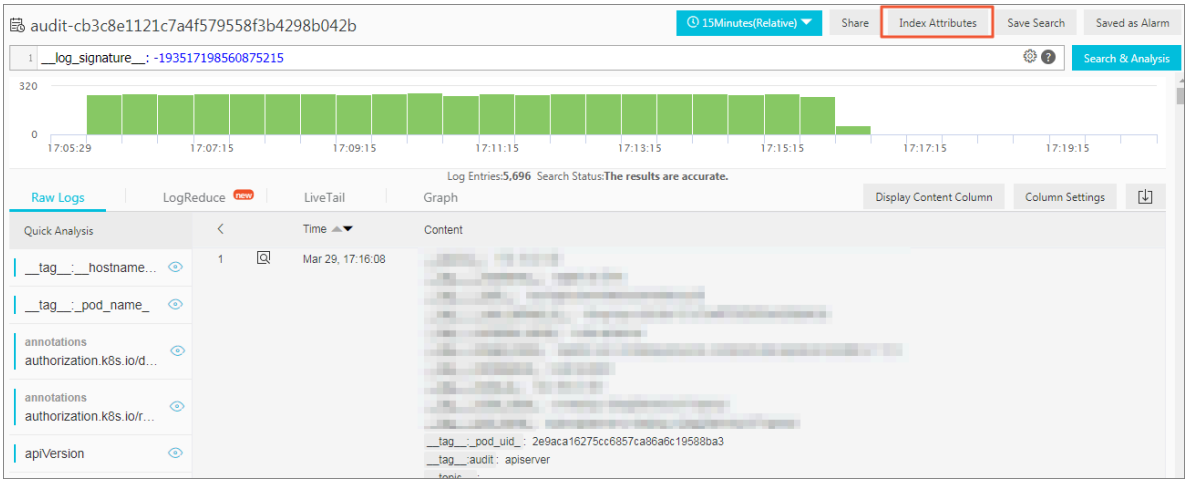
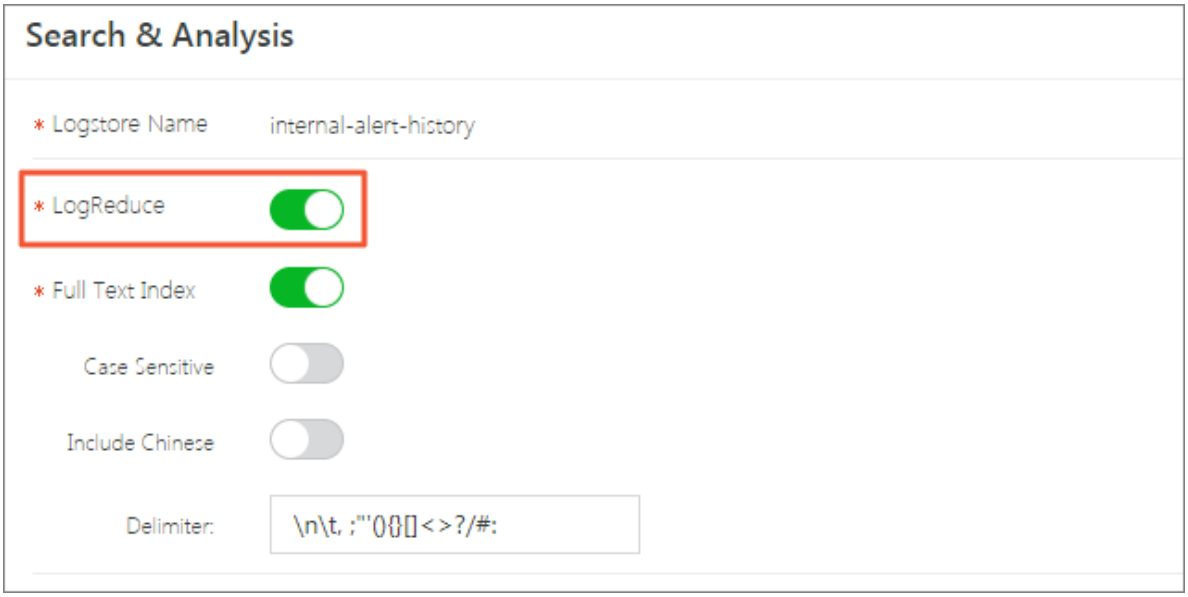


Figure 6-4: Modify the log index



4. Set index parameters, and click the switch to enable LogReduce.

   Figure 6-5: Enable LogReduce



5. Click OK.

   Note:
   After you enable LogReduce, Log Service automatically groups collected log data.

   Then, you can perform the following operations:

   - *View the log grouping result and the raw log*.
   - *Adjust the log grouping precision*.
   - *Show the number of grouped log entries in different periods*.

View the log grouping result and the raw log

1. On the Search & Analysis page, enter a search and analysis statement in the search box, and click Search & Analysis.

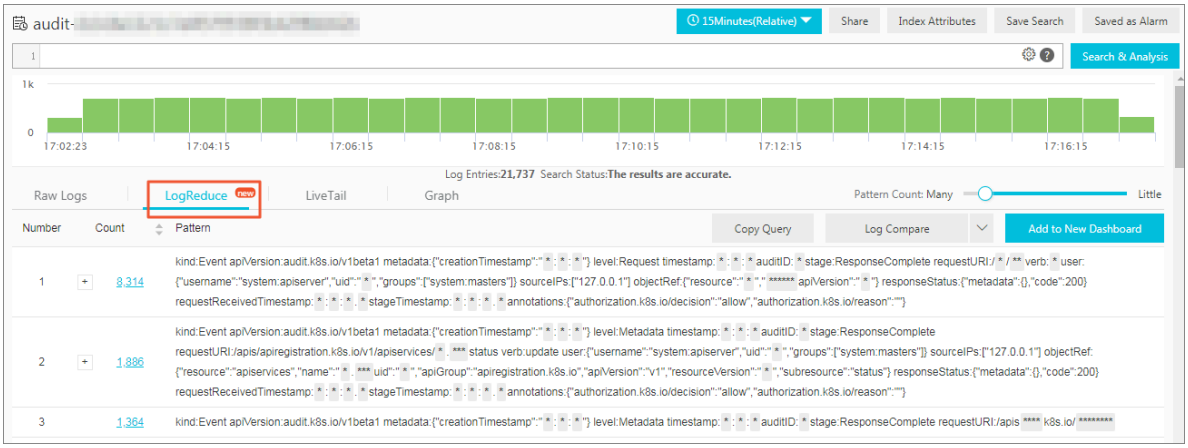
   Note:

   - You can also use key words to filter the grouped log entries.
   - The SQL type of statements is not supported by the LogReduce function. This means that analysis results of log data cannot be grouped by this function.

2. Click the LogReduce tab to view the result.

Item	Description
Number	Indicates the sequence number of a log group.
Count	Indicates the number of log entries of a log group.
Pattern	Indicates the log patterns. Each log group has one or more than one sub-patterns.

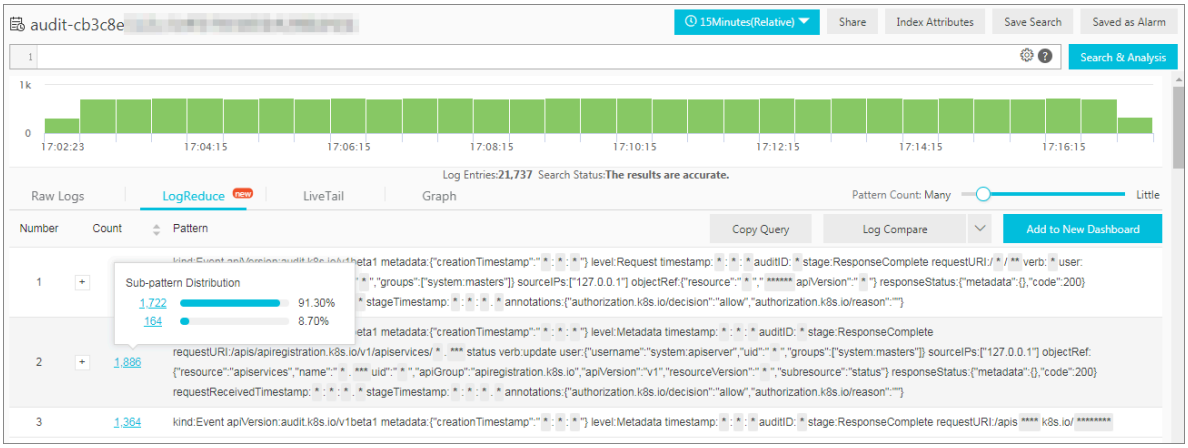
Figure 6-6: Result



3. Move your pointer over a Count value to show the sub-patterns of this log group and the proportion of each sub-pattern in the log group.

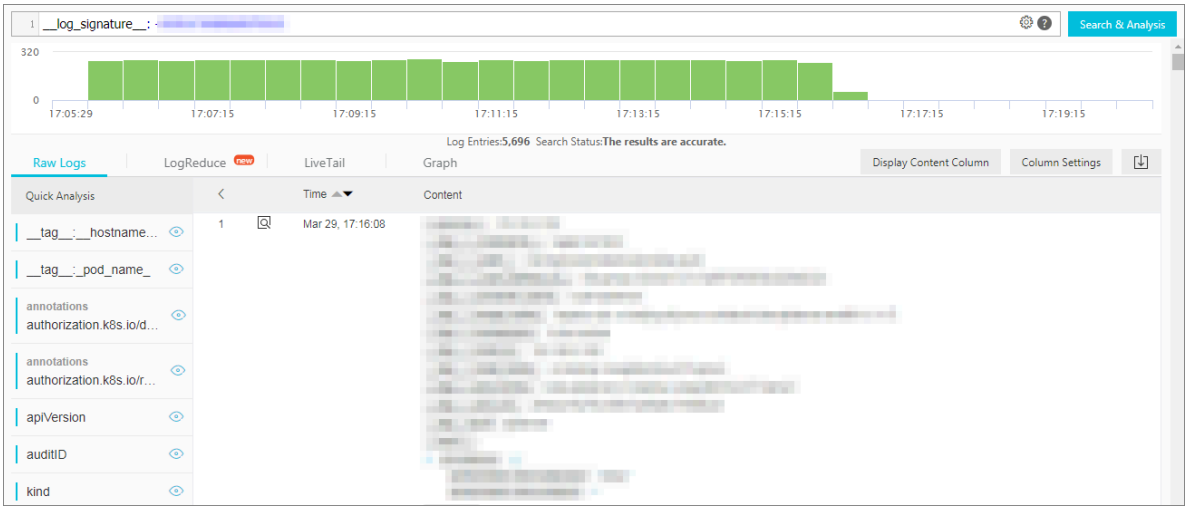
> **Note:**
> You can also click + in front of a Count value to show the pattern list of the log group.

Figure 6-7: View log grouping details



4. Click a Count value to view the raw log of the log group.

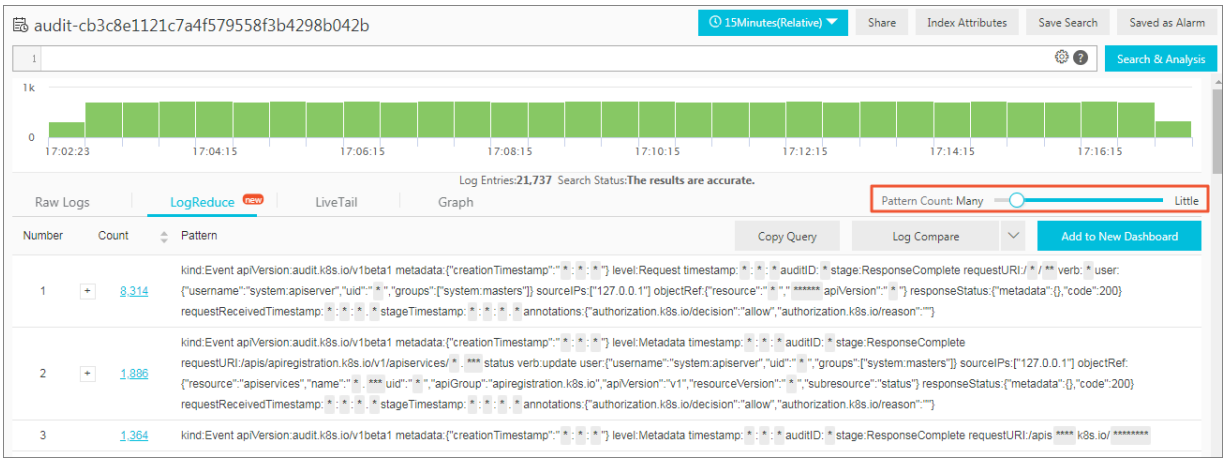
Figure 6-8: View the raw log



Adjust the log grouping tolerance

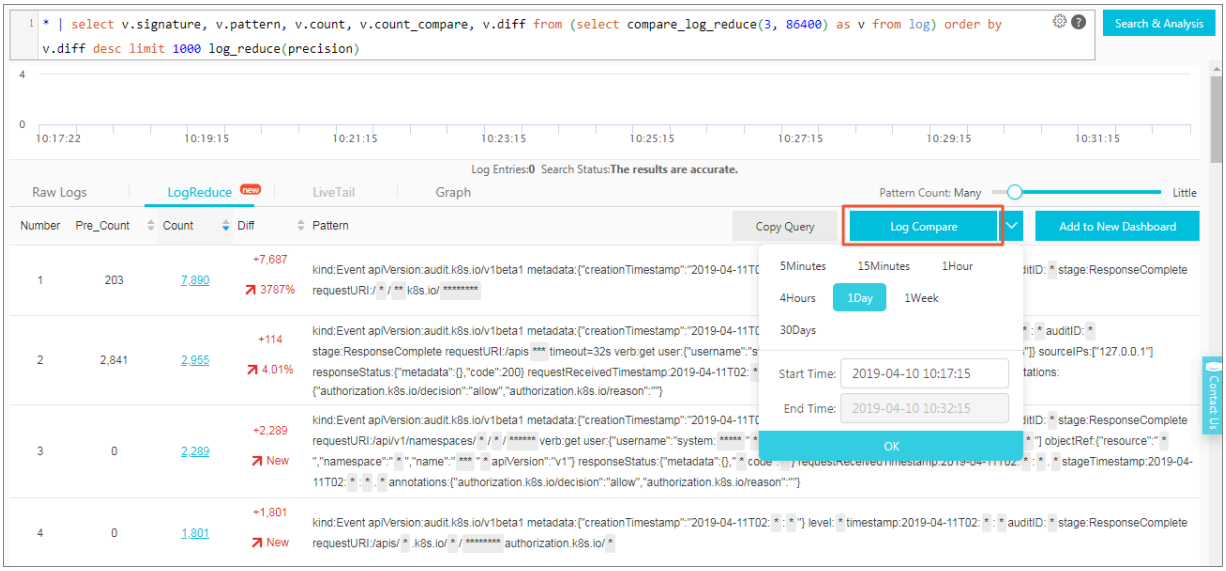
1. On the Search & Analysis page, click the LogReduce tab.
2. In the upper-right corner of the tab page, drag the Pattern slider to adjust the log grouping tolerance.
   - If you drag the slider towards Many, the system outputs a more specific log grouping result and shows patterns in greater detail.
   - If you drag the slider towards Little, the system outputs a less specific log grouping result and shows patterns in less detail.

Figure 6-9: Adjust the log grouping tolerance



Compare the number of group log entries during different periods of time

Click Log Compare, select a time length, and then click OK.



Item	Description
Number	Displays the sequence number of a log group.
Pre_Count	Displays the number of log entries during a time range.
Count	Displays the number of log entries for the log pattern for the current time range.
Diff	Displays the difference in the number of log entries for the log pattern for the current time range and a past time range.
Pattern	Displays the log pattern.

Use the LogReduce function through API

- To obtain a log grouping result, execute the following SQL statement:

```
* | select a.pattern, a.count, a.signature, a.origin_signatures from (select log_reduce(3) as a from log) limit 1000
```

Note:

If you directly view the result through the Log Service console, you can click Copy Query to get the SQL statement executed by the system in the backend.

Parameter and field description

- The parameter in the SQL statement that you need to customize is `log_reduce ( precision )`.

  This parameter must be set to an integer that is in the range of 1 to 16. Its default value is 3. A lower tolerance value outputs a grouping result of a higher tolerance, and more log patterns.
- The execution result of the SQL statement contains the following returned fields:
  - `pattern` : indicates the sub-patterns of log entries in a log group.
  - `count` : indicates the number of log entries in a log group.
  - `signature` : indicates the log pattern of a log group.
  - `origin_sig natures` : indicates the original signature of a log group. You can use this field to search the log entries of this log group.

· To show the difference of log grouping results between different times, execute the following SQL statement:

```
* | select v . pattern , v . signature , v . count , v .
 count_comp are , v . diff from ( select compare_lo g_reduce
 ( 3 , 86400 ) as v from log ) order by v . diff desc
  limit 1000
```

 Note:

If you click Log Compare in the Log Service console to show the difference of log grouping results between different times, the system then executes an SQL statement for the log entries. You can click Copy Query to get the SQL statement.

Parameter and field description

- The parameter in the SQL statement that you need to customize is `compare_log_reduce ( precision , compare_interval )`.
  - The tolerance parameter must be an integer that is in the range of 1 to 16. Its default value is 3. A lower tolerance value outputs a grouping result of a higher tolerance, and more log patterns.
  - The `compare_interval` parameter indicates the number of seconds before which the log entries to be compared with was generated. This parameter must be set as a positive integer.
- The execution result of the SQL statement contains the following returned fields:
  - `pattern` : indicates the sub-patterns of log entries in a log group.
  - `signature` : indicates the log pattern of a log group.
  - `count` : indicates the number of log entries in a log group.
  - `count_compare` : indicates the number of log entries for a log group of the same log pattern within the specified time range.
  - `diff` : indicates the difference between the count field value and the count_compare field value.

6.7.4 Context query

When you expand a log file, each log records an event. Generally, logs are not independent from each other. Several consecutive logs allow you to view the process of a whole event in sequence.

Log context query specifies the log source (machine + files) and a log in the log source. It also queries several logs before and after the log in the original log file, providing a helpful method for troubleshooting the problem in the DevOps scenario.

The Log Service console provides a query page, you can view the context information of the specified log in the original file in the console. It is similar to paging up and down in the original log file. By viewing the context information of a specified log, you can quickly locate the problem.

Scenarios

For example, the O2O take-out website will record the transaction track of a order in the program log on the server:

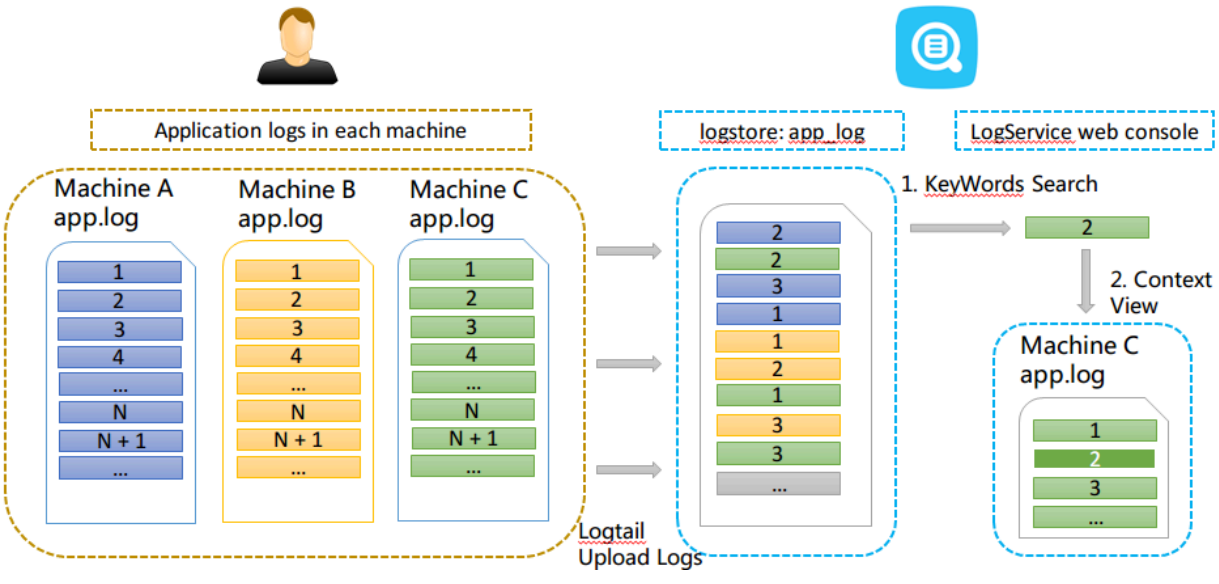
User logon > Browse products > Click items > Add to shopping cart > Place an order > Pay for the order > Deduct payment > Generate an order

If the order cannot be placed, the Operation & Maintenance (O&M) personnel must quickly locate the cause of the problem. In the conventional context query, the administrator grants the machine logon permission to related members, and then the investigator logs on to each machine where applications are deployed in turn, uses the order ID as the keyword to search application log files, and determines what causes the failure.

In Log Service, you can troubleshoot the problem by following these steps:

1. Install the log collection client Logtail on the server, and add the machine group and log collection configuration in the console. Then, Logtail starts to upload the incremental logs. You can also use producer-related SDK uploads, such as Log4J, LogBack, C-Producer
2. On the log query page in the Log Service console, specify the time range, and find the order failure log according to the order ID.
3. Based on the found error log, page up until other related logs are found (for example, the deduction failure of credit card).

Figure 6-10: Scenarios



Benefits

- No intrusion into the application. No need to modify the log file format.
- You can view the log context information of any machine or file in the Log Service console, without logging on to each machine to view the log file.
- Combined with the time when the event occurred, you can specify the time range to quickly locate the suspicious log and then query its context information in the Log Service console to improve the efficiency.
- No need to worry about the data loss caused by insufficient server storage space or log file rotation. You can view historical data in the Log Service console at any time.

Prerequisites

- *Use Logtail to collect logs* . Upload data to the Logstore. Create the machine groups and collection configuration. No other configurations are needed. You can also use producer-related SDK upload, such as Producer Library.
- Enable the Query logs function.

> **Note:**
> Currently, you cannot query the context information of syslog data.

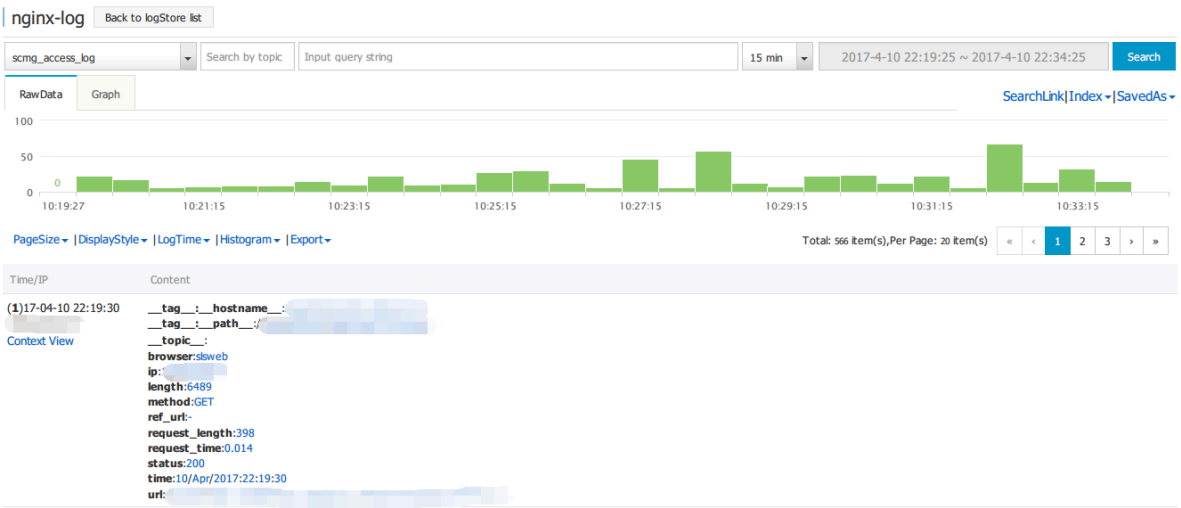
Procedure

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Query at the right of the Logstore to enter the query interface.

4. Enter your query and analysis statement and select the time range. Then, click Search.

   Click Context View at the left side of the log, and the window with the context information of the target log is displayed on the right.

   Figure 6-11: Query log



5. Select a log and click Context View. View the context log for the target log on the right pop-up page.
6. Scroll with the mouse on the page to view the context information of the selected log. To view more context logs, click Earlier or Later.

6.7.5 Saved search

Saved search is a one-click query and analysis function provided by Log Service.

Prerequisites

You have enabled and configured Index.

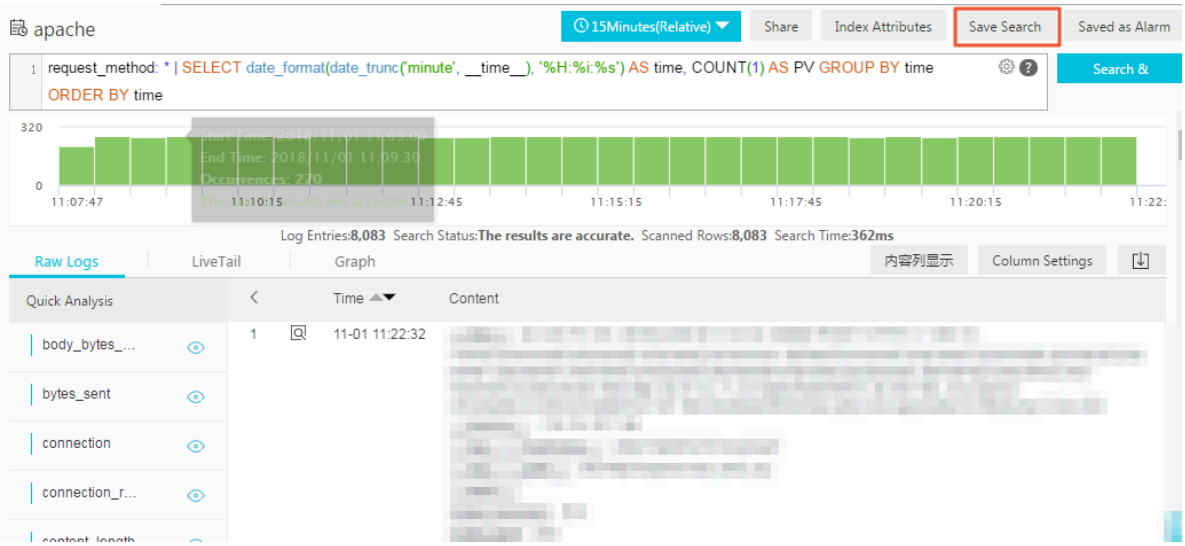
Context

If you need to frequently view the results of a query and analysis statement, save the statement as a saved search. In later result searches, you only need to click the name of the saved search on the left side of the search page. You can also use this saved search condition in alarm rules. Log Service executes the statement of this saved search periodically and sends an alarm notification when the search result meets the pre-configured condition of the statement.

To configure a drill-down event to jump to a saved search when configuring *Drill-down analysis*, you must pre-configure a saved search and set a placeholder in the query statement.

Procedure

1. Log on to the *Log Service console*, and then click the target project name.
2. Click Search in the LogSearch column on the Logstores page.
3. Enter your query analysis statement, set the time range, and click Search & Analysis.
4. Click Save Search in the upper-right corner of the page.



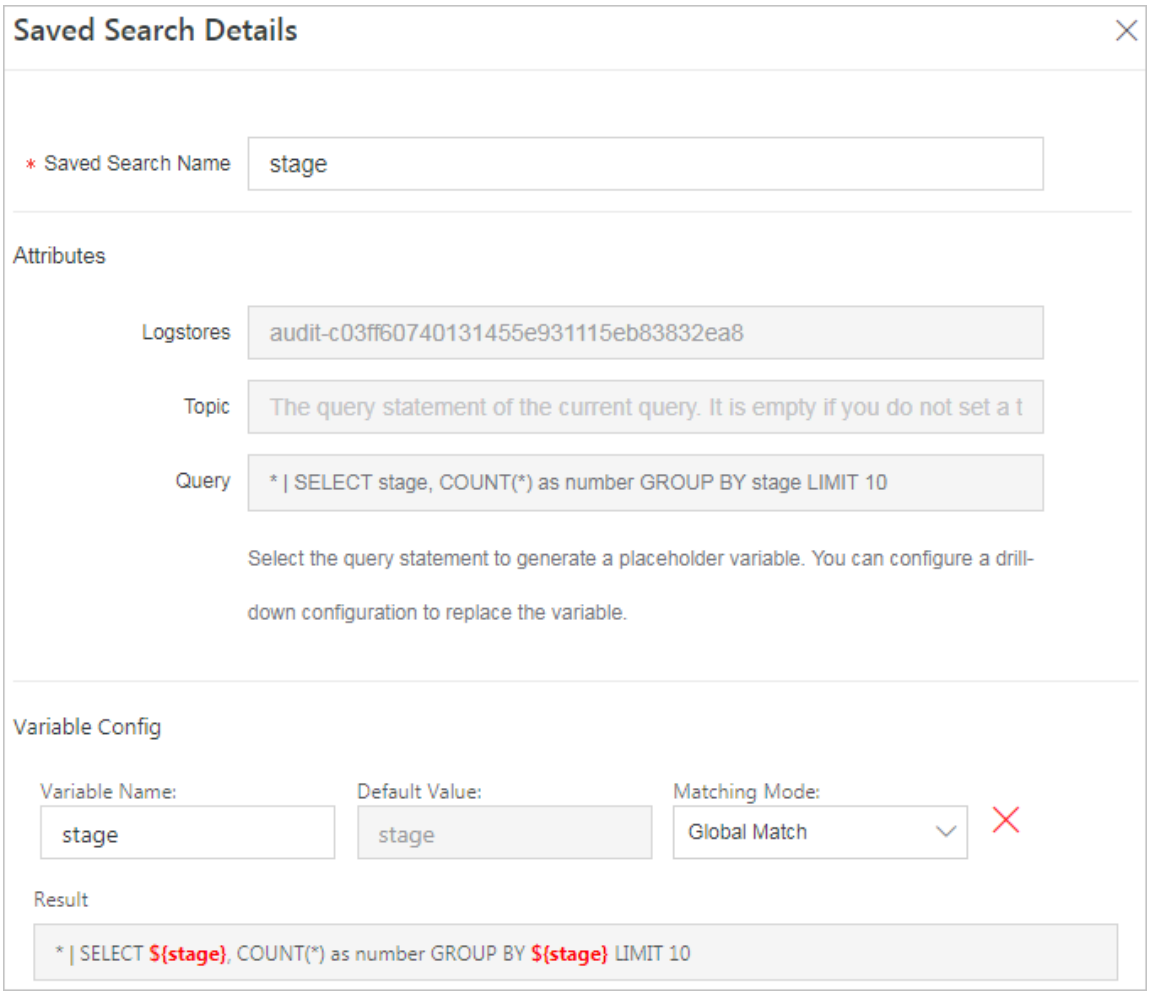
5. Configure saved search attributes.
   a) Set Saved Search Name.
      - The name can only contain lowercase letters, numbers, hyphens (-), and underscores (_).
      - The name must start and end with a lowercase letter or a number.
      - The name must be a string of 3 to 63 characters.
   b) Confirm Logstores, Topic, and Query.

      If Logstore and Topic do not meet your requirements, return to the search page to access the proper Logstore and enter your query statement, and then click Save Search again.
   c) Optional: Select part of the query statement and click Generate Variable.

      The generated variable is a placeholder variable. Name the placeholder in the Variable Name box. Default Value is the selected word.

       Note:
      If the drill-down event of a chart is to jump to the saved search and the chart has the same variable as this saved search, clicking the chart triggers the jump. Additionally, the default value of the placeholder variable is replaced with the chart value that triggers the drill-down event, and the query statement with

the variable replaced is used for querying. For more information, see *Drill-down analysis*.



6. Click OK to end the configuration.

6.7.6 Quick analysis

The quick analysis function of Log Service supports an interactive query with only one click, allowing you to quickly analyze the distribution of a field over a period of time and reduce the cost of indexing key data.

Functions and features

- Support grouping statistics for the first 10 of the first 100,000 pieces of data of `Text` fields.
- Support generating `approx_distinct` statements quickly for `Text` fields.
- Support histogram statistics for the approximate distribution of `long` or `double` fields.

- Support the quick search for the maximum, minimum, average, or sum of `long` or `double` fields.
- Support generating query statements based on quick analysis and query.

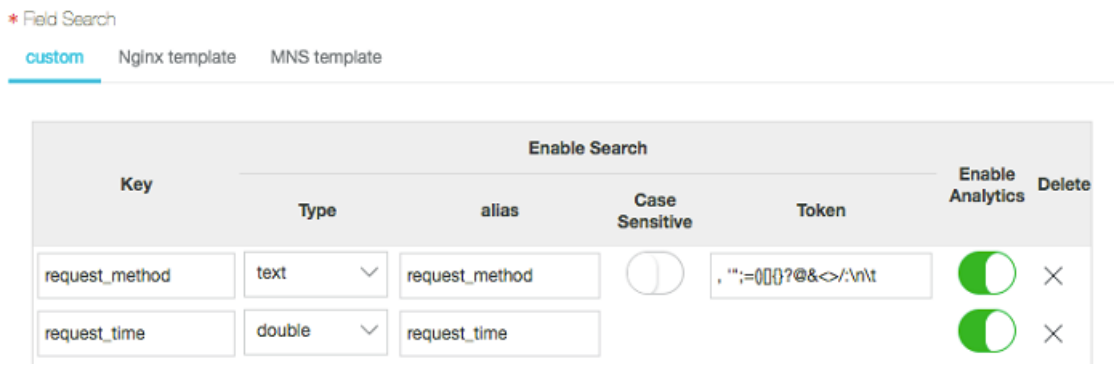
Prerequisite

You must specify the field query properties before using the quick analysis.

1. For specified field query, you must enable the index to activate the query and analysis function. For how to enable the index, see *Query and analysis*.
2. Set the `key` in the log as the field name and set the type, alias, and separator.

   If the access log contains the `request_me thod` and `request_ti me`, you can configure the following settings.

   Figure 6-12: Prerequisites

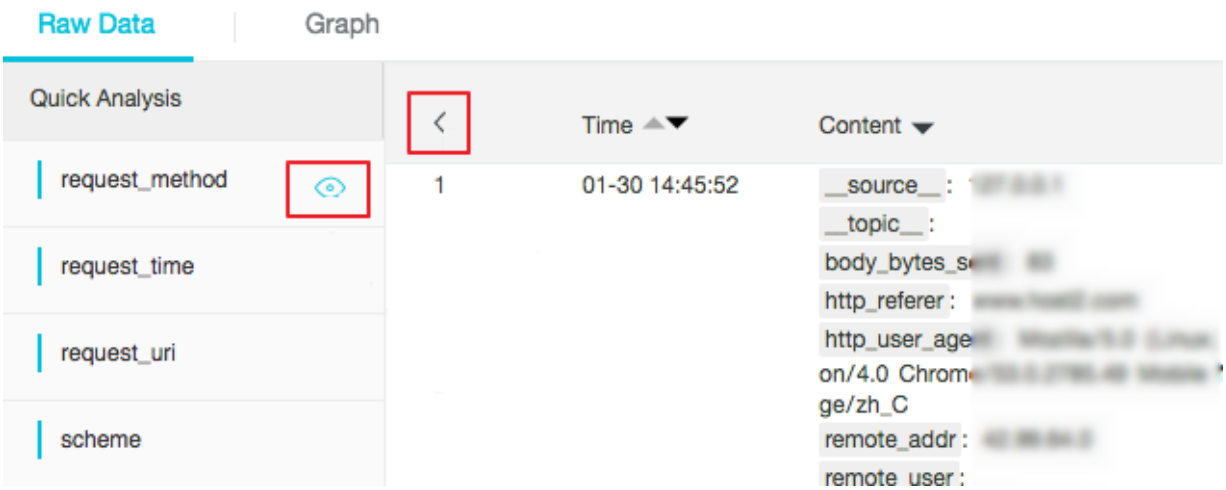


User Guide

After setting the specified field query, you can see the fields in Quick Analysis under the Raw Data tab on the query page. By clicking the 1 button above the serial number,

you can fold the page. By clicking the eye button, you can perform quick analysis based on the Current Temporal Interval and Current $Search conditions.

Figure 6-13: Original log



Text

- Grouping statistics for Text fields

  Click the eye button at the right of the filed to quickly group the first 100,000 pieces of data of this Text field and return the ratio of the first 10 pieces.

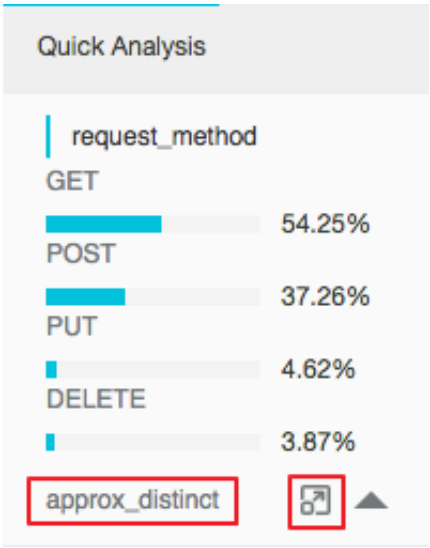
  Query statement:

```
$ Search | select ${ keyName } , pv , pv * 1 . 0 / sum ( pv )
 over () as percentage from ( select count ( 1 ) as pv ,
 "${ keyName }" from ( select "${ keyName }" from log limit
```

```
    100000 )  group   by  "${ keyName }"  order   by   pv   desc )
 order   by   pv   desc   limit   10
```

`request_me thod` returns the following result based on the grouping statistics, where GET requests are in the majority.

Figure 6-14: Group statistics



- Check the number of unique entries of the field

  Under the target fields in Quick Analysis, clickapprox_distinct to check the number of unique entries for `${ keyName }`.

  `request_me thod` can get the following result by grouping statistics, and `GET` requests account for the majority:
- Extend the query statement of grouping statistics to the search box

  Click the button at the right of approx_distinct to extend the query statement of grouping statistics to the search box for further operations.

long/double

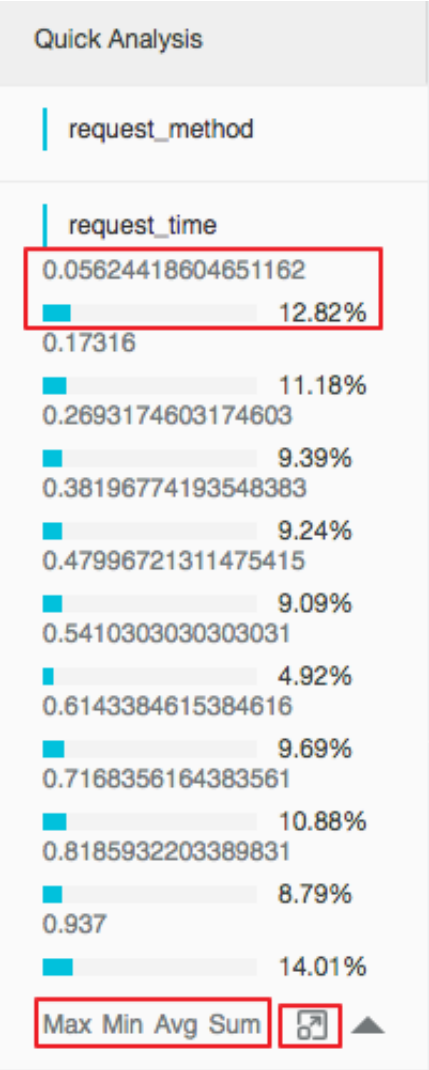
- Histogram statistics for the approximate distribution

  Grouping statistics is of little significance for the `long / double` fields, which have multiple type values. Therefore, histogram statistics for the approximate distribution is adopted by using 10 buckets.

```
$ Search | select numeric_histogram ( 10 , ${ keyName })
```

`request_time` returns the following result based on the histogram statistics for the approximate distribution, from which you can see that the request time is mostly distributed around 0.056.

Figure 6-15: Request Distribution



- Quick analysis of the `Max` `Min` `Avg` `Sum` statements

  Respectively click `Max` , `Min` , `Avg` , and `Sum` under the target fields to quickly search for the maximum, minimum, average, and sum of all ${keyName}.
- Extend the query statement of grouping statistics to the search box

  Click the button at the right of `Sum` to extend the query statement of the histogram statistics for the approximate distribution to the search box for further operations.

6.7.7 Other functions

Search and analysis functions help you to query various statements in logs, *query raw logs*, *view graphs*, *query context*, *perform quick analysis*, *perform quick queries*, *create a dashboard*, and *save a graph as an alarm*.

Query raw logs

After the index is enabled, enter the keywords in the search box and select the search time range. Then, click Search to view the histogram of the log quantity, the raw logs, and the statistical graph.

The histogram of the log quantity displays the time-based distribution of log search hit counts. With the histogram, you can view the log quantity changes over a certain period of time. By clicking the rectangular area to narrow down the time range, you can view the information about the log hits within the specified time range to refine the display of the log search results.

On the Raw Data tab, you can view the hit logs in chronological order.

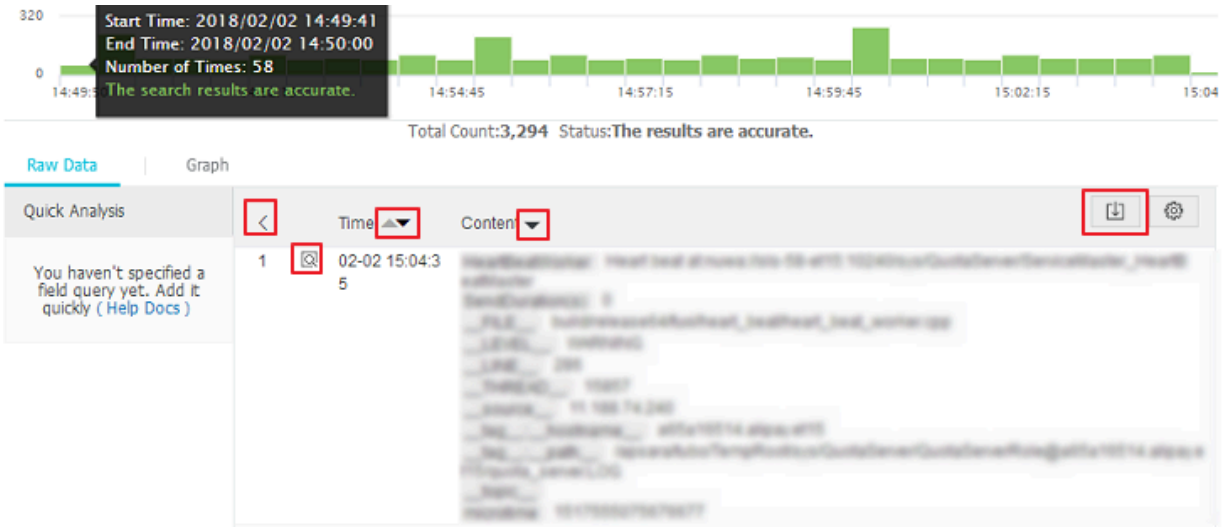
- By clicking the triangle symbol next to Time, you can switch between the chronological and reverse chronological orders.
- By clicking Display Content Column, you can switch between Display with Line Breaks and Display in One Line, or you can set Truncate Character String.
- By clicking the value keyword in the log content, you can view all logs containing this keyword.
- By clicking the Downloadbutton in the upper-right corner of the Raw Data tab, you can download the query results in CSV format. By clicking the Configbutton, you can add fields as displayed columns in the display results of raw logs so that

you can view the target field content of each raw log in the new columns in a more intuitive way.

- By clicking Context, you can view 15 logs before and after the current log entry. For more information, see *Context query*.

> **Note:**
> Currently, the context query function supports only the data uploaded with Logtail.

Figure 6-16: Raw logs



View graphs

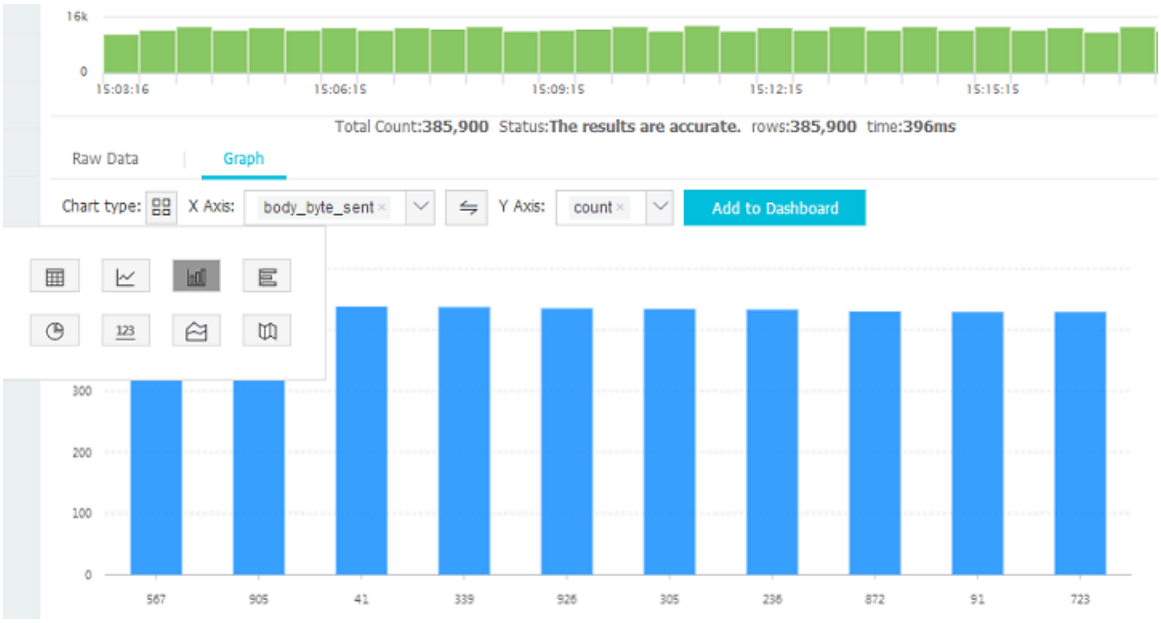
After enabling the index and entering a statement for query and analysis, you can view the statistics of logs under the Graph tab.

- Data can be displayed in tables, line charts, or other types of graphs.

  You can choose an appropriate statistical graph and custom graph settings as needed.
- You can add a graph to the Dashboard. For more information, see *Create and delete a dashboard*.

- You can set the *drill-down analysis* action for a graph. Then, after a graph is added to the dashboard, any click to a data point on the graph will trigger the drill-down analysis action, allowing you to review queries in more details.

Figure 6-17: Statistical graphs



Query context

The Log Service console provides a query page, you can view the context information of the specified log in the original file in the console. It is similar to paging up and down in the original log file. By viewing the context information of the specified log, you can quickly locate the failure information during the business troubleshooting. For more information, see *Context query*.

Perform quick analysis

The quick analysis function of Log Service supports an interactive query with only one click, allowing you to quickly analyze the distribution of a field over a period of time and reduce the cost of indexing key data. For more information, see *Quick analysis*.

Perform quick queries

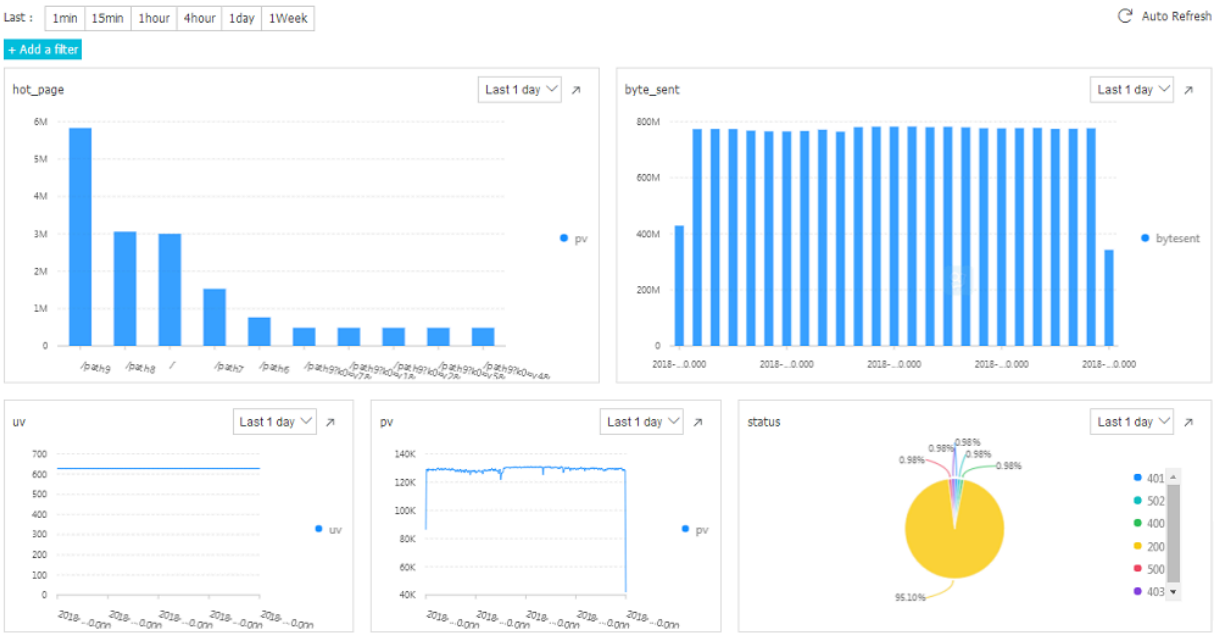
You can save the current query condition as a saved search. To perform this query again, you simply need to go to the saved search page. For more information, see *Saved search*.

You can also apply the saved search condition to alarm rules. After you set an alarm rule, Log Service will automatically run the saved search on a regular basis. If query results meet the preset threshold, Log Service will send an alarm message.

Create a dashboard

Log Service provides the dashboard function, which can visualize the query and analysis statements. For more information, see *Create and delete a dashboard*.

Figure 6-18: Dashboard



Save a graph as an alarm

Log Service can generate an alarm based on your LogSearch Results. You can configure the alarm rules so that specific alarm content can be sent to you by using in-site notifications or DingTalk messages.

For more information, see *Configure an alarm*.

6.8 Analysis grammar

6.8.1 General aggregate functions

The query and analysis function of Log Service supports analyzing logs by using general aggregate functions. The specific statements and meanings are as follows.

Statement	Meaning	Example
`arbitrary ( x )`	Returns a value in column x randomly.	`latency > 100 | select arbitrary ( method )`
`avg ( x )`	Calculates the arithmetic mean of all the values in column x.	`latency > 100 | select avg ( latency )`
`checksum ( x )`	Calculates the checksum of all the values in a column and returns the base64-encoded value.	`latency > 100 | select checksum ( method )`
`count (*)`	Calculates the number of rows in a column.	-
`count ( x )`	Calculates the number of non-null values in a column.	`latency > 100 | count ( method )`
`count_if ( X )`	Calculates the number of X = true.	`latency > 100 | count ( url like '% abc ')`
`geometric_ mean ( x )`	Calculates the geometric mean of all the values in a column.	`latency > 100 | select geometric_ mean ( latency )`
`max_by ( x , y )`	Returns the value of column x when column y has the maximum value.	The method for the maximum latency: `latency > 100 | select max_by ( method , latency )`
`max_by ( x , y , n )`	Returns the values of column x corresponding to the maximum n rows of column y.	The method for the top 3 rows with maximum latency: `latency > 100 | select max_by ( method , latency , 3 )`
`min_by ( x , y )`	Returns the value of column x when column y has the minimum value.	The method for the minimum latency: `* | select min_by ( x , y )`

Statement	Meaning	Example
`min_by ( X , Y , n )`	Returns the value of column x when column y has the minimum value.	The method for the minimum latency: `* | select min_by ( method , latency , 3 )`
`max ( x )`	Returns the maximum value.	`latency > 100 | select max ( inflow )`
`min ( x )`	Returns the minimum value	`latency > 100 | select min ( inflow )`
`sum ( x )`	Returns the sum of all the values in column x.	`latency > 10 | select sum ( inflow )`
`bitwise_an d_agg ( x )`	Do the AND calculation to all the values in a column.	-
`bitwise_or _agg ( x )`	Do the OR calculation to all the values in a column.	-

6.8.2 Security detection functions

Based on the global white hat shared security asset library, Log Service provides security detection functions. All you need to do is to pass any IP address, domain name, or URL in the log to security detection functions, you can detect whether it is secure or not.

Scenarios

1. Enterprises and institutions that have a strong demand for service operation and maintenance, such as enterprises of Internet, games, information, and more. The IT and security Operation and Maintenance (O&M) personnel of these industries can use security detection functions to timely filter for suspicious accesses, attacks , and intrusions. The security detection function also supports further in-depth analysis and measures to defend against them.
2. Enterprises and institutions that have strong demand for internal asset protection , such as banks, securities, e-commerce, and more. Their IT and security O&M personnel can instantly discover internal access to dangerous websites, download the trojan horse, and more, and take immediate action.

Features

- Reliable: Relies on the global shared white hat security asset library with timely update.
- Fast: Takes only a few seconds to detect millions of IP address, domain names, or URLs.
- Simple: Seamlessly supports any network log. The result can be obtained by calling three SQL functions: security_check_ip, security_check_domain, and security_check_url.
- Flexible: Supports both interactive queries and building report views. You can configure alarms and take further action.

Function list

Function name	Description	Example
security_check_ip	Check if the IP address is secure, where: · Return 1: Hit, indicating insecure · Return 0: Missing	`select security_check_ip ( real_client_ip )`
security_check_domain	Check if the domain is secure, where: · Return 1: Hit, indicating insecure · Return 0: Missing	`select security_check_domain ( site )`
security_check_url	Check if the URL is secure, where: · Return 1: Hit, indicating insecure · Return 0: Missing	`select security_check_domain ( concat ( host , url )`

Example

- Check external suspicious access behavior and generate reports

  An e-commerce collects logs of the Ngnix server that it operates, and intends to scan clients that access the server to check if insecure client IP addresses exist. In this case, pass the ClientIP field in the Ngnix log to the `security_check_ip`

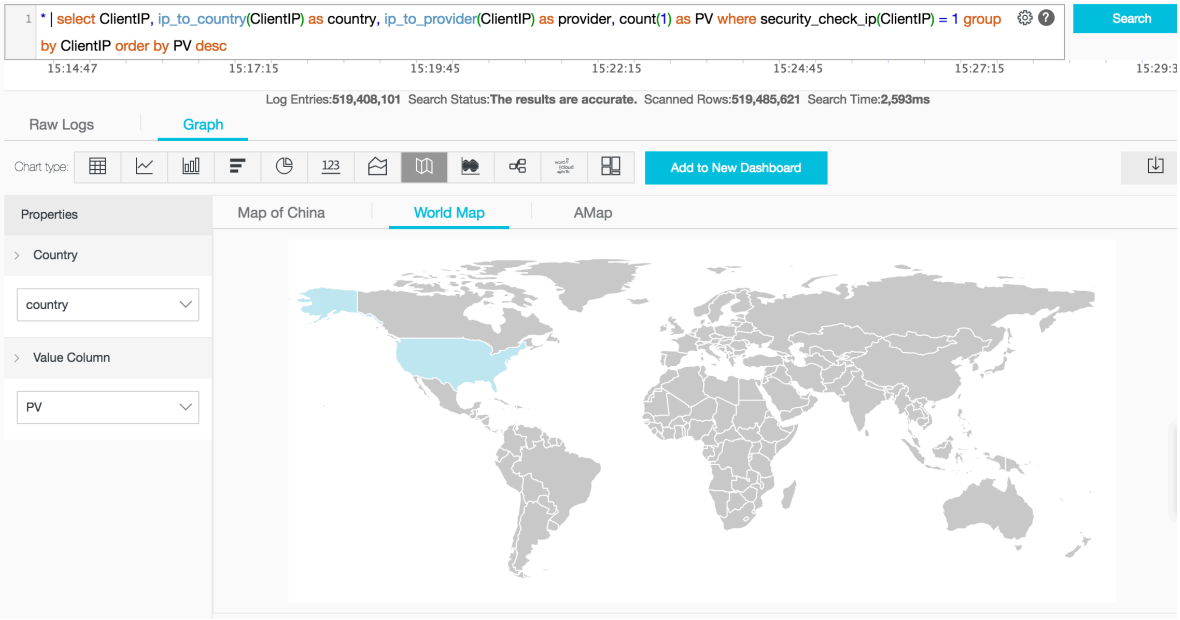
funciton, display IP addresses whose return value is 1, and show the country, the network operator and other related information of the IP addresses.

The query analysis statement is:

```
* | select ClientIP , ip_to_coun try ( ClientIP ) as country
, ip_to_prov ider ( ClientIP ) as provider , count ( 1 ) as
PV where security_c heck_ip ( ClientIP ) = 1 group by
ClientIP order by PV desc
```

ClientIP	sec	country	provider	PV
[redacted]	1	中国	电信	575
[redacted]	1	中国	联通	241
[redacted]	1	中国	电信	185
[redacted]	1	中国	联通	179
[redacted]	1	中国	联通	32
[redacted]	1	中国	电信	28

Set to map view display:



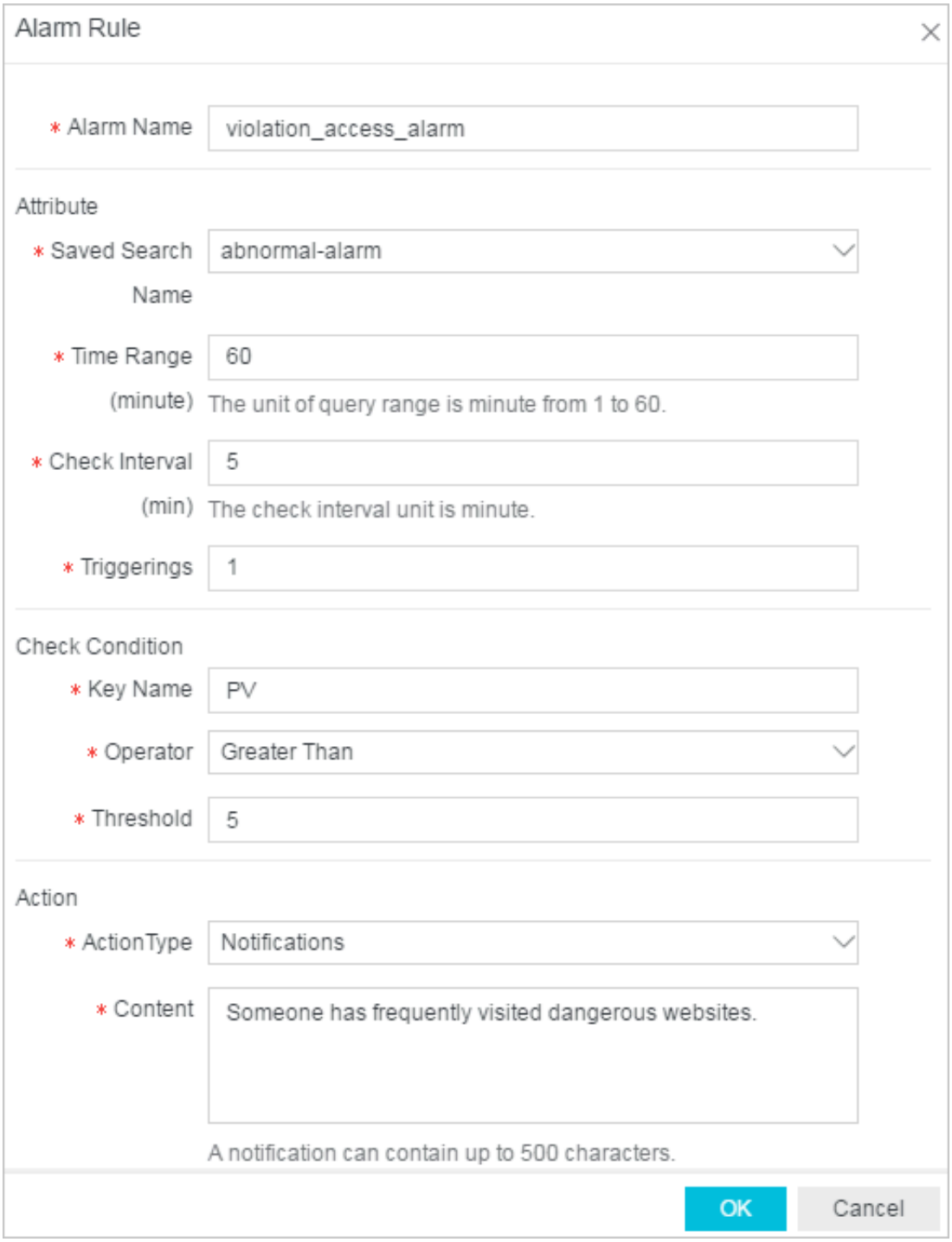
- Check internal suspicious access behavior and configure alarms

  For example, a securities operator collects network traffic logs recorded when its internal devices access the external network through a gateway proxy. To check if someone has accessed websites with problems, perform the following query:

```
* | select client_ip , count ( 1 ) as PV where
security_c heck_ip ( remote_add r ) = 1 or security_c
```

```
heck_site ( site ) =  1   or   security_c  heck_url ( concat ( site
,  url )) =  1   group   by   client_ip   order   by   PV   desc
```

You can also save this statement as a quick query and configure a security alarm. When a client access dangerous websites frequently, the alarm is triggered. Configure 5-minute intervals for checking if someone has accessed dangerous websites frequently (more than 5 times) during the past one hour. Change parameters based on your needs. The configuration is as follows:



6.8.3 Mapping function

The Log Service query analysis function supports log analysis by using mapping functions with detailed statements and implications described in the following table:

Function	Description	Example
Subscript operator []	Gets the result of a key in the map.	-
histogram(x)	Performs GROUP BY according to each value of column x and calculates the count. The syntax is equivalent to `select count group by x` .  Note: Returned information must be in JSON format.	`latency > 10 | select histogram ( status )`, which is equivalent to `latency > 10 | select count ( 1 ) group by status`
histogram_u(x)	Performs GROUP BY according to each value of column x and calculates the count.  Note: Returned information must be in multi-row multi-column format.	`latency > 10 | select histogram ( status )`, which is equivalent to `latency > 10 | select count ( 1 ) group by status`
map_agg(Key,Value)	Returns a map of key, value, and shows the random latency of each method.	`latency > 100 | select map_agg ( method , latency )`
multimap_agg(Key,Value)	Returns a multi-value map of key, value, and returns all the latency for each method.	`latency > 100 | select multimap_agg ( method , latency )`
cardinality(x) → bigint	Gets the size of the map.	-
element_at(map`< K , V >`, key) → V	Gets the value corresponding to the key.	-

Function	Description	Example
map() → map< unknown , unknown >	Returns an empty map.	-
map(array< K >, array< V >) → map< K , V >	Converts two arrays into 1-to-1 maps.	SELECT map (ARRAY [1 , 3], ARRAY [2 , 4]); — { 1 -> 2 , 3 -> 4 }
map_from_entries(array< row < K , V >>) → map< K , V >	Converts a multidimen sional array into a map.	SELECT map_from_e ntries (ARRAY [(1 , ‘ x ’), (2 , ‘ y ’)]); — { 1 -> ‘ x ’, 2 -> ‘ y ’}
map_entries(map< K , V >) → array< row < K , V >>	Converts an element in a map into an array.	SELECT map_entrie s (MAP (ARRAY [1 , 2], ARRAY [‘ x ’, ‘ y ’])); — [ROW (1 , ‘ x ’), ROW (2 , ‘ y ’)]
map_concat(map1< K , V >, map2< K , V >, ⋯, mapN< K , V >) → map< K , V >	The Union of multiple maps is required, if a key exists in multiple maps, take the first one.	-
map_filter(map< K , V >, function) → map< K , V >	Refer to the lambda map_filter function.	-
transform_keys(map< K1 , V >, function) → MAP< K2 , V >	Refer to the lambda transform_keys function.	-
transform_values(map< K , V1 >, function) → MAP< K , V2 >	Refer to the lambda transform_values function .	-
map_keys(x< K , V >) → array< K >	Gets all the keys in the map and returns an array.	-
map_values(x< K , V >) → array< V >	Gets all values in the map and returns an array.	-

Function	Description	Example
map_zip_with(map< K , V1 >, map< K , V2 >, function< K , V1 , V2 , V3 >) → map< K , V3 >	Refer to power functions in Lambda.	-

6.8.4 Estimating functions

The query and analysis function of Log Service supports analyzing logs by using estimating functions. The specific statements and meanings are as follows.

Function	Description	Examples
approx_distinct (x)	Estimates the number of unique values in column x.	-
approx_percentile (x , percentage)	Sorts the column x and returns the value approximately at the given percentage position.	Returns the value at the half position: approx_percentile (x , 0 . 5)
approx_percentile (x , percentages)	Similar to the preceding statement, but you can specify multiple percentages to return the values at each specified percentage position.	approx_percentile (x , array [0 . 1 , 0 . 2])

Function	Description	Examples
`numeric_hi stogram ( buckets , Value )`	Collects values in the numeric column by bucket. That is, you need to enter the *Value* column into buckets. The number of buckets is determined by *buckets*. The returned information is the key of each bucket and the corresponding count. This works similarly to `select count group by` for numbers. Note: Returned results must be in JSON format.	For POST requests, the delay is divided into 10 buckets. You can run `method : POST | select numeric_hi stogram ( 10 , latency )` to check the size of each bucket.

Function	Description	Examples
`numeric_hi stogram ( buckets , Value )`	Collects values in the numeric column by bucket. That is, you need to enter the *Value* column into buckets. The number of buckets is determined by *buckets*. The returned information is the key of each bucket and the corresponding count. This works similarly to `select count group by` for numbers. Note: Returned results must be in multi-row multi-column format.	For POST requests, the delay is divided into 10 buckets. You can run `method : POST | select numeric_hi stogram ( 10 , latency )` to check the size of each bucket.

6.8.5 Mathematical statistics functions

The query and analysis function of Log Service supports analyzing logs by using mathematical statistics functions. The specific statements and meanings are as follows.

Statements	Meaning	Example
`corr ( y , x )`	Returns the correlation coefficient of two columns. The result is from 0 to 1.	`latency > 100 | select corr ( latency , request_si ze )`
`covar_pop ( y , x )`	Calculates the population covariance.	`latency > 100 | select covar_pop ( request_si ze , latency )`

Statements	Meaning	Example
`covar_samp ( y , x )`	Calculates the sample covariance.	`Latency > 100 | select covar_samp ( request_si ze , latency )`
`regr_inter cept ( y , x )`	Returns the linear regression intercept of input values. y is the dependent value. x is the independent value.	`latency > 100 | select regr_inter cept ( request_si ze , latency )`
`regr_slope ( y , x )`	Returns the linear regression slope of input values. y is the dependent value. x is the independen t value.	`latency > 100 | select regr_slope ( request_si ze , latency )`
`stddev ( x )` or `stddev_sam p ( x )`	Returns the sample standard deviation of column x.	`latency > 100 | select stddev ( latency )`
`stddev_pop ( x )`	Returns the population standard deviation of column x.	`latency > 100 | select stddev_pop ( latency )`
`variance ( x )` or `Var_samp ( X )`	Calculates the sample variance of column x.	`latency > 100 | select variance ( latency )`
`var_pop ( x )`	Calculates the population variance of column x.	`latency > 100 | select variance ( latency )`

6.8.6 Mathematical calculation functions

The query and analysis function of Log Service supports analyzing logs by using mathematical calculation functions. By combining query statements with mathematical calculation functions, you can perform mathematical calculation to the log query results.

Mathematical operators

Mathematical operators support plus sign (+), minus sign (-), multiplication sign (*), division sign (/), and percent sign (%), which can be used in the SELECT clause.

Example:

```
*| select   avg ( latency )/ 100  ,  sum ( latency )/ count ( 1 )
```

Description of mathematical calculation function

Log Service supports the following operating functions.

Function name	Meaning
`abs ( x )`	Returns the absolute value of column x.
`Cbrt  ( X )`	Returns the cube root of column x.
`ceiling  ( x )`	Returns the number rounded up to the nearest integer of column x.
`cosine_sim  ilarity ( x , y )`	Returns the cosine similarity between the sparse vectors x and y.
`degrees`	Converts radians to degrees.
`e ()`	Returns the natural constant.
`exp ( x )`	Returns the exponent of the natural constant.
`floor ( x )`	Returns the number rounded down to the nearest integer of column x.
`from_base ( string , radix )`	Returns the string interpreted in the base -radix notation.
`ln ( x )`	Returns the natural logarithm.Returns the natural log.
`log2 ( x )`	Returns the base-2 logarithm of x.
`log10 ( x )`	Returns the base-10 logarithm of x.
`log ( x , b )`	Returns the base-b logarithm of x.
`pi ()`	Returns π.
`pow ( x , b )`	Returns x to the power of b.
`radians ( x )`	Converts degrees to radians.
`rand ()`	Returns a random number.
`random ( 0 , n )`	Returns a random number in the range of [0,n).
`round ( x )`	Returns x rounded to the nearest integer.

Function name	Meaning
`round ( x , y )`	Returns x rounded to the nearest integer.
`sqrt ( x )`	Returns the square root of x.
`to_base ( x , radix )`	Returns the base-radix representation of x.
`truncate ( x )`	Returns x rounded to integer by dropping digits after decimal point.
`acos ( x )`	Returns the arc cosine.
`Asin ( X )`	Returns the arc sine.
`atan ( x )`	Returns the arc tangent.
`atan2 ( y , x )`	Returns the arc tangent of y/x.
`cos ( x )`	Returns the cosine.
`sin ( x )`	Returns the sine.
`cosh ( x )`	Returns the hyperbolic cosine.
`tan ( x )`	Returns the tangent.
`tanh ( x )`	Returns the hyperbolic tangent.
`Infinity ()`	Returns the double maximum value.
`is_infinit y ( x )`	Determines whether it is the maximum value or not.
`is_finity ( x )`	Determines whether it is the maximum value or not.
`is_nan ( x )`	Determines whether it is a number or not .

6.8.7 String functions

The query and analysis function of Log Service supports analyzing logs by using string functions. The specific statements and description are as follows.

Function name	Description
`chr ( x )`	Converts the int type to the corresponding ASCII string, for example `chr ( 65 )`=’A’.

Function name	Description
`codepoint ( x )`	Converts the ASCII type to the corresponding int string, for example `codepoint (' a ')`=97.
`length ( x )`	Returns the length of a field.
`levenshtei n_distance ( string1 , string2 )`	Returns the minimum edit distance between two strings.
`lower ( string )`	Converts the string to lowercase characters.
`lpad ( string , size , padstring )`	Aligns the string to the size. If it is smaller than the size, uses padstring to fill the size from the left side. If it is larger than size, it is truncated to size.
`rpad ( string , size , padstring )`	The same as the lpad, complement the string from the right.
`ltrim ( string )`	Deletes the white-space characters on the left.
`replace ( string , search )`	Deletes search from the string.
`replace ( string , search , rep )`	Replaces search with rep in the string.
`reverse ( string )`	Returns a string with the characters in the reverse order.
`rtrim ( string )`	Deletes the white-space characters at the end of the string.
`split ( string , delimeter , limit )`	Split the string into array and get a maximum of limit values. The generated result is an array with subscripts starting at 1.
`split_part ( string , delimeter , offset )`	Splits the string into an array and obtains the offset string. The generated result is an array with subscripts starting at 1.
`split_to_m ap ( string , entryDelim iter , keyValueDe limiter ) → map < varchar , varchar >`	The string is divided into multiple entries according to entryDelemiter, and each entry is divided into key values according to keyValueDelimiter. Eventually returns a map.
`position ( substring IN string )`	Get the position in the string where the substring starts.

Function name	Description
`strpos ( string , substring )`	Finds the starting position of the substring in the string. The returned result starts at 1. If not found, 0 is returned.
`substr ( string , start )`	Returns a substring of a string with a subscript starting at 1.
`substr ( string , start , length )`	Returns a substring of a string with a subscript starting at 1 and length.
`trim ( string )`	Deletes the white-space characters at the beginning and end of the string.
`upper ( string )`	Converts the string to uppercase characters.
`concat ( string , string ......)`	Splices two or more strings into a single string.
`hamming_distance ( string1 , string2 )`	Returns the hamming distance between two strings.

 Note:

Strings must be enclosed in single quotation marks, and double quotation marks indicate column names. For example, a=’ abc’ indicates column a = string abc, and a = “abc” means column a = column abc.

6.8.8 Date and time functions

Log Service supports time functions, date functions, and interval functions. You can use the date, time, and interval functions introduced in this document in the analysis syntax.

Date and time type

1. unixtime: Indicates the number of seconds since January 1, 1970 in the type of int. For example, `1512374067` indicates the time `Mon Dec 4 15 : 54 : 27 CST 2017` . The built-in time `__time__` in each log of Log Service is of this type.
2. timestamp type: Indicates the time in the format of string. For example, `2017 - 11 - 01 13 : 30 : 00` .

Date functions

The common date functions supported by Log Service are as follows.

Function	Meaning	Example
`current_da te`	Returns the current date.	`latency > 100 | select current_da te`
`current_ti me`	Returns the current time.	`latency > 100 | select current_ti me`
`current_ti mestamp`	Returns the result combined by current_date and current_time.	`latency > 100 | select current_ti mestamp`
`current_ti mezone ()`	Returns the time zone.	`latency > 100 | select current_ti mezone ()`
`from_iso86 01_timesta mp ( string )`	Converts an iso8601 time to a date with time zone.	`latency > 100 | select from_iso86 01_timesta mp ( iso8601 )`
`from_iso86 01_date ( string )`	Converts an iso8601 time to a date.	`latency > 100 | select from_iso86 01_date ( iso8601 )`
`from_unixt ime ( unixtime )`	Converts a UNIX time to a timestamp.	`latency > 100 | select from_unixt ime ( 1494985275 )`
`from_unixt ime ( unixtime , string )`	Converts a UNIX time to a timestamp by using the string as the time zone.	`latency > 100 | select from_unixt ime ( 1494985275 ,' Asia / Shanghai ')`
`localtime`	Returns the current time.	`latency > 100 | select localtime`
`localtimes tamp`	Returns the current timestamp.	`latency > 100 | select localtimes tamp`

Function	Meaning	Example
now ()	Equivalent to current_timestamp .	-
to_unixtime (timestamp)	Timestamp is converted into unixtime.	*| select to_unixtime ('2017-05-17 09:45:00.848 Asia/Shanghai')

Time Function

MySQL time format

Log Service supports the MySQL time format such as %a, %b, and %y.

Function	Meaning	Example
date_format (timestamp , format)	Converts the timestamp into a format representation.	latency > 100 | select date_format (date_parse ('2017-05-17 09:45:00','%Y-%m-%d %H:%i:%S'), '%Y-%m-%d') group by method
date_parse (string , format)	Parses a string into a timestamp by using the format.	latency > 100 | select date_parse ('2017-05-17 09:45:00','%Y-%m-%d %H:%i:%S') group by method

Table 6-2: Description

Format	Description
%a	The abbreviation of a day in a week, such as Sun and Sat.
%b	The abbreviation of a month, such as Jan and Dec.
%c	Month, in the numeric type: 1 to 12.
%D	The day of each month with a suffix, such as 0th, 1st, 2nd, and 3nd.

Format	Description
%d	The day of each month, which is in decimal format and in the range of 01 to 31.
%e	The day of each month, which is in decimal format and in the range of 1 to 31.
%H	The hour in 24-hour format.
%h	The hour in 12-hour format.
%I	The hour in 12-hour format.
%i	Minutes, which is in the type of number and in the range of 00 to 59.
%j	The day of each year, which is in the range of 001 to 366.
%k	Hour, which is in the range of 0 to 23.
%l	Hour, which is in the range of 1 to 12.
%M	The English expression of a month, which is in the range of January to December.
%m	Month, which is in the numeric format and in the range 01 to 12.
%p	AM or PM.
%r	Time in 12-hour format: `hh : mm : ss AM / PM`.
% S	Seconds, in the range of 00 to 59.
%s	Seconds, in the range of 00 to 59.
%T	Time, in the 24-hour format: `hh : mm : ss`.
%U	The week number of each year. Sunday is the first day of each week. The value range is from 00 to 53.
%u	The week number of each year. Monday is the first day of each week. The value range is 00 to 53.
%V	The week number of each year. Sunday is the first day of each week. The value range is 01 to 53. Use this format in conjunction with %X.
%v	The week number of each year. Monday is the first day of each week. The value range is 01 to 53. Use this format in conjunction with %x.
%W	The name of each day of a week, in the range of Sunday to Saturday.

Format	Description
%w	The day of the week, in the range of 0 to 6. Sunday is the day 0.
%Y	The year in the 4-digit format.
%y	The year in the 2-digit format.
%%	%Escape character

Time period alignment functions

Log Service supports time period alignment functions, which can be aligned according to seconds, minutes, hours, days, months, and years. Time period alignment functions are usually used when statistics are made according to time.

Function syntax:

```
date_trunc ( unit ,  x )
```

Parameters:

The optional values for Unit are as follows (x is `2001 - 08 - 22  03 : 04 : 05 . 000`):

Unit	Converted result
second	2001-08-22 03:04:05.000
minute	2001-08-22 03:04:00.000
hour	2001-08-22 03:00:00.000
day	2001-08-22 00:00:00.000
week	2001-08-20 00:00:00.000
month	2001-08-01 00:00:00.000
quarter	2001-07-01 00:00:00.000
year	2001-01-01 00:00:00.000

x can be of the timestamp type or UNIX time type.

date_trunc can only make statistics every fixed time period. If you need to make statistics according to flexible time dimension, for example, make the statistics every five minutes, perform GROUP BY according to the mathematical modulus method.

```
* | SELECT count ( 1 ) as pv , __time__ - __time__ % 300
 as minute5gro upby minute5 limit 100
```

The `% 300` indicates to make the modulus and alignment every five minutes.

Date function example

The following is a comprehensive example using the time format:

```
*| select date_trunc (' minute ' , __time__ ) as t ,
        truncate ( avg ( latency ) ) ,
        current_da te
        group by t
        order by t desc
        limit 60
```

Interval functions

Interval functions are used to perform interval related calculation. For example, add or delete an interval in the date, or calculate the time between two dates.

Function	Description	Example
`date_add (`*unit*`, `*value*`, `*timestamp*`)`	Add `value unit` to `timestamp`. To perform minus calculation, use a negative `value`.	`date_add (' day ', - 7 , ' 2018 - 08 - 09 00 : 00 : 00 ')` indicates seven days before August 9.
`date_diff(`*unit*`, `*timestamp1*`, `*timestamp2*`)`	The number of `unit` between `timestamp1` and `timestamp2`.	`date_diff (' day ', ' 2018 - 08 - 02 00 : 00 : 00 ', ' 2018 - 08 - 09 00 : 00 : 00 ') = 7`

The function supports the following interval units:

Unit	Description
millisecond	Milliseconds
second	Seconds
minute	Minutes
hour	Hours

Unit	Description
day	Days
week	Weeks
month	Months
quarter	A quarter, namely, three months.
year	Years

6.8.9 URL functions

URL functions support extracting fields from standard URL paths. A standard URL is as follows:

```
[ protocol :][// host [: port ]][ path ][?  query ][# fragment ]
```

Common URL functions

Function Name	Meaning	Example
`url_extrac t_fragment ( url )`	Extracts the fragment from a URL and the result is of varchar type.	`*| select url_extrac t_fragment ( url )`
`url_extrac t_host ( url )`	Extracts the host from a URL and the result is of varchar type.	`*| select url_extrac t_host ( url )`
`url_extrac t_paramete r ( url , name )`	Extracts the value of the name parameter in the query from a URL and the result is of varchar type.	`*| select url_extrac t_paramete r ( url )`
`url_extrac t_path ( url )`	Extracts the path from a URL and the result is of varchar type.	`*| select url_extrac t_path ( url )`
`url_extrac t_port ( url )`	Extracts the port from a URL and the result is of bigint type.	`*| select url_extrac t_port ( url )`
`url_extrac t_protocol ( url )`	Extracts the protocol from a URL and the result is of varchar type.	`*| select url_extrac t_protocol ( url )`
`url_extrac t_query ( url )`	Extracts the query from a URL and the result is of varchar type.	`*| select url_extrac t_query ( url )`

Function Name	Meaning	Example
`url_encode ( value )`	Encodes a URL.	`*| select url_encode ( url )`
`url_decode ( value )`	Decodes a URL.	`*| select url_decode ( url )`

6.8.10 Regular expression functions

A regular expression function parses a string and returns the needed substrings.

The common regular expression functions and the meanings are as follows:

Function name	Meaning	Example
`regexp_extract_all ( string , pattern )`	Returns all the substrings that match the regular expression in the string as a string array.	`*| SELECT regexp_extract_all (' 5a 67b 890m ', '\ d +')`, results in `[' 5 ',' 67 ',' 890 ']`, `*| SELECT regexp_extract_all (' 5a 67a 890m ', '(\ d +) a ')` returns `[' 5a ',' 67a ']`.
`regexp_extract_all ( string , pattern , group )`	Returns the part of the string that hits the regular () part of the group, returns the result as an array of strings.	`*| ` SELECT regexp_extract_all (‘ 5a 67a 890m ’, ‘(\ d +) a ’, 1)` returns `[‘ 5 ’,’ 67 ’]`
`regexp_extract ( string , pattern )`	Returns the first substring that matches the regular expression in the string.	`*| SELECT regexp_extract (' 5a 67b 890m ', '\ d +')` returns `' 5 '`
`regexp_extract ( string , pattern , group )`	Returns the first substring within the regular group () that hit the string.	`*| SELECT regexp_extract (' 5a 67b 890m ', '(\ d +)([ a - z ]+)', 2 )` returns `' b '`

Function name	Meaning	Example
`regexp_lik e ( string , pattern )`	Determines if the string matches the regular expression and returns a bool result. The regular expression is allowed to match part of the string.	`*| SELECT regexp_lik e (' 5a 67b 890m ', '\ d + m ')` returns true
`regexp_rep lace ( string , pattern , replacemen t )`	Replaces the part that matches the regular expression in the string with replacement.	`*| SELECT regexp_rep lace (' 5a 67b 890m ', '\ d +',' a ')` returns `' aa ab am '`
`regexp_rep lace ( string , pattern )`	Removes the part that matches the regular expression in the string, which is equivalent to `regexp_rep lace ( string , patterm ,'')`.	`*| SELECT regexp_rep lace (' 5a 67b 890m ', '\ d +')` returns `' a b m '`
`regexp_spl it ( string , pattern )`	Splits the string to an array by using the regular expression.	`*| SELECT regexp_spl it (' 5a 67b 890m ', '\ d +')` returns `[' a ',' b ',' m ']`

6.8.11 JSON functions

JSON functions can parse a string as the JSON type and extract the fields in JSON. JSON mainly has the following two structures: map and array. If a string fails to be parsed as the JSON type, the returned value is null.

To split JSON into multiple lines, see *UNNEST function*.

Log Service supports the following common JSON functions.

Function	Description	Example
`json_parse ( string )`	Converts a string to the JSON type.	`SELECT json_parse ('[ 1 , 2 , 3 ]')` returns a JSON array

Function	Description	Example
`json_forma t ( json )`	Converts the JSON type to a string.	`SELECT json_forma t ( json_parse ('[ 1 , 2 , 3 ]'))` returns a string
`json_array _contains ( json , value )`	Determines whether a JSON type value or string (whose content is a JSON array) contains a value or not.	`SELECT json_array _contains ( json_parse ('[ 1 , 2 , 3 ]'), 2 ) or SELECT json_array _contains ('[ 1 , 2 , 3 ]', 2 )`
`json_array _get ( json_array , index )`	The same as `json_array _contains` , which is used to obtain the element of a subscript of a JSON array.	`SELECT json_array _get ('[" a ", " b ", " c "]', 0 )` returns `' a '`
`json_array _length ( json )`	Returns the size of the JSON array.	`SELECT json_array _length ('[ 1 , 2 , 3 ]') Returns 3`
`json_extra ct ( json , json_path )`	Extracts the value from a JSON object. The JSON path syntax is similar to `$. store . book [ 0 ]. title` . The returned result is a JSON object.	`SELECT json_extra ct ( json , '$. store . book ');`
`json_extra ct_scalar ( json , json_path )`	Similar to `json_extra ct` , but returns a string.	-
`json_size ( json , json_path )`	Obtains the size of the JSON object or array.	`Select json_size ('[ 1 , 2 , 3 ]') returns 3`

6.8.12 Type conversion functions

Log Service supports the long, double, and text types in the configurations and the bigint, double, varchar, timestamp, and int types in the query.

The type conversion functions forcibly convert a column to a specified type:

```
cast ( value AS type ) → type
```

```
try_cast ( value AS type ) → type
```

6.8.13 IP functions

IP recognition function can recognize whether the IP is an intranet IP or an Internet IP, and can determine the country, province, and city to which the IP belongs.

Function name	Meaning	Example
`ip_to_domain ( ip )`	Determines the domain in which the IP resides and whether the IP is an intranet IP or an Internet IP. The returned value is intranet or Internet.	`SELECT ip_to_domain ( ip )`
`ip_to_country ( ip )`	Determines the country in which the IP resides.	`SELECT ip_to_country ( ip )`
`ip_to_province ( ip )`	Determines the province in which the IP resides.	`SELECT ip_to_province ( ip )`
`ip_to_city ( ip )`	Determines the city in which the IP resides.	`SELECT ip_to_city ( ip )`
`ip_to_geo ( ip )`	Determines the longitude and latitude of the city where IP is located, the result of the range is in the form of `latitude and longitude`.	`SELECT ip_to_geo ( ip )`
`ip_to_city_geo ( ip )`	Determines the longitude and latitude of the city where IP is located. Returns the latitude and longitude of the city, each city has only one latitude and longitude. The result of the range is in the form of `latitude and longitude`.	`SELECT ip_to_city_geo ( ip )`
`ip_to_provider ( ip )`	Obtains the network operator of the IP.	`SELECT ip_to_provider ( ip )`

Function name	Meaning	Example
`ip_to_coun try ( ip ,' en ')`	Determines the country where the IP is located and return the international code.	`SELECT ip_to_coun try ( ip ,' en ')`
`ip_to_coun try_code ( ip )`	Determine the country where the IP is located and return the international code.	`SELECT ip_to_coun try_code ( ip )`
`ip_to_prov ince ( ip ,' en ')`	Determines the province where the IP is located, and returns the English province name or the Chinese alphabet.	`SELECT ip_to_prov ince ( ip ,' en ')`
`ip_to_city ( ip ,' en ')`	Determines the city where IP is located, and returns the English city name or the Chinese alphabet.	`SELECT ip_to_city ( ip ,' en ')`

Example

- Filter out the intranet access requests in the query and view the total number of requests

```
* | selectcoun t ( 1 ) whereip_to _domain ( ip )! =' intranet '
```

- View the top 10 access provinces

```
* | SELECT count ( 1 ) as pv , ip_to_prov ince ( ip ) as
 province GROUP BY province order by pv desc limit
 10
```

Response result example:

```
[
    {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " Zhejiang province ",
        " pv ": " 4045 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " Shanghai city ",
        " pv ": " 3727 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " Beijing city ",
```

```
        " pv ": " 954 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " intranet   IP ",
        " pv ": " 698 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " Guangdong   Province  ",
        " pv ": " 472 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ":  Fujian   Province  ",
        " pv ": " 71 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " United   ArabEmirat  es  ( UAE )",
        " pv ": " 52 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " United   States  ",
        " pv ": " 43 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " Germany  ",
        " pv ": " 26 "
    }, {
        " __source__ ": "",
        " __time__ ": " 1512353137 ",
        " province ": " Kuala   Lumpur  ",
        " pv ": " 26 "
    }
]
```


The preceding results include the intranet IP. Sometimes developers make tests from the intranet. To filter out these access requests, use the following analysis syntax.

- Filter out the intranet requests and view the top 10 network access provinces

```
* | SELECT count ( 1 ) as pv , ip_to_prov ince ( ip ) as
province WHERE ip_to_doma in ( ip ) ! = ' intranet ' GROUP
BY province ORDER BY pv desc limit 10
```

- Check the average response latency, the maximum response latency, and the request of the maximum latency in different countries

```
* | SELECT AVG ( latency ), MAX ( latency ), MAX_BY ( requestId
, latency ), ip_to_coun try ( ip ) as country group by
country limit 100
```

- View the average latency for different network operators

```
* | SELECT AVG ( latency ), ip_to_prov ider ( ip ) as
provider group by provider limit 100
```

- View the latitude and longitude of the IP, and build a map

```
* | select count ( 1 ) as pv , ip_to_geo ( ip ) as geo
group by geo order by pv desc
```

The returned format is:

pv	geo
100	35.3284,-80.7459

6.8.14 GROUP BY syntax

GROUP BY supports multiple columns and indicating the corresponding KEY by using the SELECT column alias.

Example:

```
method : PostLogsto reLogs | select avg ( latency ), projectNam
e , date_trunc (' hour ', __time__ ) as hour group by
projectNam e , hour
```

The alias hour represents the third SELECT column `date_trunc (' hour ', __time__ )`('hour',__time__). This kind of usage is very helpful for some very complicated queries.

GROUP BY supports GROUPING SETS, CUBE, and ROLLUP.

Example:

```
method : PostLogsto reLogs | select avg ( latency ) group by
cube ( projectNam e , logstore )
```

```
method : PostLogsto reLogs | select avg ( latency ) group by
  GROUPING SETS ( ( projectNam e , logstore ), ( projectNam e ,
method ))
method : PostLogsto reLogs | select avg ( latency ) group by
rollup ( projectNam e , logstore )
```

Practical example

Perform GROUP BY according to time

Each log has a built-in time column `__time__`. When the statistical function of any column is activated, the statistics will be automatically made for the time column.

Use the `date_trunc` function to align the time column to hour, minute, day, month, and year. `date_trunc` accepts an aligned unit and a UNIX time or timestamp type column, such as `__time__`.

- Count and compute PV every hour or minute

```
* | SELECT count ( 1 ) as pv , date_trunc (' hour ',
 __time__ ) as hour group by hour order by hour
 limit 100
* | SELECT count ( 1 ) as pv , date_trunc (' minute ',
 __time__ ) as minute group by minute order by minute
  limit 100
```

> **Note:**
> limit limit 100 indicates to obtain 100 rows at most. If the LIMIT statement is not added, at most 10 rows of data can be obtained by default.

- Make statistics according to flexible time dimension. For example, make the statistics every five minutes. date_trunc can only make statistics every fixed time period. In this situation, perform GROUP BY according to the mathematical modulus method.

```
* | SELECT count ( 1 ) as pv , __time__ - __time__ % 300
 as minute5 group by minute5 limit 100
```

The %300 indicates to make the modulus and alignment every five minutes.

Extract non-agg column in GROUP BY

In the standard SQL, if you use the GROUP BY syntax, you can only select the original contents of the SELECT GROUP BY columns when you perform SELECT or you are not allowed to obtain the contents of non-GROUP BY columns when you perform aggregation calculation on any column.

For example, the following syntax is illegal. This is because b is the non-GROUP BY column and multiple rows of b are available when you perform GROUP BY according to a, the system does not know which row of output is to be selected.

```
*| select  a ,  b  ,  count ( c )  group  by  a
```

To achieve the preceding aim, use the arbitrary function to output b:

```
*| select  a ,  arbitrary ( b ),  count ( c )  group  by  a
```

6.8.15 Window functions

Window functions are used for cross-row calculation. Common SQL aggregate functions calculate the results of only one row or aggregate all rows into one row for calculation. Window functions support cross-row calculation and enter the calculation results in each row.

Syntax of window functions:

```
SELECT  key1 ,  key2 ,  value ,
       rank ()  OVER  ( PARTITION  BY  key2
                  ORDER  BY  value  DESC )  AS  rnk
FROM  orders
ORDER  BY  key1 , rnk
```

Core part is:

```
rank ()  OVER  ( PARTITION  BY  KEY1  ORDER  BY  KEY2  DESC )
```

rank() is an aggregate function. You can use any function in analysis syntax or the function listed in this document. PARTITION BY indicates the buckets based on which values are calculated.

Special aggregate functions used in windows

Function name	Meaning
rank()	Sorts data based on a specific column in a window and returns the serial numbers in the window.
row_number()	Returns the row numbers in the window.
first_value(x)	Returns the first value in the window. Generally used to obtain the maximum value after values are sorted in the window.

Function name	Meaning
last_value(x)	Opposite to first_value.
nth_value(x, offset)	Value of the No. offset row in xth column in the window.
lead(x,offset,defaut_value)	Value of the No. offset row after a certain row in xth column in the window. If that row does not exist, use the default_value.
lag(x,offset,defaut_value)	Value of the No. offset row before a certain row in xth column in the window . If that row does not exist, use the default_value.

Example

- Rank the salaries of employees in their respective departments

```
* | select department , persionId , sallary , rank () over
 ( PARTITION BY department order by sallary desc ) as
 sallary_ra nk order by department , sallary_ra nk
```

Response results:

department	persionId	sallary	sallary_rank
dev	john	9000	1
dev	Smith	8000	2
dev	Snow	7000	3
dev	Achilles	6000	4
Marketing	Blan Stark	9000	1
Marketing	Rob Stark	8000	2
Marketing	Sansa Stark	7000	3

- Calculate the salaries of employees as percentages in their respective departments

```
* | select department , persionId , sallary * 1 . 0 / sum (
 sallary ) over ( PARTITION BY department ) as sallary_pe
 rcentage
```

Response results:

department	persionId	sallary	sallary_percentage
dev	john	9000	0.3

department	persionId	sallary	sallary_percentage
dev	Smith	8000	0.26
dev	Snow	7000	0.23
dev	Achilles	6000	0.2
Marketing	Blan Stark	9000	0.375
Marketing	Rob Stark	8000	0.333
Marketing	Sansa Stark	7000	0.29

- Calculate the daily UV increase over the previous day

```
* | select day , uv , uv * 1 . 0 /( lag ( uv , 1 , 0 ) over
 () ) as diff_perce ntage from

 select approx_dis tinct ( ip ) as uv , date_trunc (' day ',
 __time__ ) as day from log group by day order by
 day asc
```

Response results:

day	uv	diff_percentage
2017-12-01 00:00:00	100	null
2017-12-02 00:00:00	125	1.25
2017-12-03 00:00:00	150	1.2
2017-12-04 00:00:00	175	1.16
2017-12-05 00:00:00	200	1.14
2017-12-06 00:00:00	225	1.125
2017-12-07 00:00:00	250	1.11

6.8.16 HAVING syntax

The query and analysis function of Log Service supports the Having syntax of standard SQL, which is used together with the GROUP BY syntax to filter the GROUP BY results.

Format:

```
method : PostLogsto reLogs | select avg ( latency ), projectNam
e group by projectNam e HAVING avg ( latency ) > 100
```

Difference between HAVING and WHERE

HAVING is used to filter the aggregation and calculation results after performing GROUP BY. WHERE is used to filter the original data during the aggregation calculation.

Example

Calculate the average rainfall of each province whose temperature is greater than 10 ℃ and only display the provinces whose average rainfall is greater than 100 mL in the final result:

```
* | select avg ( rain ) , province where temperatur e > 10
group by province having avg ( rain ) > 100
```

6.8.17 ORDER BY syntax

ORDER BY is used to sort the output results. Currently, you can only sort the results by one column.

Syntax format:

```
order by Column name [ desc | asc ]
```

Example:

```
method : PostLogsto reLogs | select avg ( latency ) as
avg_latenc y , projectNam e group by projectNam e
HAVING avg ( latency ) > 5700000
order by avg_latenc y desc
```

6.8.18 LIMIT syntax

Limit syntax is used to limit the number of rows in the output results.

Syntax format:

Log Service supports the following two LIMIT syntax formats.

- Read only the first N rows:

```
limit N
```

- Read N rows from the S rows:

```
limit S , N
```



Note:

- When you use LIMIT syntax to read results across pages, it is used to get only the final result and cannot be used to get results in the middle of SQL.
- LIMIT syntax cannot be used in subquery. For example:

```
* | select count ( 1 ) from ( select distinct ( url ) from
   limit 0 , 1000 )
```

Examples

- Get only 100 rows of results:

```
* | select distinct ( url ) from log limit 100
```

- Get results from 0 rows to 999th rows, 1000 rows in total:

```
* | select distinct ( url ) from log limit 0 , 1000
```

- Get results from 1000th rows to 1999th rows, 1000 rows in total:

```
* | select distinct ( url ) from log limit 1000 , 1000
```

6.8.19 Case when and if branch syntax

Log Service supports CASE WHEN syntax to classify the continuous data. For example , extract the information from http_user_agent and classify the information into two types: Android and iOS.

```
SELECT
 CASE
 WHEN http_user_ agent like '% android %' then ' android '
 WHEN http_user_ agent like '% ios %' then ' ios '
 ELSE ' unknown ' END
 as http_user_ agent ,
   count ( 1 ) as pv
```

```
    group   by   http_user_  agent
```

Example

- The ratio of requests with 200 as the computing status code to the total number of requests:

```
* |  SELECT
  sum (
  CASE
  WHEN   status  = 200   then   1
  ELSE   0   end
 ) * 1 . 0  /  count ( 1 )  as   status_200  _percentag  e
```

- Make statistics of the distribution of different latency intervals

```
* |  SELECT  `
  CASE
  WHEN   latency  <  10   then  ' s10 '
  WHEN   latency  <  100   then  ' s100 '
  WHEN   latency  <  1000   then  ' s1000 '
  WHEN   latency  <  10000   then  ' s10000 '
  else  ' s_large '  end
  as   latency_sl  ot ,
  count ( 1 )  as   pv
  group   by   latency_sl  ot
```

IF syntax

The if syntax is logically equivalent to the CASE WHEN syntax.

```
Case
     WHEN   condition   THEN   true_value
    [  ELSE   false_valu  e  ]
 END
```

- if(condition, true_value)

  If condition is true, the column true_value is returned, otherwise null.

- if(condition, true_value, false_value)

  If condition is true, the column true_value is returned, otherwise the column false_value is returned.

Coalesce syntax

Coalesce returns the first non-null value for multiple columns.

```
Coalesce  ( value1 ,  value2  [,...])
```

NULLIF syntax

If value1 and value2 are equal, null is returned, otherwise value1 is returned.

```
nullif ( value1 ,  value2 )
```

TRY syntax

The try syntax can catch some of the underlying exceptions, such as the 0 error, to return a null value.

```
try ( expression )
```

6.8.20 Nested subquery

For some complicated query scenarios, you can use the SQL nested query to meet the complicated requirements when the one-level SQL cannot meet the requirements.

The difference between nested subquery and non-nested query is that you need to specify the from condition in the SQL statement. Specifying the keyword `from log` in the query indicates to read original data from the logs.

Example:

```
* |  select   sum ( pv )  from
(
 select   count ( 1 )  as   pv   from   log   group   by   method
)
```

6.8.21 Arrays

Statement	Meaning	Example
Subscript operator []	[] is used to obtain a certain element in the array.	-

Statement	Meaning	Example

Connection operator ||	|| is used to connect two arrays into one.	`SELECT ARRAY [ 1 ] || ARRAY [ 2 ]; – [ 1 , 2 ]`
		`SELECT ARRAY [ 1 ] || 2 ; – [ 1 , 2 ]`
		`SELECT 2 || ARRAY [ 1 ]; – [ 2 , 1 ]`

array_distinct	Obtain the distinct elements in the array by means of array deduplication.	-
array_intersect(x, y)	Obtain the intersection of arrays x and y.	-
array_union(x, y) → array	Obtain the union of arrays x and y.	-
array_except(x, y) → array	Obtain the subtraction of arrays x and y.	-
array_join(x, delimiter , null_replacement) → varchar	Join string arrays with the delimiter into a string and replace null values with null_replacement. Note: The maximum size of the returned result of this array_join is 1 KB. If the returned result exceeds 1 KB, the excessive part will be removed.	-
array_max(x) → x	Obtain the maximum value in array x.	
array_min(x) → x	Obtain the minimum value in array x.	-

Statement	Meaning	Example
array_position(x, element) → bigint	Obtain the subscript of the element in array x. The subscript starts from 1. 0 is returned if no subscript is found.	-
Array_remove (x, element)-array	Remove the element from the array.	-
array_sort(x) → array	Sort the array and move null values to the end.	-
cardinality(x) → bigint	Obtain the array size.	-
concat(array1, array2, ···, arrayN) → array	Concatenate arrays.	-
contains(x, element) → boolean	Returns TRUE if array x contains the element.	-
This is a Lambda function. See filter() in Lambda.	Concatenate a two-dimensional array into a one-dimensional array.	-
flatten(x) → array	Concatenate a two-dimensional array into a one-dimensional array.	-
reduce(array, initialState, inputFunction, outputFunction) → x	See function reduce() in *Lambda functions*.	-
reverse(x) → array	Sort array x in reverse order.	-
sequence(start, stop) → array	Generate a sequence from start to stop and increment each step by 1.	-
sequence(start, stop, step) → array	Generate a sequence from start to stop and increment each step by the specified step value.	-

Statement	Meaning	Example
sequence(start, stop, step) → array	Generate a timestamp array from start to stop. Start and stop are of the timestamp type. Step is of the interval type, which can be from DAY to SECOND, and can also be YEAR or MONTH.	-
shuffle(x) → array	Shuffle the array.	-
slice(x, start, length) → array	Create a new array with length elements from start in array x.	-
transform(array, function) → array	See transform() in *Lambda functions*.	-
zip(array1, array2[, ···]) → array	Merge multiple arrays. In the result, the Nth parameter in the Mth element is the Mth element in the Nth original array, which is equivalent to transposing multiple arrays.	`SELECT zip ( ARRAY [ 1 , 2 ], ARRAY [‘ 1b ’, null , ‘ 3b ’]); — [ ROW ( 1 , ‘ 1b ’), ROW ( 2 , null ), ROW ( null , ‘ 3b ’)]`
zip_with(array1, array2, function) → array	For more information, see *Lambda functions* zip_with.	-
array_agg (key)	array_agg (key) is an aggregation funtion that is used to aggregate all the content of the key column as an array to return.	* | select array_agg(key)

6.8.22 Binary string functions

The binary string type varbinary is different from the string type varchar.

Statement	Description
Connection function ||	The result of `a || b` is `ab`.
length(binary) → bigint	Returns the length in binary.
concat(binary1, ···, binaryN) → varbinary	Connect the binary strings, which is equivalent to ||.

Statement	Description
to_base64(binary) → varchar	Convert a binary string to a Base64 string .
from_base64(string) → varbinary	Convert a Base64 string to a binary string .
to_base64url(binary) → varchar	Convert a string to a URL-safe Base64 string.
from_base64url(string) → varbinary	Convert a URL-safe Base64 string to a binary string.
to_hex(binary) → varchar	Convert a binary string to a hexadecimal string.
from_hex(string) → varbinary	Convert a hexadecimal string to a binary string.
to_big_endian_64(bigint) → varbinary	Convert a number to a binary string in big endian mode.
from_big_endian_64(binary) → bigint	Convert a binary string in big endian mode to a number.
md5(binary) → varbinary	Calculate the MD5 value of a binary string.
sha1(binary) → varbinary	Calculate the SHA1 value of a binary string.
sha256(binary) → varbinary	Calculate the SHA256 hash value of a binary string.
sha512(binary) → varbinary	Calculate the SHA512 value of a binary string.
xxhash64(binary) → varbinary	Calculate the xxhash64 value of a binary string.

6.8.23 Bit operation

Statements	Description	Example

bit_count(x, bits) → bigint	Count the number of 1 in the binary expression of x.	`SELECT bit_count ( 9 , 64 ); — 2`
		`SELECT bit_count ( 9 , 8 ); — 2`
		`SELECT bit_count (- 7 , 64 ); — 62`
		`SELECT bit_count (- 7 , 8 ); — 6`

bitwise_and(x, y) → bigint	Perform the AND operation on x and y in the binary form.	-
bitwise_not(x) → bigint	Calculate the opposite values of all bits of x in the binary form.	-
bitwise_or(x, y) → bigint	Perform the OR operation on x and y in the binary form.	-
bitwise_xor(x, y) → bigint	Perform the XOR operation on x and y in the binary form.	-

6.8.24 Interval-valued comparison and periodicity-valued comparison functions

Interval-valued comparison and periodicity-valued comparison functions are used to compare the calculation results of the current period with those of a specified previous period.

Function	Description	Example
`compare ( value , time_windo w )`	This function compares the value calculated for the current period with that calculated by time_window. The values are double or long type values. The unit of time_window is seconds . The return values are in array format. Possible return values include the current value, the value before time_window, and the ratio of the current value to the value before time_window.	`*| select compare ( pv , 86400 ) from ( select count ( 1 ) as pv from log )`
`compare ( value , time_windo w1 , time_windo w2 )`	This function compares the current value with the values of periods before time_window1 and time_window2. The comparison results are in JSON array format, where the values must be placed in the following sequence : [current value, value before time_window1, value before time_windo w2, current value/value before time_window1, current value/value before time_window2].	`* | select compare ( pv , 86400 , 172800 ) from ( select count ( 1 ) as pv from log )`

Function	Description	Example
`compare ( value , time_windo w1 , time_windo w2 , time_windo w3 )`	This function compares the current value with the values of periods before time_window1 , time_window2 and time_window3. The comparison results are in JSON array format, where the values must be placed in the following sequence: [current value, value before time_window1, value before time_window2, value before time_windo w3, current value/Value before time_window1 , current value/value before time_window2, current value/value before time_window3].	`* | select compare ( pv , 86400 , 172800 , 604800 ) from ( select count ( 1 ) as pv from log )`
`compare_re sult ( value , time_windo w )`	This function works similarly to `compare ( value , time_windo w )`. However, the return values are string type values in the format of `" Current value ( Increased percentage %)"`. The increased percentage value is rounded to two decimal places by default.	`* | select compare_re sult ( pv , 86400 ) from ( select count ( 1 ) as pv from log )`

Function	Description	Example
`compare_re sult ( value , time_windo w1 , time_windo w2 )`	This function works similarly to `compare ( value , time_windo w1 , time_windo w2 )`. However, the return values are string type values in the format of `" Current value ( Increased percentage compared with the first period %) ( Increased percentage compared with the second period )"`. The increased percentage values are rounded to two decimal places by default.	`* | select compare_re sult ( pv , 86400 , 172800 ) from ( select count ( 1 ) as pv from log )`

Function	Description	Example
`ts_compare ( value , time_windo w )`	This function compares the current value with the values of periods before `time_windo w1` and `time_windo w2`. The comparison results are in JSON array format, where the values must be placed in the following sequence: [current value, value before time_window1, current value/value before time_window1, unix timestamp at the start time of the previous period]. This function is used to compare time series functions. This requires the GROUP BY operation to be included in SQL statements on the time column.	For example, `* | select t , ts_compare ( pv , 86400 ) as d from ( select date_trunc (' minute ', __time__ ) as t , count ( 1 ) as pv from log group by t order by t ) group by t` indicates that the function compares the calculation result of every minute in the current period with that of every minute in the last period. The comparison result is `d :[ 1251 . 0 , 1264 . 0 , 0 . 9897151898 734177 , 1539843780 . 0 , 1539757380 . 0 ] t : 2018 - 10 - 19 14 : 23 : 00 . 000 .`

Examples

- Calculate the ratio of the PV in the current hour to that in the same time period as yesterday.

  The start time is 2018-07-25 14:00:00, and the end time is 2018-07-25 15:00:00.

  Statement for query and analysis:

  ```
  * | select compare ( pv , 86400 ) from ( select count ( 1 ) as pv from log )
  ```

  where 86400 indicates that 86400 seconds are subtracted from the current period.

  Return result:

  ```
  [ 9 . 0 , 19 . 0 , 0 . 4736842105 2631579 ]
  ```

  where:

  - 9.0 is the PV value from 2018-07-25 14:00:00 to 2018-07-25 15:00:00.
  - 19.0 is the PV value from 2018-07-24 14:00:00 to 2018-07-24 15:00:00.
  - 0.47368421052631579 is the ratio of the PV value of the current period to that of a previous period.

  If you want to expand the array into three columns of numbers, the analysis statement is:

  ```
  * | select diff [ 1 ], diff [ 2 ], diff [ 3 ] from ( select compare ( pv , 86400 ) as diff from ( select count ( 1 ) as pv from log ))
  ```

- Calculate the ratio of the PV in every minute of the current hour to that in the same time period as yesterday, and display the results in a line chart.

  1. Calculate the ratio of the PV in every minute of the current hour to that in the same time period as yesterday. The start time is 2018-07-25 14:00:00, and the end time is 2018-07-25 15:00:00.

     Statement for query and analysis:

     ```
     *| select t , compare ( pv , 86400 ) as diff from ( select count ( 1 ) as pv , date_forma t ( from_unixt ime
     ```

```
  ( __time__ ), '% H :% i ')  as  t  from  log  group  by  t
  )  group  by  t  order  by  t
```

Return results:

t	diff
14:00	[9520.0,7606.0,1.2516434393899554]
14:01	[8596.0,8553.0,1.0050274757395066]
14:02	[8722.0,8435.0,1.0340248962655603]
14:03	[7499.0,5912.0,1.2684370771312586]

where t indicates the time in the format of `Hour : Minute`. The content of the diff column is an array containing the following:

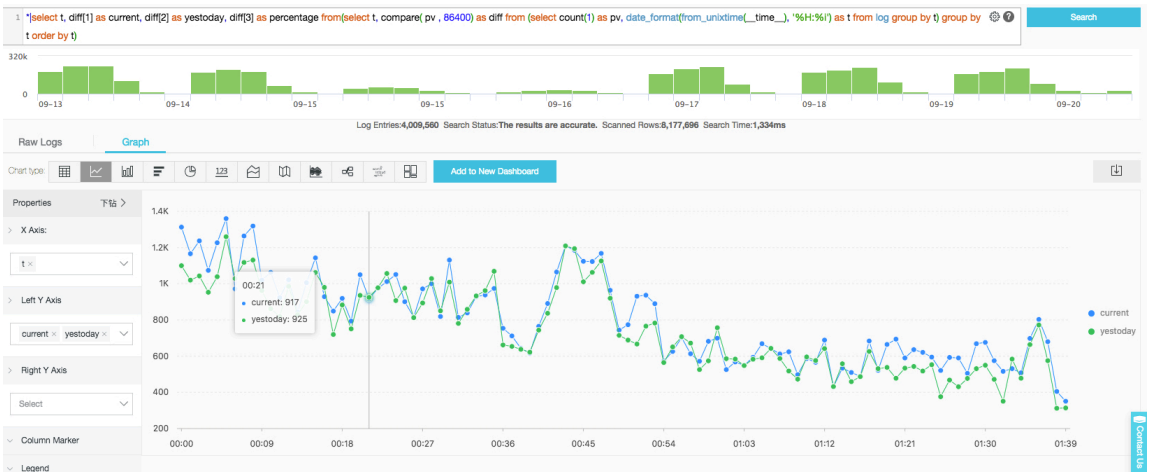
- The PV value of the current period.
- The PV value of the previous period.
- The ratio of the PV value in the current period to that in the previous period.

2. To show the query results in a line chart, use the following statement:

```
*| select  t , diff [ 1 ]  as  current , diff [ 2 ]  as
 yestoday , diff [ 3 ]  as  percentage  from ( select  t ,
 compare ( pv , 86400 )  as  diff  from ( select  count ( 1
 )  as  pv , date_forma t ( from_unixt ime ( __time__ ), '% H
 :% i ')  as  t  from  log  group  by  t )  group  by  t
 order  by  t )
```

The two lines indicate the PV values of today and yesterday.

Figure 6-19: Line chart



6.8.25 Comparison functions and operators

Comparison functions and operators

A comparison operation compares the values of two parameters, which can be used for any comparable types, such as int, bigint, double, and text.

Comparison operators

A comparison operator is used to compare two parameter values. During the comparison, if the logic is true, TRUE is returned. Otherwise, FALSE is returned.

Operator	Meaning
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
=	Equal to
<>	Not equal to
!=	Not equal to

Range operator BETWEEN

BETWEEN is used to determine whether a parameter value is between the values of two other parameters. The range is a closed interval.

- If the logic is true, TRUE is returned. Otherwise, FALSE is returned.

  Example: `SELECT 3 BETWEEN 2 AND 6 ;`. The logic is true, and TRUE is returned.

  The preceding example is equivalent to `SELECT 3 >= 2 AND 3 <= 6 ;`.
- BETWEEN can follow NOT to determine the opposite logic.

  Example: `SELECT 3 NOT BETWEEN 2 AND 6 ;`. The logic is false, and FALSE is returned.

  The preceding example is equivalent to `SELECT 3 < 2 OR 3 > 6 ;`.
- If the value of any parameter is NULL, NULL is returned.

IS NULL and IS NOT NULL

These operators are used to determine whether a parameter value is NULL.

IS DISTINCT FROM and IS NOT DISTINCT FROM

Similar to determining whether two values are equal or not, but these operators can determine whether a NULL value exists.

Example:

```
SELECT NULL IS DISTINCT FROM NULL ; -- false
SELECT NULL IS NOT DISTINCT FROM NULL ; -- true
```

As described in the following table, the DISTINCT operator can be used to compare parameter values in most cases.

a	b	a = b	a <> b	a DISTINCT b	a NOT DISTINCT b
1	1	TRUE	FALSE	FALSE	TRUE
1	2	FALSE	TRUE	TRUE	FALSE
1	NULL	NULL	NULL	TRUE	FALSE
NULL	NULL	NULL	NULL	FALSE	TRUE

GREATEST and LEAST

These operators are used to obtain the maximum or minimum values among multiple columns.

Example:

```
select greatest ( 1 , 2 , 3 ) ; -- 3 is returned .
```

Comparison conditions: ALL, ANY, and SOME

Comparison conditions are used to determine whether a parameter meets the specified conditions.

- ALL is used to determine whether a parameter meets all the conditions. If the logic is true, TRUE is returned. Otherwise, FALSE is returned.
- ANY is used to determine whether a parameter meets any of the conditions. If the logic is true, TRUE is returned. Otherwise, FALSE is returned.
- Same as ANY, SOME is used to determine whether a parameter meets any of the conditions.

· ALL, ANY, and SOME must immediately follow the comparison operators.

comparison and determination in many cases.

Expression	Meaning
A = ALL (…)	TRUE is returned when A is equal to all values.
A <> ALL (…)	TRUE is returned when A is not equal to all values.
A < ALL (…)	TRUE is returned when A is less than all values.
A = ANY (…)	TRUE is returned when A is equal to any value, which is equivalent to A IN (…).
A <> ANY (…)	TRUE is returned when A is not equal to any value.
A < ANY (…)	TRUE is returned when A is less than the maximum value.

Example:

```
SELECT ' hello ' = ANY ( VALUES ' hello ', ' world '); -- true
SELECT 21 < ALL ( VALUES 19 , 20 , 21 ); -- false
SELECT 42 >= SOME ( SELECT 41 UNION ALL SELECT 42
UNION ALL SELECT 43 ); -- true
```

6.8.26 Lambda functions

Lambda expressions

Lambda expressions are written with `->`.

Example:

```
x -> x + 1
( x , y ) -> x + y
x -> regexp_like e ( x , ' a +')
x -> x [ 1 ] / x [ 2 ]
x -> IF ( x > 0 , x , - x )
x -> COALESCE ( x , 0 )
x -> CAST ( x AS JSON )
x -> x + TRY ( 1 / 0 )
```

Most MySQL expressions can be used in Lambda.

filter(array<T>, function<T, boolean>) → ARRAY<T>

Filter data from an array and obtain only elements that the function returns TRUE.

Example:

```
SELECT filter ( ARRAY [], x -> true ); -- []
SELECT filter ( ARRAY [ 5 , - 6 , NULL , 7 ], x -> x > 0 );
-- [ 5 , 7 ]
SELECT filter ( ARRAY [ 5 , NULL , 7 , NULL ], x -> x IS
NOT NULL ); -- [ 5 , 7 ]
```

map_filter(map<K, V>, function<K, V, boolean>) → MAP<K,V>

Filter data from a map and obtain only element pairs that the function returns TRUE.

Example:

```
SELECT map_filter ( MAP ( ARRAY [], ARRAY []), ( k , v ) -> true
); -- {}
SELECT map_filter ( MAP ( ARRAY [ 10 , 20 , 30 ], ARRAY [' a ',
NULL , ' c ']), ( k , v ) -> v IS NOT NULL ); -- { 10 -> a
, 30 -> c }
SELECT map_filter ( MAP ( ARRAY [' k1 ', ' k2 ', ' k3 '], ARRAY [
20 , 3 , 15 ]), ( k , v ) -> v > 10 ); -- { k1 -> 20 , k3 -
> 15 }
```

reduce(array<T>, initialState S, inputFunction<S, T, S>, outputFunction<S, R>) → R

The reduce() function traverses each element in the array in turn from the initial state, calculates inputFunction(S,T) based on the state S, and generates a new state. It finally applies outputFunction to convert the final state S to the output result R.

1. Initial state S.
2. Traverse each element T.
3. Calculate inputFunction(S,T) and generate the new state S.
4. Repeat steps 2 and 3 until the last element is traversed and has the new state generated.
5. Uses the final state S to obtain the final output result R.

Example:

```
SELECT reduce ( ARRAY [], 0 , ( s , x ) -> s + x , s -> s
); -- 0
SELECT reduce ( ARRAY [ 5 , 20 , 50 ], 0 , ( s , x ) -> s +
x , s -> s ); -- 75
SELECT reduce ( ARRAY [ 5 , 20 , NULL , 50 ], 0 , ( s , x ) -
> s + x , s -> s ); -- NULL
SELECT reduce ( ARRAY [ 5 , 20 , NULL , 50 ], 0 , ( s , x ) -
> s + COALESCE ( x , 0 ), s -> s ); -- 75
SELECT reduce ( ARRAY [ 5 , 20 , NULL , 50 ], 0 , ( s , x ) -
> IF ( x IS NULL , s , s + x ), s -> s ); -- 75
SELECT reduce ( ARRAY [ 2147483647 , 1 ], CAST ( 0 AS
BIGINT ), ( s , x ) -> s + x , s -> s ); -- 2147483648
SELECT reduce ( ARRAY [ 5 , 6 , 10 , 20 ], -- calculates
arithmetic average : 10 . 25
```

```
               CAST ( ROW ( 0 . 0 ,  0 )  AS   ROW ( sum   DOUBLE ,
count   INTEGER )),
            ( s ,  x ) ->  CAST ( ROW ( x  +  s . sum ,  s . count
+  1 )  AS   ROW ( sum   DOUBLE ,  count   INTEGER )),
               s  ->  IF ( s . count  =  0 ,  NULL ,  s . sum  /  s .
count ));
```

transform(array<T>, function<T, U>) → ARRAY<U>

Calls function for each element in the array to generate the new result U.

Example:

```
SELECT   transform ( ARRAY  [],  x  ->  x  +  1 ); -- []
SELECT   transform ( ARRAY  [ 5 ,  6 ],  x  ->  x  +  1 ); -- [ 6 ,
7 ]  Increment   each   element   by   1 .
SELECT   transform ( ARRAY  [ 5 ,  NULL ,  6 ],  x  ->  COALESCE ( x
,  0 ) +  1 ); -- [ 6 ,  1 ,  7 ]
SELECT   transform ( ARRAY  [' x ', ' abc ', ' z '],  x  ->  x  || '
0 '); -- [' x0 ', ' abc0 ', ' z0 ']
SELECT   transform ( ARRAY  [ ARRAY  [ 1 ,  NULL ,  2 ],  ARRAY [ 3
,  NULL ]],  a  ->  filter ( a ,  x  ->  x   IS   NOT   NULL )); --
[[ 1 ,  2 ], [ 3 ]]
```

transform_keys(map<K1, V>, function<K1, V, K2>) → MAP<K2,V>

Apply the function for each key in the map in turn to generate a new key.

Example:

```
 SELECT   transform_  keys ( MAP ( ARRAY [],  ARRAY []), ( k ,  v ) -
>  k  +  1 ); -- {}
 SELECT   transform_  keys ( MAP ( ARRAY  [ 1 ,  2 ,  3 ],  ARRAY  ['
 a ', ' b ', ' c ']), ( k ,  v ) ->  k  +  1 ); -- { 2  ->  a ,  3  -
>  b ,  4  ->  c }  Increment   each   key   by   1 .
 SELECT   transform_  keys ( MAP ( ARRAY  [' a ', ' b ', ' c '],
 ARRAY  [ 1 ,  2 ,  3 ]), ( k ,  v ) ->  v  *  v ); -- { 1  ->  1 ,  4
  ->  2 ,  9  ->  3 }
 SELECT   transform_  keys ( MAP ( ARRAY  [' a ', ' b '],  ARRAY  [ 1
 ,  2 ]), ( k ,  v ) ->  k  ||  CAST ( v   as   VARCHAR )); -- { a1  -
>  1 ,  b2  ->  2 }
 SELECT   transform_  keys ( MAP ( ARRAY  [ 1 ,  2 ],  ARRAY  [ 1 . 0
 ,  1 . 4 ]), -- { one  ->  1 . 0 ,  two  ->  1 . 4 }
                     ( k ,  v ) ->  MAP ( ARRAY [ 1 ,  2 ],  ARRAY ['
 one ', ' two '])[ k ]);
```

transform_values(map<K, V1>, function<K, V1, V2>) → MAP<K, V2>

Apply the function for all values in the map, convert V1 to V2, and generate a new map `< K , V2 >`.

```
SELECT   transform_  values ( MAP ( ARRAY [],  ARRAY []), ( k ,  v )
->  v  +  1 ); -- {}
SELECT   transform_  values ( MAP ( ARRAY  [ 1 ,  2 ,  3 ],  ARRAY  [
10 ,  20 ,  30 ]), ( k ,  v ) ->  v  +  1 ); -- { 1  ->  11 ,  2  ->
22 ,  3  ->  33 }
```

```
SELECT transform_ values ( MAP ( ARRAY [ 1 , 2 , 3 ], ARRAY
[' a ', ' b ', ' c ']), ( k , v ) -> k * k ); -- { 1 -> 1 , 2
-> 4 , 3 -> 9 }
SELECT transform_ values ( MAP ( ARRAY [' a ', ' b '], ARRAY [
1 , 2 ]), ( k , v ) -> k || CAST ( v as VARCHAR )); -- { a
-> a1 , b -> b2 }
SELECT transform_ values ( MAP ( ARRAY [ 1 , 2 ], ARRAY [ 1 .
0 , 1 . 4 ]), -- { 1 -> one_1 . 0 , 2 -> two_1 . 4 }
                    ( k , v ) -> MAP ( ARRAY [ 1 , 2 ], ARRAY
[' one ', ' two '])[ k ] || ' _ ' || CAST ( v AS VARCHAR ));
```

zip_with(array<T>, array<U>, function<T, U, R>) → array<R>

Merge two arrays, and specify the elements of the newly generated array by using the function. Element T in the first array and element U in the second array are used to generate the new result R.

Example:

```
SELECT zip_with ( ARRAY [ 1 , 3 , 5 ], ARRAY [' a ', ' b ', ' c
'], ( x , y ) -> ( y , x )); -- Transpose the elements of
the two arrays to generate a new array . Result : [
ROW (' a ', 1 ), ROW (' b ', 3 ), ROW (' c ', 5 )]
SELECT zip_with ( ARRAY [ 1 , 2 ], ARRAY [ 3 , 4 ], ( x , y ) -
> x + y ); -- Result : [ 4 , 6 ]
SELECT zip_with ( ARRAY [' a ', ' b ', ' c '], ARRAY [' d ', ' e
', ' f '], ( x , y ) -> concat ( x , y )); Concatenat e the
elements of the two arrays to generate a new string
. Result : [' ad ', ' be ', ' cf ']
```

map_zip_with(map<K, V1>, map<K, V2>, function<K, V1, V2, V3>) → map<K, V3>

Merge two maps, use values V1 and V2 to generate V3 based on each key, and generate a new map`< K , V3 >`.

```
SELECT map_zip_wi th ( MAP ( ARRAY [ 1 , 2 , 3 ], ARRAY [' a ',
' b ', ' c ']),
                    MAP ( ARRAY [ 1 , 2 , 3 ], ARRAY [' d ', ' e
', ' f ']),
                                        ( k , v1 , v2 ) -> concat (
v1 , v2 )); Merge values which have the same map keys
. -- { 1 -> ad , 2 -> be , 3 -> cf }
SELECT map_zip_wi th ( MAP ( ARRAY [' k1 ', ' k2 '], ARRAY [ 1 ,
2 ]),
                    MAP ( ARRAY [' k2 ', ' k3 '], ARRAY [ 4 , 9 ]),
                   ( k , v1 , v2 ) -> ( v1 , v2 )); Generate an
  array by using the two values . -- { k1 -> ROW ( 1 ,
null ), k2 -> ROW ( 2 , 4 ), k3 -> ROW ( null , 9 )}
SELECT map_zip_wi th ( MAP ( ARRAY [' a ', ' b ', ' c '], ARRAY [
1 , 8 , 27 ]),
                    MAP ( ARRAY [' a ', ' b ', ' c '], ARRAY [ 1 ,
2 , 3 ]),
                   ( k , v1 , v2 ) -> k || CAST ( v1 / v2 AS
VARCHAR )); -- Concatenat es the key values and division
```

```
results of the two values -- { a -> a1 , b -> b4 , c
-> c9 }
```

6.8.27 Logical functions

Logical operators

Table 6-3: Logical operators

Operator	Description	Example
AND	Returns TRUE only when both the left and right operands are TRUE.	a AND b
OR	Returns TRUE if either the left or right operand is TRUE.	a OR b
NOT	Returns TRUE only when the right operand is FALSE.	NOT a

NULL involved in logical operation

The following table lists the true values when the values of a and b are TRUE, FALSE, and NULL respectively.

Table 6-4: Truth Table 1

a	b	a AND b	A or B
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL
NULL	TRUE	NULL	TRUE
NULL	FALSE	FALSE	NULL
NULL	NULL	NULL	NULL

Table 6-5: Truth Table 2

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

6.8.28 Column alias

In the SQL standard, the column name must be consisted of English letters, numbers, and underlines (_) and start with an English letter.

If a column name (for example, User-Agent) that does not conform to the SQL standard is configured in the log collection configuration, give the column an alias used for query on the page of configuring statistical properties. The alias is only used for the SQL statistics. In the underlying storage, the column name is the original name. Use the original column name to query.

Besides, you can give the column an alias to replace the original column name for query when the column name is very long.

Table 6-6: Alias Example:

Original column name	Alias
User-Agent	ua
User.Agent	ua
123	col
abceefghijklmnopqrstuvw	a

6.8.29 Geospatial functions

Geospatial concept

Geospatial functions support the geometries in the Well-Known Text (WKT) format.

Table 6-7: Geometry format

Geometry	Well-maid text (WKT) Format
Point	`POINT ( 0 0 )`

Geometry	Well-maid text (WKT) Format
Line string	`LINESTRING ( 0 0 , 1 1 , 1 2 )`
Polygon	`Polygon`
Multi-point	`MULTIPOINT ( 0 0 , 1 2 )`
Multi-line string	`MULTILINES TRING (( 0 0 , 1 1 , 1 2 ), ( 2 3 , 3 2 , 5 4 ))`
Multi-polygon	`MULTIPOLYG ON ((( 0 0 , 4 0 , 4 4 , 0 4 , 0 0 ), ( 1 1 , 2 1 , 2 2 , 1 2 , 1 1 )), ((- 1 - 1 , - 1 - 2 , - 2 - 2 , - 2 - 1 , - 1 - 1 )))`
Geometry collection	`GEOMETRYCO LLECTION ( POINT ( 2 3 ), LINESTRING ( 2 3 , 3 4 ))`

Constructors

Table 6-8: Constructors Description

Function	Description
ST_Point(double, double) → Point	Returns a geometry type point with the given coordinate values.
ST_LineFromText(varchar) → LineString	Returns a geometry type line string from WKT representation.
ST_Polygon(varchar) → Polygon	Returns a geometry type polygon from WKT representation.
ST_GeometryFromText(varchar) → Geometry	Returns a geometry type object from WKT representation.
ST_AsText(Geometry) → varchar	Returns the WKT representation of the geometry.

Operations

Function	Description
ST_Boundary(Geometry) → Geometry	Returns the closure of the combinatorial boundary of this geometry.
ST_Buffer(Geometry, distance) → Geometry	Returns the geometry that represents all points whose distance from the specified geometry is less than or equal to the specified distance.
ST_Difference(Geometry, Geometry) → Geometry	Returns the geometry value that represents the point set difference of the given geometries.
ST_Envelope(Geometry) → Geometry	Returns the bounding rectangular polygon of a geometry.
ST_ExteriorRing(Geometry) → Geometry	Returns a line string representing the exterior ring of the input polygon.
ST_Intersection(Geometry, Geometry) → Geometry	Returns the geometry value that represents the point set intersection of two geometries.
ST_SymDifference(Geometry, Geometry) → Geometry	Returns the geometry value that represents the point set symmetric difference of two geometries.

Relationship tests

Function	Description
ST_Contains(Geometry, Geometry) → boolean	Returns true if and only if no points of the second geometry lie in the exterior of the first geometry, and at least one point of the interior of the first geometry lies in the interior of the second geometry . Returns false if the two geometries at least share an interior point.
ST_Crosses(Geometry, Geometry) → boolean	Returns true if the supplied geometries have some, but not all, interior points in common.
ST_Disjoint(Geometry, Geometry) → boolean	Returns true if the given geometries do not spatially intersect.
ST_Equals(Geometry, Geometry) → boolean	Returns true if the given geometries represent the same geometry.

Function	Description
ST_Intersects(Geometry, Geometry) → boolean	Returns true if the given geometries spatially intersect in two dimensions (share any portion of space) and false if they do not (they are disjoint).
ST_Overlaps(Geometry, Geometry) → boolean	Returns true if the given geometries share space, are of the same dimension, but are not completely contained by each other.
ST_Relate(Geometry, Geometry) → boolean	Returns true if the first geometry is spatially related to the second geometry.
ST_Touches(Geometry, Geometry) → boolean	Geometry) → boolean Returns true if the given geometries have at least one point in common, but their interiors do not intersect.
ST_Within(Geometry, Geometry) → boolean	Returns true if the first geometry is completely inside the second geometry. Returns false if the two geometries have at least one point in common.

Accessors

Function	Description
ST_Area(Geometry) → double	Returns the area of a polygon using Euclidean measurement on a two dimensional plane in projected units.
ST_Centroid(Geometry) → Geometry	Returns the point value that is the mathematical centroid of a geometry.
ST_CoordDim(Geometry) → bigint	Returns the coordinate dimension of the geometry.
ST_Dimension(Geometry) → bigint	Returns the inherent dimension of this geometry, which must be less than or equal to the coordinate dimension.
ST_Distance(Geometry, Geometry) → double	Returns the minimum distance between two geometries.
ST_IsClosed(Geometry) → boolean	Returns true if the start and end points of the line string are coincident.
ST_IsEmpty(Geometry) → boolean	Returns true if this geometry is an empty geometry collection, polygon, or point.

Function	Description
ST_IsRing(Geometry) → boolean	Returns true if and only if the line is closed and simple.
ST_Length(Geometry) → double	Returns the length of a line string or multi-line string using Euclidean measurement on a two dimensional plane (based on spatial ref) in projected units.
ST_XMax(Geometry) → double	Returns Y maxima of a bounding box of a geometry.
ST_YMax(Geometry) → double	Returns Y maxima of a bounding box of a geometry.
T_XMin(Geometry) → double	Returns X minima of a bounding box of a geometry.
ST_YMin(Geometry) → double	Returns Y minima of a bounding box of a geometry.
ST_StartPoint(Geometry) → point	Returns the first point of a line string geometry.
ST_EndPoint(Geometry) → point	Returns the last point of a line string geometry.
ST_X(Point) → double	Returns the X coordinate of the point.
ST_Y(Point) → double	Returns the Y coordinate of the point.
ST_NumPoints(Geometry) → bigint	Returns the number of points in a geometry.
ST_NumInteriorRing(Geometry) → bigint	Returns the number of interior rings of a polygon.

6.8.30 Geo functions

For more information about functions that determine the country, province, city, ISP, and the longitude and latitude of specified IP addresses, see *IP functions*.

Table 6-9: Geo functions

Function	Description	Example
geohash(string)	Returns the geohash value of the specified geographic al coordinate. The geographical coordinate is represented by a string in the format of "latitude, longitude" (the values for latitude and longitude are separated by a comma).	`select geohash (' 34 . 1 , 120 . 6 ')= ' wwjcbrdnzs '`
geohash(lat, lon)	Returns the geohash value of the specified geographic al coordinate. The geographical coordinate is represented by two separate parameters that indicate the latitude and longitude.	`select geohash ( 34 . 1 , 120 . 6 )= ' wwjcbrdnzs '`

6.8.31 Join syntax

Join is used for combining fields from multiple tables. Besides Join for a single Logstore, Log Service also supports Join for Logstore and RDS, and for several Logstores. This document describes how to use the Join function between Logstores.

Procedure

1. *Download* the latest version of Python SDK.
2. Use the GetProjectLogs interface for query.

SDK sample

```
/ usr / bin / env   python
# encoding :  utf - 8
 import   time , sys , os
 From   aliyun .  log .  logexcepti  on   import   logexcepti  on
 from   aliyun . log . logitem   import   LogItem
 from   aliyun . log . logclient   import   LogClient
 From   aliyun .  log .  getlogsreq  uest   import   getlogsreq  uest
 from   aliyun . log . getproject  logsreques  t   import
 GetProject  LogsReques  t
 from   aliyun . log . putlogsreq  uest   import   PutLogsReq  uest
 from   aliyun . log . listtopics  request   import   ListTopics
 Request
 from   aliyun . log . listlogsto  resrequest   import   ListLogsto
 resRequest
```

```
from aliyun . log . gethistogr amsrequest import GetHistogr
amsRequest
from aliyun . log . index_conf ig import *
from aliyun . log . logtail_co nfig_detai l import *
from aliyun . log . machine_gr oup_detail import *
from aliyun . log . acl_config import *
if __name__ ==' __main__ ':
    token = None
    endpoint = " http :// cn - hangzhou . log . aliyuncs . com "
    accessKeyI d = ' LTAIvKy7U '
    accessKey =' 6gXLNTLyCf dsfwrewrfh dskfdsfuiw u '
    client = LogClient ( endpoint , accessKeyI d , accessKey ,
token )
    logstore = " meta "
   # In the query statement , specify two Logstores .
For each Logstore specify its time range and the
key
    req = GetProject LogsReques t ( project ," select count
( 1 ) from sls_operat ion_log s join meta m on s .
__date__ >' 2018 - 04 - 10 00 : 00 : 00 ' and s . __date__ <
' 2018 - 04 - 11 00 : 00 : 00 ' and m . __date__ >' 2018 - 04 -
23 00 : 00 : 00 ' and m . __date__ <' 2018 - 04 - 24 00 : 00
: 00 ' and s . projectid = cast ( m . ikey as varchar )");
    Res = client . Fig ( req )
    res . log_print ();
    exit ( 0 )
```

6.8.32 UNNEST function

Scenario

Columns of log data are usually of a primitive data type, such as string or number. In certain scenarios with more complex data structures, the columns of log data may involve complex data types, such as arrays, maps, and JSON objects. The UNNEST function can be used to enumerate an array of complex data into rows for easier querying and analysis.

Example:

```
__source__ : 1 . 1 . 1 . 1
__tag__ : __hostname __ : vm - req - 1701032323 16569850 -
tianchi111 932 . tc
__topic__ : TestTopic_ 4
array_colu mn : [ 1 , 2 , 3 ]
double_col umn : 1 . 23
map_column : {" a ": 1 ," b ": 2 }
text_colum n : Product
```

The `array_colu mn` field is of array type. To obtain an aggregate of all elements in an `array_colu mn` array, you must enumerate all elements of the array for each row.

UNNEST function

Syntax	Description
`unnest ( array ) as table_alias ( column_nam e )`	Splits the specified array into multiple rows. Each row has a name specified in the `column_nam e` column.
`unnest ( map ) as table ( key_name , value_name )`	Splits the specified map into multiple rows. Each key has a key name specified in the `key_name` column, and a value specified in the `value_name` column.



Note:

Note: The UNNEST function takes only arrays or maps. If you specify a string, the string must represent a JSON object. Then you can parse the string into an array or map by using the `cast ( json_parse ( array_colu mn ) as array ( bigint ))` syntax.

Enumerate the elements of an array

Split an array into multiple rows using SQL:

```
* | select array_colu mn , a from log , unnest ( cast (
 json_parse ( array_colu mn ) as array ( bigint ) ) ) as t ( a
 )
```

The UNNEST function splits the array into multiple rows and stores the rows in a new table referenced as t, with each column referenced as a.

- Calculate the sum of the elements in the array:

```
* | select sum ( a ) from log , unnest ( cast ( json_parse
 ( array_colu mn ) as array ( bigint ) ) ) as t ( a )
```

- Perform a GROUP BY operation on the array by a (each element of the array):

```
* | select a , count ( 1 ) from log , unnest ( cast (
 json_parse ( array_colu mn ) as array ( bigint ) ) ) as t (
 a ) group by a
```

Enumerate the elements of the map

- Enumerate the elements of the map:

```
* | select map_column , a , b from log , unnest ( cast (
 json_parse ( map_column ) as map ( varchar , bigint ) ) ) as
 t ( a , b )
```

- Perform a GROUP BY operation on the map by key:

```
* | select key , sum ( value ) from log , unnest ( cast (
 json_parse ( map_column ) as map ( varchar , bigint ) ) ) as
 t ( key , value ) GROUP BY key
```

Visualize histogram, numeric_histogram query results

- histogram

  The histogram function is similar to the COUNT GROUP BY syntax. For more information about the syntax, see *Mapping function*.

  The histogram function usually returns a JSON string that cannot be directly visualized. The following is an example:

```
* | select histogram ( method )
```

  You can use the UNNEST function to split JSON data into multiple rows. The following is an example:

```
* | select key , value from ( select histogram ( method )
 as his from log ) , unnest ( his ) as t ( key , value )
```

- Numeric_histogram

The numeric_histogram syntax sorts values in a numeric column into multiple bins. This operation is similar to performing a GROUP BY operation on the numeric column. For more information about the syntax, see *Estimating functions*.

```
* | select numeric_histogram ( 10 , Latency )
```

Use the following query statement to visualize the result:

```
* | select key , value from ( select numeric_histogram ( 10
 , Latency ) as his from log ) , unnest ( his ) as t (
 key , value )
```

6.8.33 Phone number functions

Phone number functions are used to query the attributes of phone numbers that are registered in Mainland China.

Functions

Function name	Description	Example
mobile_province	This function is used to query the provincial attribute of a phone number. The phone number must be of the numeric type. If the phone number is of the string type, you can use `try_cast` to convert the type to numeric.	`* | select mobile_province ( 12345678 )` `* | select mobile_province ( try_cast ( ' 12345678 ' as bigint ) )`
mobile_city	This function is used to query the urban attribute of a phone number. The phone number must be of the numeric type. If the phone number is of the string type, you can use `try_cast` to convert the type to numeric.	`* | select mobile_city ( 12345678 )` `* | select mobile_city ( try_cast (' 12345678 ' as bigint ) )`

Function name	Description	Example
mobile_carrier	This function is used to query the telecom operator to which a phone number belongs. The phone number must be of the numeric type. If the phone number is of the string type, you can use `try_cast` to convert the type to numeric.	`* | select mobile_carrier ( 12345678 )` `* | select mobile_carrier ( try_cast (' 12345678 ' as bigint ) )`

Scenarios

- Query phone number attributes and generate a report.

  Assume that an e-commerce company collects logs about the activities of its customers. The company can extract the fields involving phone numbers, and then collects the attributes of the phone numbers by using the following query statement:

  ```
  SELECT mobile_cit y ( try_cast (" mobile " as bigint )) as " city ", mobile_pro vince ( try_cast (" mobile " as bigint )) as " province ", count ( 1 ) as " number of requests " group by " province ", " city " order by " number of request " desc limit 100
  ```

  In the statement, `mobile` is used as the input field of the `mobile_cit y` and `mobile_pro vince` functions to show the provinces and cities of the phone numbers. The returned information from the preceding query is shown in the following figure.

  You can also choose to show the phone number attributes on a map, as shown in the following figure.

- Check phone number attributes and report abnormal logon information.

  If a telecom operator wants to filter its customers whose daily locations are different from their phone number attributes (according to additional attributes and frequently accessed IP addresses), the following statement can be used:

  ```
  * | select mobile , client_ip , count ( 1 ) as PV where mobile_cit y ( try_cast (" mobile " as bigint )) ! = ip_to_city ( client_ip ) and ip_to_city ( client_ip ) ! = '' group by client_ip , mobile order by PV desc
  ```

  Furthermore, you can create alarm rules that use phone number attributes.

6.9 Machine learning syntax and functions

6.9.1 Introduction

Log Service provides a machine learning feature that supports multiple algorithms and calling methods. During log query and analysis, you can use SELECT statements and machine learning functions to call machine learning algorithms to analyze the characteristics of a field or fields within a period of time.

In particular, Log Service offers diversified time series analysis algorithms to help you quickly solve problems related to time series prediction, time series anomaly detection, sequence decomposition, and multi-time series clustering. Additionally, the algorithms are compatible with standard SQL interfaces, which greatly simplifies use of the algorithms and improves troubleshooting efficiency.

Features

- Various smooth operations on single-time series sequences are supported.
- Algorithms related to prediction, anomaly detection, change point detection , inflexion point detection, and multi-period estimation of single-time series sequences are supported.
- Decomposition operations on single-time series sequences are supported.
- Various clustering algorithms of multi-time series sequences are supported.
- Multi-field pattern mining (based on numeric or text columns) are supported.

Limits

- The input time series data must be sampled from the same interval.
- The input time series data cannot contain data repeatedly sampled from the same time point.

Item	Limit
Valid capacity of time-series data processing	Data collected from a maximum of 150,000 consecutive time points If the limit is exceeded, you need to aggregate the data or reduce the sample data amount.
Clustering capacity of the density-based clustering algorithm	A maximum of 5,000 time series curves, each of which cannot contain more than 1,440 time points

Item	Limit
Clustering capacity of the hierarchical clustering algorithm	A maximum of 2,000 time series curves, each of which cannot contain more than 1,440 time points

Functions

	Category	Function	Description
Time series	*Smooth function*	ts_smooth_simple	This function uses the Holt Winters algorithm to smooth time series data.
		ts_smooth_fir	This function uses the FIR filter to smooth time series data.
		ts_smooth_iir	This function uses the IIR filter to smooth time series data.
	Multi-period estimation function	ts_period_detect	This function estimates period information in a time series.
	Change point detection function	ts_cp_detect	This function finds intervals with different statistical characteristics within a time series. The interval endpoints are change points.
		ts_breakout_detect	This function finds the time point when statistics abruptly increase or decrease within a time series.
	Prediction and anomaly detection function	ts_predicate_simple	This function models time series data by using default parameters and performs simple time series prediction and anomaly detection.

	Category	Function	Description
		ts_predicate_ar	This function models time series data by using an autoregressive model and performs simple time series prediction and anomaly detection.
		ts_predicate_arma	This function models time series data by using the autoregressive moving average model and performs simple time series prediction and anomaly detection.
		ts_predicate_arima	This function models time series data by using the autoregressive integrated moving average model with differences and performs simple time series prediction and anomaly detection.
	Sequence decomposition function	ts_decompose	This function uses the STL algorithm to decompose the sequence of time series data.
	Time series clustering function	ts_density_cluster	This function clusters time series data by using the density-based clustering method.
		ts_hierarchical_cluster	This function clusters time series data by using the hierarchical clustering method.
		ts_similar_instance	This function queries curves that are similar to a specified curve.

	Category	Function	Description
Pattern mining	*Frequent pattern statistics*	pattern_stat	This function indicates the frequent pattern in statistical patterns. It is used to mine representative combinations of attributes among the given multi-attribute field samples.
	Differential pattern statistics	pattern_diff	This function finds the pattern that causes differences between two sets under specified conditions.

6.9.2 Smooth function

The smooth function is used to smooth and filter input time series curves. Filtering is the first step to discover the shape of a time series curve.

Function list

Function	Description
`ts_smooth_simple`	This function is the default smooth function, which uses the Holt Winters algorithm to smooth time series data.
`ts_smooth_fir`	This function uses the FIR filter to smooth time series data.
`ts_smooth_iir`	This function uses the IIR filter to smooth time series data.

ts_smooth_simple

Function format:

```
select ts_smooth_simple(x, y)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds

Parameter	Description	Value
y	Numeric column corresponding to the data at a specified time point	--

Example:

- Statement for query and analysis:

```
* | select ts_smooth_simple ( stamp , value ) from ( select __time__ - __time__ % 120 as stamp , avg ( v ) as value from log GROUP BY stamp order by stamp )
```

- Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Data timestamp in seconds
Longitudinal axis	src	Data before filtering
	filter	Data after filtering

ts_smooth_fir

Function format:

- When filter parameters are undetermined, you can use the parameters in the built-in window to filter time series data:

```
select ts_smooth_fir(x, y,winType,winSize,samplePeriod,sampleMethod)
```

- When filter parameters are specified, you can set them as needed:

```
select ts_smooth_fir(x, y,array[],samplePeriod,sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
winType	Filter window type	Value range: · rectangle: rectangle window · hanning: hanning window · hamming: hamming window · blackman: blackman window Note: We recommend that you set this parameter to rectangle for better display.
winSize	Length of the filter window	Long type values ranging form 2 to 15
array[]	Specific FIR filter parameters	Values in array format, sum of Array is 1.0. For example, [0.2, 0.4, 0.3, 0.1].
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Example:

· Statement for query and analysis:

```
* | select ts_smooth_ fir ( stamp , value , ' rectangle ', 4
 , 1 , ' avg ') from ( select __time__ - __time__ % 120
```

```
    as   stamp ,  avg ( v )  as   value   from   log   GROUP   BY
 stamp   order   by   stamp  )
```

Result:



- **Statement for query and analysis:**

```
* |  select   ts_smooth_  fir ( stamp ,  value ,  array [ 0 . 2 ,  0
 . 4 ,  0 . 3 ,  0 . 1 ],  1 , ' avg ')  from  (  select   __time__
 -  __time__  %  120   as   stamp ,  avg ( v )  as   value   from
 log   GROUP   BY   stamp   order   by   stamp  )
```

Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Unixtime timestamp in seconds
Longitudinal axis	src	Data before filtering
	filter	Data after filtering

ts_smooth_iir

Function format:

```
select
 ts_smooth_iir(x, y, array[], array[], samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
array[]	Specific parameter of x_i in the IIR filter algorithm	Array values with a length ranging from 2 to 15. Sum of Array is 1.0. For example, array[0.2, 0.4, 0.3, 0.1]
array[]	Specific parameter of y_{i-1} in the IIR filter algorithm	Array values with a length ranging from 2 to 15. Sum of Array is 1.0. For example, array[0.2, 0.4, 0.3, 0.1]
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the sampling window · max: maximum value of the data in the sampling window · min: minimum value of the data in the sampling window · sum: sum of the data in the sampling window

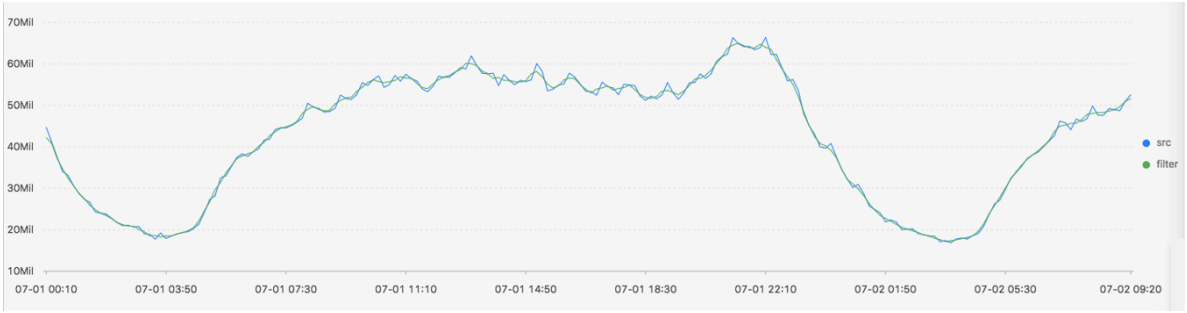
Example:

· Statement for query and analysis:

```
* | select ts_smooth_iir ( stamp , value , array [ 0 . 2 ,
 0 . 4 , 0 . 3 , 0 . 1 ], array [ 0 . 4 , 0 . 3 , 0 . 3 ], 1
 , ' avg ') from ( select __time__ - __time__ % 120 as
```

```
stamp , avg ( v ) as value from log GROUP BY stamp
order by stamp )
```

- Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Unixtime timestamp in seconds
Longitudinal axis	src	Data before filtering
	filter	Data after filtering

6.9.3 Multi-period estimation function

The multi-period estimation function can estimate time series in different time periods and perform a series of operations, such as Fourier transform, to extract period data.

ts_period_detect

The function estimates time series data on a period basis.

Function format:

```
select
 ts_period_detect(x, y,minPeriod,maxPeriod,samplePeriod,sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
minPeriod	Ratio of the minimum length of the pre-estimation period to the total length of the time series	Value range: (0, 1]

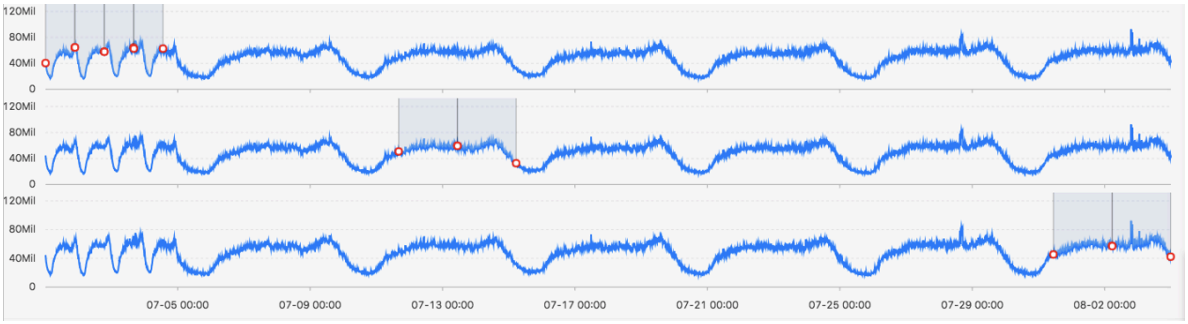
Parameter	Description	Value
maxPeriod	Ratio of the maximum length of the pre-estimation period to the total length of the time series Note: The value of *maxPeriod* must be greater than that of *minPeriod*.	Value range: (0, 1]
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Example:

· Statement for query and analysis:

```
* | select ts_period_ detect ( stamp , value , 0 . 2 , 1 . 0
, 1 , ' avg ') from ( select __time__ - __time__ % 120
as stamp , avg ( v ) as value from log GROUP BY
stamp order by stamp )
```

· Result:



The following table describes the display items.

Display item	Description
period_id	Array composed of period IDs with an array length of 1. The value 0.0 indicates the original series.
time_series	Timestamp sequence
data_series	Result for each timestamp · It indicates the original sequence when period_id is 0.0. · It indicates the period estimation result after filtering when period_id is not 0.0.

6.9.4 Change point detection function

The change point detection function detects change points in time series data.

The change point detection function supports two types of change points:

- Changes of statistical characteristics within a specified time period
- Obvious faulting in a sequence

Function list

Function	Description
ts_cp_detect	This function finds intervals with different statistical characteristics within a time series. The interval endpoints are change points.
ts_breakout_detect	This function finds the time point when statistics steeply increase or decrease within a time series.

ts_cp_detect

Function format:

- If you are not sure about the window size, use the ts_cp_detect function in the following format. Then, the algorithm called by the function will use a window with a length of 10 to detect change points.

```
select ts_cp_detect(x, y, amplePeriod,sampleMethod)
```

- If you need to adjust the display effect of a service curve, use the ts_cp_detect function in the following format. Then, you can optimize the display effect by setting the minSize parameter.

```
select ts_cp_detect(x, y, minSize, samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
minSize	Minimum length of consecutive intervals	The minimum value is 3, and the maximum value cannot exceed 1/10 of the length of the current input data.
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

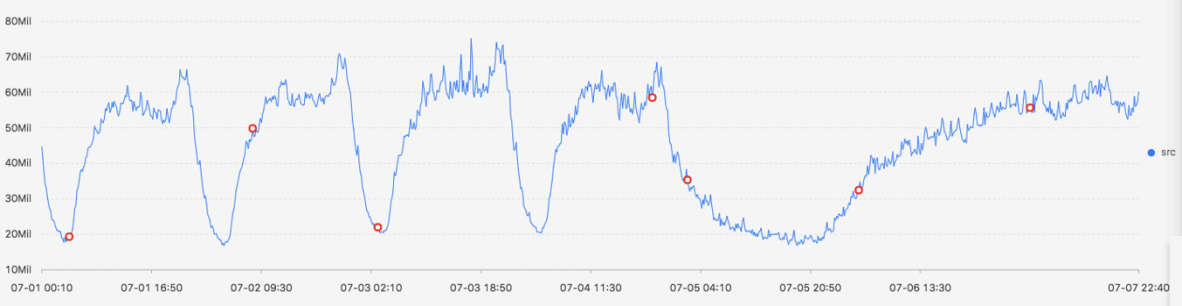
Example:

- Statement for query and analysis:

```
* | select ts_cp_dete ct ( stamp , value , 3 , 1 , ' avg ')
 from ( select __time__ - __time__ % 10 as stamp , avg
```

```
( v ) as value from log GROUP BY stamp order by stamp )
```

- Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Data timestamp in seconds, for example, 1537071480
Longitudinal axis	src	Data before filtering, for example, 1956092.7647745228
	prob	Probability that a point is a change point . Its value ranges from 0 to 1.

ts_breakout_detect

Function format:

```
select ts_breakout_detect(x, y, winSize, samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
winSize	Minimum length of consecutive intervals	The minimum value is 3, and the maximum value cannot exceed 1/10 of the length of the current input data.
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds

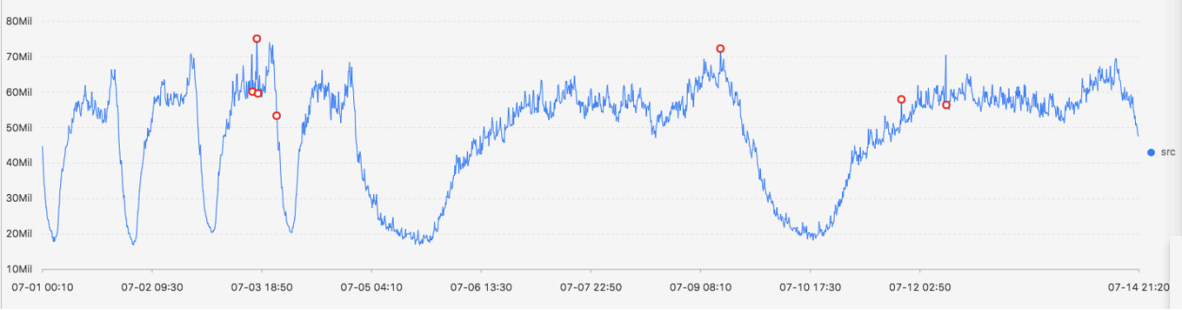
Parameter	Description	Value
`sampleMethod`	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Example:

· Statement for query and analysis:

```
* | select ts_breakou t_detect ( stamp , value , 3 , 1 , '
 avg ') from ( select __time__ - __time__ % 10 as stamp
 , avg ( v ) as value from log GROUP BY stamp order
  by stamp )
```

· Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Data timestamp in seconds, for example, 1537071480
Longitudinal axis	src	Data before filtering, for example, 1956092.7647745228
	prob	Probability that a point is a change point . Its value ranges from 0 to 1.

6.9.5 Maximum value detection function

The maximum value detection function is used to identify the maximum value of a sequence in a specified window.

Function format:

```
select ts_find_peaks(x, y, winSize, samplePeriod, sampleMethod)
```

The following table describes the function parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
winSize	Minimum length of a window	Long-type values range from 1 to the actual data length. We recommend that you specify one tenth of the actual data length as *winSize*.
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

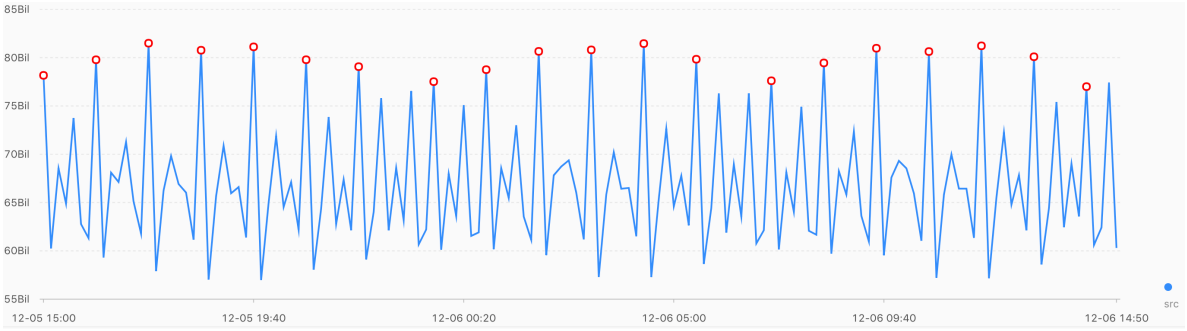
Examples:

· Statement for query and analysis:

```
*  and   h  :  nu2h05202 . nu8   and   m :  NET  |   select
 ts_find_pe  aks ( stamp ,  value ,  30 ,  1 , ' avg ')  from  (
```

```
select __time__ - __time__ % 10 as stamp , avg ( v ) as value from log GROUP BY stamp order by stamp )
```

- Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Data timestamp in seconds, for example, 1537071480
Longitudinal axis	src	Data before filtering, for example, 1956092.7647745228
	peak_flag	Whether the current point has the maximum value. Value range: · 1.0: The point has the maximum value. · 0.0: The point does not have the maximum value.

6.9.6 Prediction and anomaly detection function

The prediction and anomaly detection function detects anomalies by predicting time series curves and identifying Ksigma and quantiles of the errors between a predicted curve and an actual curve.

Function list

Function	Description
ts_predicate_simple	This function models time series data by using default parameters and performs simple time series prediction and anomaly detection.
ts_predicate_ar	This function models time series data by using an autoregressive model and performs simple time series prediction and anomaly detection.

Function	Description
ts_predicate_arma	This function models time series data by using an autoregressive moving model and performs simple time series prediction and anomaly detection.
ts_predicate_arima	This function models time series data by using an autoregressive moving model with differences and performs simple time series prediction and anomaly detection.



Note:

The display items of the prediction and anomaly detection function are consistent. For details about the results and descriptions, see *Example of the ts_predicate_simple output result*.

ts_predicate_simple

Function format:

```
select ts_predicate_simple(x, y, nPred, samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
nPred	Number of points for prediction	Long type values ranging from 1 to 5 x *p*
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds

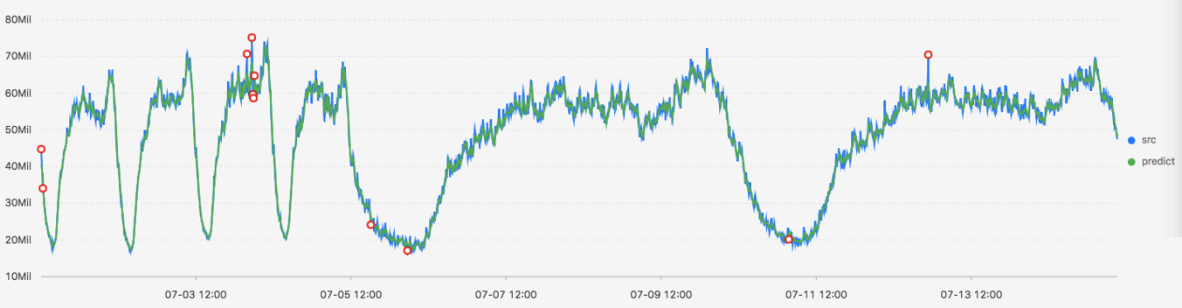
Parameter	Description	Value
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Example:

· Statement for query and analysis:

```
* | select ts_predica te_simple ( stamp , value , 6 , 1 , '
 avg ') from ( select __time__ - __time__ % 60 as stamp
 , avg ( v ) as value from log GROUP BY stamp order
  by stamp )
```

· Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Unixtime timestamp in seconds
Longitudinal axis	src	Raw data
	predict	Data after filtering
	upper	Upper limit of the prediction. By default , the current confidence is 0.85, which is unmodifiable.
	lower	Lower limit of the prediction. By default , the current confidence is 0.85, which is unmodifiable.

Display item		Description
	anomaly_prob	Probability that a point is an anomaly point. Its value ranges from 0 to 1.

ts_predicate_ar

Function format:

```
select ts_predicate_ar(x, y, p, nPred, samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
p	Order of the autoregressive model	Long type values ranging from 2 to 8
nPred	Number of points for prediction	Long type values ranging from 1 to 5 x *p*
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Statement for query and analysis:

```
* | select ts_predica te_ar ( stamp , value , 3 , 4 , 1 , '
 avg ') from ( select __time__ - __time__ % 60 as stamp ,
```

```
 avg ( v )  as   value   from   log   GROUP   BY   stamp   order   by
   stamp )
```

ts_predicate_arma

Function format:

```
select
 ts_predicate_arma(x, y, p, q, nPred, samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
p	Order of the autoregressive model	Long type values ranging from 2 to 100
q	Order of the moving average model	Long type values ranging from 2 to 8
nPred	Number of points for prediction	Long type values ranging from 1 to 5 x *p*
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Statement for query and analysis:

```
* |  select   ts_predica  te_arma ( stamp ,  value ,  3 ,  2 ,  4 ,
 1 , ' avg ')  from  ( select   __time__  -  __time__  %  60   as
```

```
 stamp ,  avg ( v )  as   value   from   log   GROUP   BY   stamp
 order   by   stamp )
```

ts_predicate_arima

Function format:

```
select
 ts_predicate_arima(x, y, p, d, q nPred, samplePeriod, sampleMethod)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
p	Order of the autoregressive model	Long type values ranging from 2 to 8
d	Order of the difference model	Long type values ranging from 1 to 3
q	Order of the moving average model	Long type values ranging from 2 to 8
nPred	Number of points for prediction	Long type values ranging from 1 to 5 x *p*
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Statement for query and analysis:

```
* |  select   ts_predica  te_arima ( stamp ,  value ,  3 ,  1 ,  2 ,
 4 ,  1 , ' avg ')  from  ( select   __time__  -  __time__  %  60   as
```

```
    stamp ,  avg ( v )  as   value   from   log   GROUP   BY   stamp
  order   by   stamp )
```

6.9.7 Sequence decomposition function

The sequence decomposition function can decompose service curves and highlight information about the curve trends and periods.

ts_decompose

Function format:

```
select ts_decompose(x, y, samplePeriod, sampleMethod)
```

The following table describes the parameters.

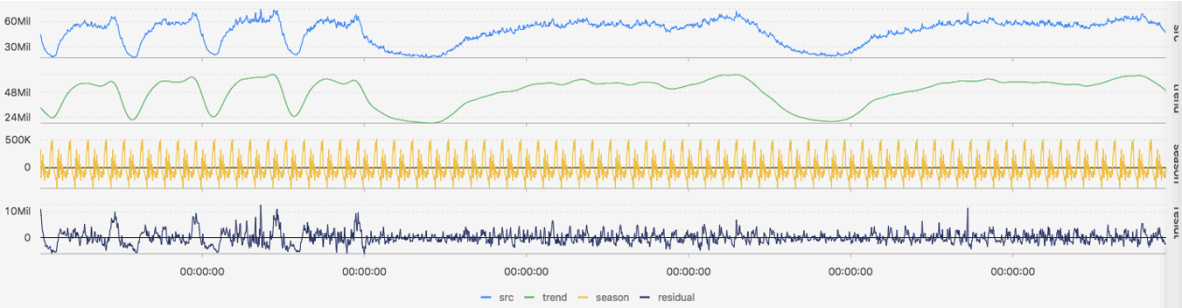
Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
samplePeriod	Period during which the current time series data is sampled	Long type values ranging from 1 to 86399 seconds
sampleMethod	Method for sampling the data in the sampling window	Value range: · avg: average value of the data in the window · max: maximum value of the data in the window · min: minimum value of the data in the window · sum: sum of the data in the window

Example:

- Statement for query and analysis:

```
* | select ts_decompose(stamp, value, 1, 'avg') from
  (select __time__ - __time__ % 60 as stamp, avg(v)
  as value from log GROUP BY stamp order by stamp)
```

- Result:



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	Unixtime timestamp in seconds
Longitudinal axis	src	Raw data
	trend	Curve trend after decomposition
	season	Curve period after decomposition
	residual	Residual data after decomposition

6.9.8 Time series clustering function

The time series clustering function is used to automatically cluster input time series data into different curve shapes. After that, the function quickly finds the corresponding clustering centers and curves with shapes that are different from the existing curve shapes.

Function list

Function	Description
ts_density_cluster	This function clusters time series data by using the density-based clustering method.
ts_hierarchical_cluster	This function clusters time series data by using the hierarchical clustering method.
ts_similar_instance	This function queries curves that are similar to a specified curve.

ts_density_cluster

Function format:

```
select ts_density_cluster(x, y, z)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
z	Metric name corresponding to the data at a specified time point	String type values, for example, machine01.cpu_usr

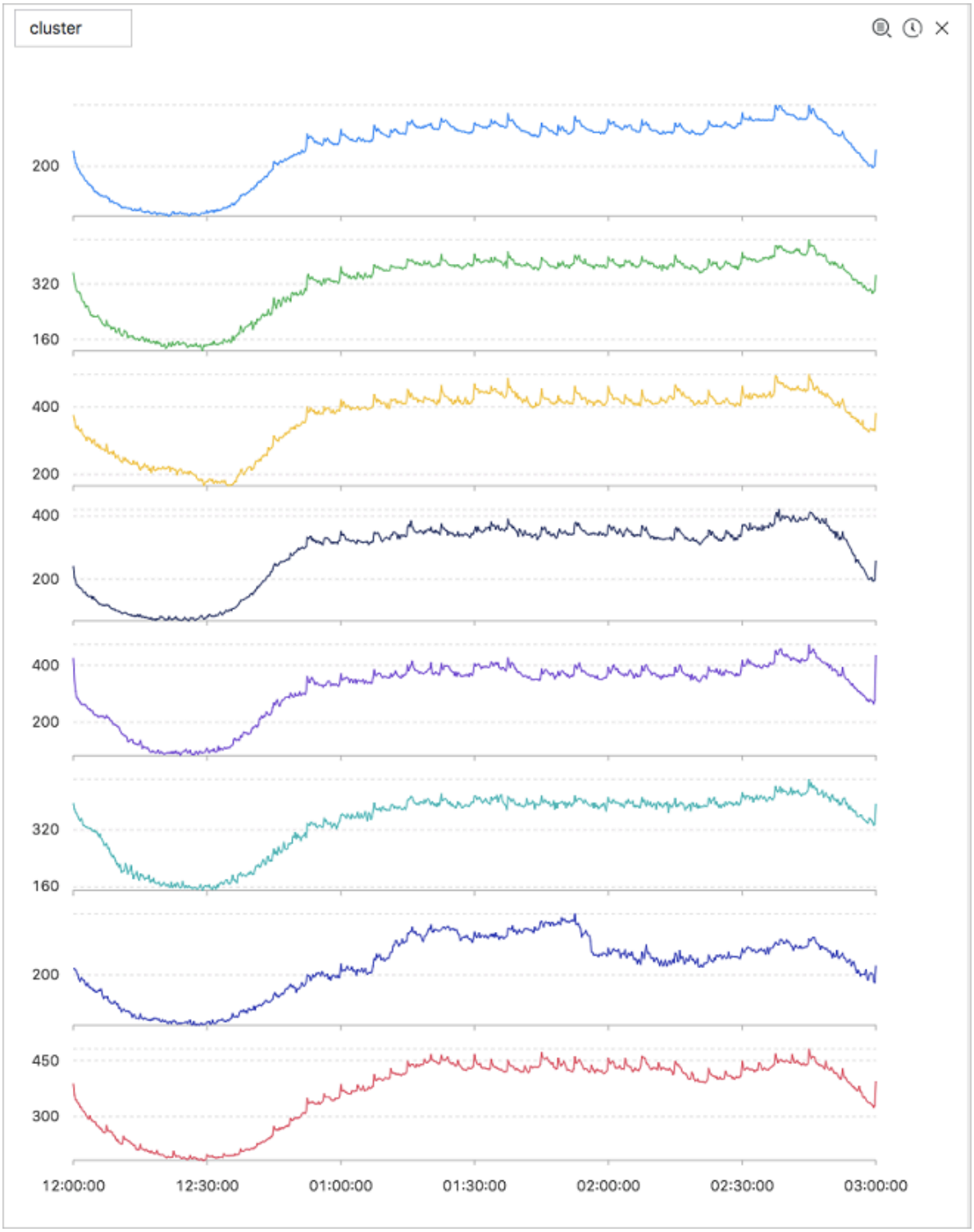
Example:

- **Statement for query and analysis:**

```
*  and  ( h : " machine_01 "  OR   h : " machine_02 "  OR   h
 : " machine_03 ") |  select   ts_density  _cluster ( stamp ,
 metric_val  ue , metric_nam  e  )  from  (  select   __time__
 -  __time__  %  600   as   stamp ,  avg ( v )  as   metric_val
```

```
ue , h as metric_nam e from log GROUP BY stamp ,
metric_nam e order BY metric_nam e , stamp )
```

- **Result:**



The following table describes the display items.

Display item	Description
cluster_id	Clustering type. The value -1 indicates that the clustering cannot be categorized into any clustering centers.
rate	Proportion of instances in the clustering
time_series	Timestamp sequence of the clustering center
data_series	Data sequence of the clustering center
instance_names	Set of instances included in the clustering center
sim_instance	Name of an instance in the clustering

ts_hierarchical_cluster

Function format:

```
select ts_hierarchical_cluster(x, y, z)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
z	Metric name corresponding to the data at a specified time point	String type values, for example, machine01.cpu_usr

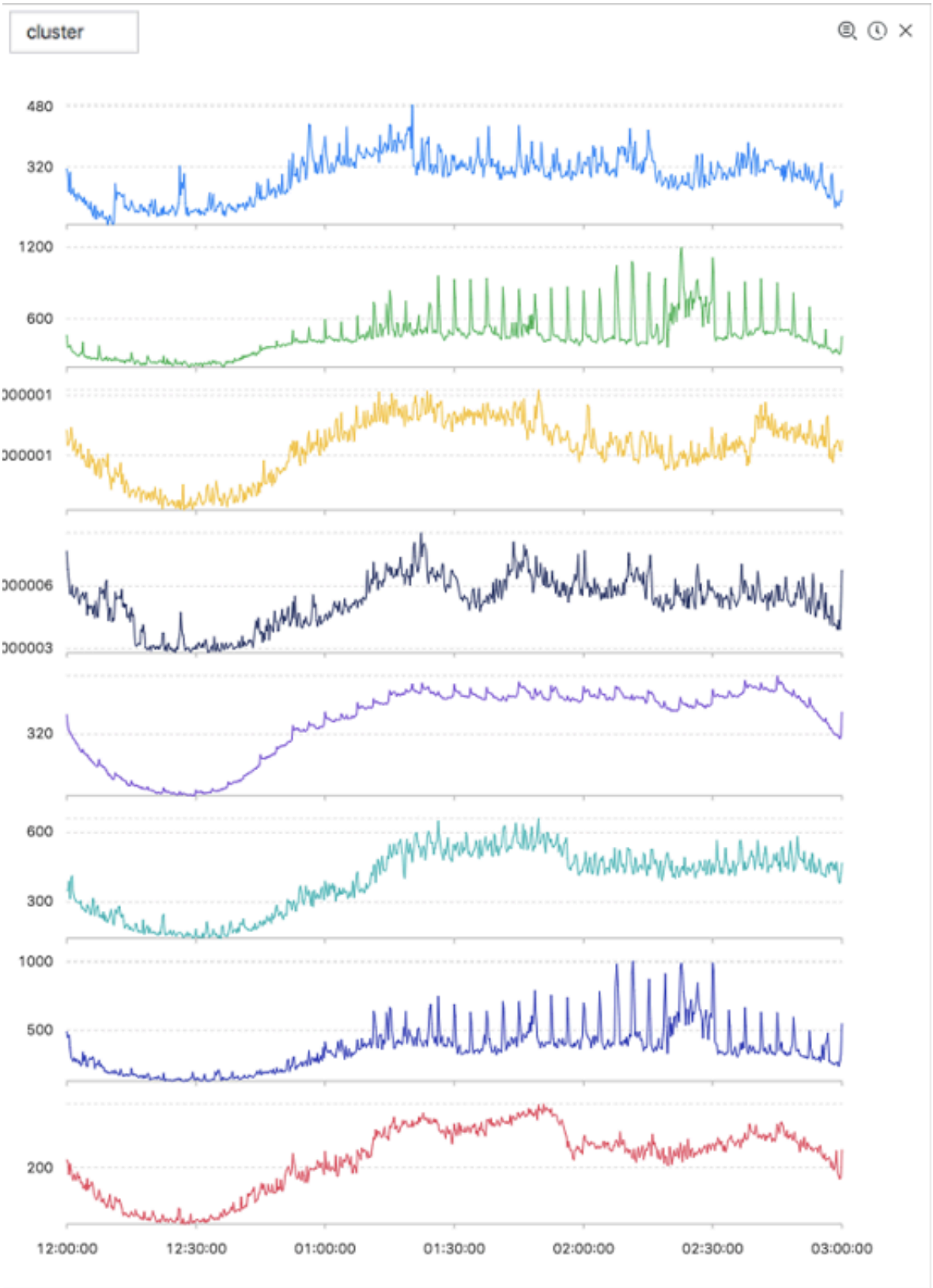
Example:

- Statement for query and analysis:

```
* and ( h : " machine_01 " OR h : " machine_02 " OR h :
 " machine_03 ") | select ts_hierarc hical_clus ter ( stamp
 , metric_val ue , metric_nam e ) from ( select __time__
 - __time__ % 600 as stamp , avg ( v ) as metric_val
```

```
ue , h as metric_nam e from log GROUP BY stamp ,
metric_nam e order BY metric_nam e , stamp )
```

- **Result:**



The following table describes the display items.

Display item	Description
cluster_id	Clustering type. The value -1 indicates that the clustering cannot be categorized into any clustering centers.
rate	Proportion of instances in the clustering
time_series	Timestamp sequence of the clustering center
data_series	Data sequence of the clustering center
instance_names	Set of instances included in the clustering center
sim_instance	Name of an instance in the clustering

ts_similar_instance

Function format:

```
select ts_similar_instance(x, y, z, instance_name)
```

The following table describes the parameters.

Parameter	Description	Value
x	Time column in ascending order	Unixtime timestamp in seconds
y	Numeric column corresponding to the data at a specified time point	-
z	Metric name corresponding to the data at a specified time point	String type values, for example, machine01.cpu_usr
instance_name	Name of a specified metric to be queried	String values in the *z* set, for example, machine01.cpu_usr Note: The metric must be an existing one.

Statement example for query and analysis:

```
* and ( h : " machine_01 " OR h : " machine_02 " OR h
 : " machine_03 ") | select ts_similar _instance ( stamp ,
 metric_val ue , metric_nam e , ' nu4e01524 . nu8 ' ) from (
 select __time__ - __time__ % 600 as stamp , avg ( v ) as
   metric_val ue , h as metric_nam e from log GROUP BY
   stamp , metric_nam e order BY metric_nam e , stamp )
```

The following table describes the display items.

Display item	Description
instance_name	Result list containing metrics that are similar to the specified metric
time_series	Timestamp sequence of the clustering center
data_series	Data sequence of the clustering center

6.9.9 Frequent pattern statistical function

The frequent pattern statistical function mines representative combinations of attributes from the given multi-attribute field samples to summarize the current logs.

pattern_stat

Function format:

```
select pattern_stat(array[col1, col2, col3], array['col1_name',
 'col2_name', 'col3_name'], array[col5, col6], array['col5_name',
 'col6_name'], supportScore, sample_ratio)
```

The following table describes the parameters.

Parameter	Description	Value
array[col1, col2, col3]	Input column composed of character type values	Values in array format, for example, array[clientIP, sourceIP, path, logstore]
array['col1_name', 'col2_name', 'col3_name']	Name corresponding to the input column composed of character type values	Values in array format, for example, array['clientIP', 'sourceIP', 'path', 'logstore']
array[col5, col6]	Input column composed of numeric values	Values in array format, for example, array[Inflow, OutFlow]
array['col5_name', 'col6_name']	Name corresponding to the input column composed of numeric values	Values in array format, for example, array['Inflow', 'OutFlow']
supportScore	Support level of positive and negative samples for pattern mining	Double type values. Range: (0,1].
sample_ratio	Sampling ratio with the default value of 0.1, which indicates that only 10% of the total samples are used	Double type values. Range: (0,1].

Example:

- Statement for query and analysis:

```
* | select pattern_st at ( array [ Category , ClientIP ,
 ProjectNam e , LogStore , Method , Source , UserAgent ],
 array [ ' Category ', ' ClientIP ', ' ProjectNam e ', ' LogStore
 ', ' Method ', ' Source ', ' UserAgent ' ], array [ InFlow ,
 OutFlow ], array [ ' InFlow ', ' OutFlow ' ], 0 . 45 , 0 . 3 )
 limit 1000
```

- Result:

count	supportscore	pattern
468235	0.9880626809484018	InFlow >= 0.0 and InFlow <= 60968.7 and OutFlow >= 0.0 and OutFlow <= 15566.4
459356	0.9693263443991458	Status = '200' and OutFlow >= 0.0 and OutFlow <= 15566.4
458757	0.9680623433187309	Status = '200' and InFlow >= 0.0 and InFlow <= 60968.7
456228	0.9627256843331392	InFlow >= 0.0 and InFlow <= 60968.7 and Status = '200' and OutFlow >= 0.0 and OutFlow <= 15566.4
417662	0.8813442725346703	InFlow >= 0.0 and InFlow <= 60968.7 and UserAgent = 'sls-cpp-sdk v0.6' and Status = '200'
417662	0.8813442725346703	UserAgent = 'sls-cpp-sdk v0.6' and InFlow >= 0.0 and InFlow <= 60968.7
415133	0.8760076135490787	OutFlow >= 0.0 and OutFlow <= 15566.4 and InFlow >= 0.0 and InFlow <= 60968.7 and UserAgent = 'sls-cpp-sdk v0.6' and Status = '200'
415133	0.8760076135490787	OutFlow >= 0.0 and OutFlow <= 15566.4 and UserAgent = 'sls-cpp-sdk v0.6' and InFlow >= 0.0 and InFlow <= 60968.7
415133	0.8760076135490787	OutFlow >= 0.0 and OutFlow <= 15566.4 and UserAgent = 'sls-cpp-sdk v0.6' and Status = '200'
415133	0.8760076135490787	UserAgent = 'sls-cpp-sdk v0.6' and OutFlow >= 0.0 and OutFlow <= 15566.4
414167	0.8739691744110473	InFlow >= 0.0 and InFlow <= 60968.7 and Method = 'PullData' and Status = '200'
414167	0.8739691744110473	Method = 'PullData' and InFlow >= 0.0 and InFlow <= 60968.7

The following table describes the display items.

Display item	Description
count	Number of samples for the current pattern
supportScore	Support level for the current pattern
pattern	Pattern content, which is organized in the format of conditional queries

6.9.10 Differential pattern statistical function

Based on the given multi-attribute field samples and conditions, the differential pattern statistical function analyzes the set of differential patterns affecting the

conditions. This helps you quickly diagnose the causes for the differences between the conditions.

pattern_diff

Function format:

```
select
 pattern_diff(array_char_value, array_char_name, array_numeric_value, array_numer
```

The following table describes the parameters.

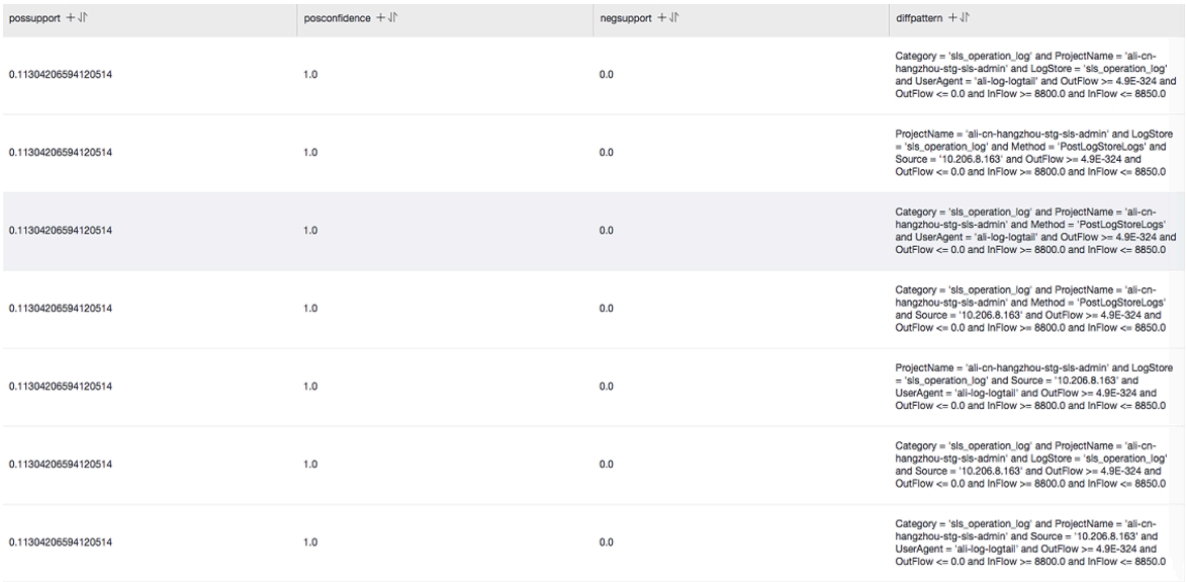
Parameter	Description	Value
array_char_value	Input column composed of character type values	Values in array format, for example, array[clientIP, sourceIP, path, logstore]
array_char_name	Name corresponding to the input column composed of character type values	Values in array format, for example, array['clientIP', 'sourceIP', 'path', 'logstore']
array_numeric_value	Input column composed of numeric values	Values in array format, for example, array[Inflow, OutFlow]
array_numeric_name	Name corresponding to the input column composed of numeric values	Values in array format, for example, array['Inflow', 'OutFlow']
condition	Data filtering condition. True indicates positive samples, and False indicates negative samples.	For example: latency ⩽ 300
supportScore	Support degree of positive and negative samples for pattern mining	Double type values. Range: (0,1].
posSampleRatio	Sampling ratio of positive samples with a default value of 0.5, which indicates that only half of the positive samples are used	Double type values. Range: (0,1].
negSampleRatio	Sampling ratio of negative samples with a default value of 0.5, which indicates that only half of the negative samples are used	Double type values. Range: (0,1].

Example:

- Statement for query and analysis:

```
* | select pattern_diff(array[Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent], array['Category', 'ClientIP', 'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent'], array[InFlow, OutFlow], array['InFlow', 'OutFlow'], Latency > 300, 0.2, 0.1, 1.0) limit 1000
```

- Result:



possupport	posconfidence	negsupport	diffpattern
0.11304206594120514	1.0	0.0	Category = 'sls_operation_log' and ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and LogStore = 'sls_operation_log' and UserAgent = 'ali-log-logtail' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0
0.11304206594120514	1.0	0.0	ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and LogStore = 'sls_operation_log' and Method = 'PostLogStoreLogs' and Source = '10.206.8.163' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0
0.11304206594120514	1.0	0.0	Category = 'sls_operation_log' and ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and Method = 'PostLogStoreLogs' and UserAgent = 'ali-log-logtail' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0
0.11304206594120514	1.0	0.0	Category = 'sls_operation_log' and ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and Method = 'PostLogStoreLogs' and Source = '10.206.8.163' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0
0.11304206594120514	1.0	0.0	ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and LogStore = 'sls_operation_log' and Source = '10.206.8.163' and UserAgent = 'ali-log-logtail' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0
0.11304206594120514	1.0	0.0	Category = 'sls_operation_log' and ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and LogStore = 'sls_operation_log' and Source = '10.206.8.163' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0
0.11304206594120514	1.0	0.0	Category = 'sls_operation_log' and ProjectName = 'ali-cn-hangzhou-stg-sls-admin' and Source = '10.206.8.163' and UserAgent = 'ali-log-logtail' and OutFlow >= 4.9E-324 and OutFlow <= 0.0 and InFlow >= 8800.0 and InFlow <= 8850.0

The following table describes the display items.

Display item	Description
possupport	Support level of positive samples for the mined pattern
posconfidence	Confidence of positive samples for the mined pattern
negsupport	Support level of negative samples for the mined pattern
diffpattern	Content of the mined pattern

6.10 Advanced analysis

6.10.1 Case study

Case list

1. *Trigger an alarm when the error 500 percentage increases rapidly*

2. *Trigger an alarm when traffic decreases sharply*
3. *Calculate the average latency of each bucket set by data interval*
4. *Return percentages in GROUP BY results*
5. *Count the number of logs that meet the query condition*

Trigger an alarm when the error 500 percentage increases rapidly

Count the percentage of error 500 every minute. An alarm is triggered when the percentage exceeds 40% in the last five minutes.

```
status : 500 | select __topic__ , max_by ( error_coun t ,
window_tim e )/ 1 . 0 / sum ( error_coun t ) as error_rati o ,
sum ( error_coun t ) as total_erro r from (
select __topic__ , count (*) as error_coun t , __time__
- __time__ % 300 as window_tim e from log group by
__topic__ , window_tim e

group by __topic__ having max_by ( error_coun t ,
window_tim e )/ 1 . 0 / sum ( error_coun t ) > 0 . 4 and sum (
error_coun t ) > 500 order by total_erro r desc limit
100
```

Trigger an alarm when traffic decreases sharply

Count the traffic every minute. An alarm is triggered when traffic decreases sharply recently. Data in the last one minute does not cover a full minute. Therefore, divide the statistical value by (max(time) - min(time)) for normalization to count the average traffic per minute.

```
* | SELECT SUM ( inflow ) / ( max ( __time__ ) - min ( __time__
)) as inflow_per _minute , date_trunc (' minute ', __time__ )
as minute group by minute
```

Calculate the average latency of each bucket set by data interval

```
* | select avg ( latency ) as latency , case when
originSize < 5000 then ' s1 ' when originSize < 20000
then ' s2 ' when originSize < 500000 then ' s3 ' when
```

```
originSize < 100000000 then ' s4 ' else ' s5 ' end as os
group by os
```

Return percentages in GROUP BY results

List the count results of different departments and the related percentages. This query combines subquery and window functions. sum(c) over() indicates to calculate the sum of values in all rows.

```
* | select department , c * 1 . 0 / sum ( c ) over () from (
select count ( 1 ) as c , department from log groupby
department )
```

Count the number of logs that meet the query condition

We must count the URLs by characteristics.In this situation, use the CASE WHEN syntax. You can also use the count_if syntax, which is simpler.

```
* | select count_if ( uri like '% login ') as login_num
, count_if ( uri like '% register ') as register_n um ,
date_forma t ( date_trunc (' minute ', __time__ ), '% m -% d % H
:% i ') as time group by time order by time limit
100
```

6.10.2 Optimize query for analysis

The analysis efficiency varies from query to query. Common ways to optimize the query are as follows for your references:

Avoid running Group By on string columns if possible

Running Group By on strings leads to a large amount of hash calculations, which usually accounts for more than 50% of total calculations.

For example:

```
* | select count ( 1 ) as pv , date_trunc (' hour ', __time__
) as time group by time
* | select count ( 1 ) as pv , from_unixt ime ( __time__ -
__time__ % 3600 ) as time group by __time__ - __time__ %
3600
```

Both Query 1 and Query 2 calculate the log count value every hour. However, Query 1 converts time into a string, for example, `2017 - 12 - 12 00 : 00 : 00`, and then runs Group By on this string. Query 2 calculates the on-the-hour time value, runs Group By on the result, and then converts the value into a string. Query 1 is less efficient than Query 2 because the former one needs to hash strings.

List fields with relatively large dictionary values on top when running Group By on multiple columns

For example, 13 provinces have 100 million users.

```
Fast : * |  select   province , uid , count ( 1 )  group   by
province , uid
Slow : * |  select   province , uid , count ( 1 )  group   by   uid ,
province
```

Estimating functions

provide much stronger performance than accurate calculation. Estimation sacrifices some acceptable accuracy for fast calculation.

```
Fast : * |  select   approx_dis  tinct ( ip )
Slow : * |  select   count ( distinct ( ip ))
```

Retrieve required columns in SQL and do not read all columns if possible

Use the query syntax to retrieve all columns. To speed up calculation, retrieve only the required columns in SQL if possible.

```
Fast : * | select   a , b   c
Slow : * | select  *
```

Non-group by columns, as far as possible in aggregate Functions

For example, userid, user name, must be one corresponding, we just need to follow userid for group.

```
Fast :  * |  select   userid ,  arbitrary ( username ),  count ( 1 )
groupby   userid
```

```
Slow : * |  select   userid ,  username ,  count ( 1 ) groupby
userid , username
```

6.11 Use JDBC to query and analyze logs

In addition to *Overview*, you can use JDBC and standard SQL 92 for log query and analysis.

Connection parameters

Connection parameter	Example	Description
host	regionid.example.com	*Service endpoint*The access point, Currently, only the intranet access of classic network and Virtual Private Cloud (VPC) access are supported.
port	10005	Use 10005 as the port by default.
user	bq2sjzesjmo86kq	The *AccessKey ID* .
password	4fdO1fTDDuZP	The *AccessKey Secret*.
database	sample-project	The *project* under your account.
table	sample-logstore	The *Logstore* under project.

For example, use a MySQL command to connect to Log Service as follows:

```
mysql  - hcn - shanghai - intranet . log . aliyuncs . com  -
ubq2sjzesj  mo86kq  - p4fdO1fTDD  uZP  - P10005
use   sample - project ; // Use   a   project .
```

Prerequisites

You must use the AccessKey of the main account or a sub-account to access the JDBC interface. The sub-account must belong to the project owner and have the project-level read permission.

Syntax description

Instructions

The WHERE condition must contain `__date__` or `__time__` to limit the time range of query. The type of `__date__` is timestamp, and the type of `__time__` is bigint.

Example:

- `__date__ > ' 2017 - 08 - 07 00 : 00 : 00 ' and __date__ < ' 2017 - 08 - 08 00 : 00 : 00 '`
- `__time__ > 1502691923 and __time__ < 1502692923`

At least one of the preceding conditions must be contained.

Filter syntax

The filter syntax in the WHERE condition is as follows:

Meaning	Example	Description
String search	`key = " value "`	Results after word segmentation are queried.
String fuzzy search	`key = " valu *"`	Results of fuzzy match after word segmentation are queried.
Value comparison	`num_field > 1`	Comparison operators including `>`, `>=`, `=`, `<` and `<=` are supported.
Logic operations	`and or not`	For example, `a = " x " and b =" y "` or `a = " x " and not b =" y "`.
Full-text search	`__line__ =" abc "`	Full-text index search requires the special key (`__line__`).

Computation syntax

For supported computation operators, see *Analysis syntax*.

SQL92 syntax

The SQL92 syntax is a combination of filter and computation syntaxes.

The following query is used as an example:

```
status > 200 | select avg ( latency ), max ( latency ) , count ( 1 ) as c GROUP BY method ORDER BY c DESC LIMIT 20
```

The filter part and time condition in the query can be combined into a new query condition based on standard SQL92 syntax.

```
select avg ( latency ), max ( latency ) , count ( 1 ) as c
from sample - logstore where status > 200 and __time__ >=
```

```
1500975424   and   __time__  <  1501035044   GROUP   BY   method
ORDER   BY   c   DESC   LIMIT   20
```

Access Log Service by using JDBC protocol

Program call

Developers can use the MySQL syntax to connect to Log Service in any program that supports MySQL connector. For example, JDBC or Python MySQLdb can be used.

Example:

```
import   com . mysql . jdbc .*;
Import   java .  SQL  .*;
import   java . sql . Connection ;
import   java . sql . ResultSetM  etaData ;
import   java . sql . Statement ;
public   class   testjdbc  {
    public   static   void   main ( String   args []){
        Connection   conn  =  null ;
        Statement   stmt  =  null ;
        try  {
           // STEP   2 :  Register   JDBC   driver
            Class . forName (" com . mysql . jdbc . Driver ");
           // STEP   3 :  Open   a   connection
            System . out . println (" Connecting   to   a   selected
  database ...");
            conn  =  DriverMana  ger . getConnect  ion (" jdbc :
mysql :// cn - shanghai - intranet . log . aliyuncs . com : 10005 /
sample - project "," accessid "," accesskey ");
            System . out . println (" Connected   database
successful  ly ...")
           // STEP   4 :  Execute   a   query
            System .  Out .  println  (" creating   statement  ...");
            stmt  =  conn . createStat  ement ();
            String   sql  = " SELECT   method , min ( latency , 10
)  as   c , max ( latency , 10 )  from   sample - logstore   where
  __time__ >= 1500975424   and   __time__  <  1501035044   and
latency  >  0   and   latency  <  6142629   and   not  ( method ='
Postlogsto  relogs '  or   method =' GetLogtail  Config ')  group
by   method  " ;
            String   sql - example2  = " select   count ( 1 ) , max
( latency ), avg ( latency ),  histogram ( method ), histogram (
source ), histogram ( status ), histogram ( clientip ), histogram
( __source__ )  from   test10   where   __date__  >' 2017 - 07 - 20
  00 : 00 : 00 '  and   __date__  <' 2017 - 08 - 02   00 : 00 : 00
'  and   __line__ =' abc # def '  and   latency  <  100000   and  (
method  = ' getlogstor  elogS '  or   method =' Get **'  and   method
 <> ' GetCursorO  rData ' )";
            String   sql - example3  = " select   count ( 1 )  from
  sample - logstore   where   __date__  > ' 2017 - 08 - 07   00 : 00
: 00 '  and   __date__  < ' 2017 - 08 - 08   00 : 00 : 00 '  limit
100 ";
            ResultSet   rs  =  stmt . executeQue  ry ( sql );
           // STEP   5 :  Extract   data   from   result   set
            while ( rs . next ()){
               // Retrieve   by   column   name
                ResultSetM  etaData   data  =  rs . getMetaDat  a ();
                System . out . println ( data . getColumnC  ount ());
                for ( int   i  =  0 ; i  <  data . getColumnC  ount
();++ i ) {
```

```
                    String   name  =  data . getColumnN  ame ( i + 1
);
                    System . out . print ( name +":");
                    System . out . print ( rs . getObject ( name ));

                System . out . println ();

            Rs .  Close  ();
        }  catch  ( ClassNotFo  undExcepti  on   e ) {
            e . printStack  Trace ();
        }  catch  ( SQLExcepti  on   e ) {
            e . printStack  Trace ();
        }  catch  ( Exception   e ) {
            E .  printstack  trace  ();
        }  Finally  {
            if  ( stmt  ! =  null ) {
                try  {
                    Stmt .  Close  ();
                }  catch  ( SQLExcepti  on   e ) {
                    e . printStack  Trace ();


            if  ( conn  ! =  null ) {
                try  {
                    conn . close ();
                }  catch  ( SQLExcepti  on   e ) {
                    e . printStack  Trace ();
```

Tool call

In the classic network intranet or VPC environment, use the MySQL client to connect to Log Service.



Note:

1. **Enter your project name at ①.**
2. **Enter your Logstore name at ②.**

7 Query and visualization

7.1 Analysis graph

7.1.1 Graph description

Log Service provides a function similar to the SQL aggregate computing. All the SQL aggregate computing results can be rendered by using the visualized graphs provided by Log Service.

> Note:
> Before using the visualized graphs, read *Syntax description*.

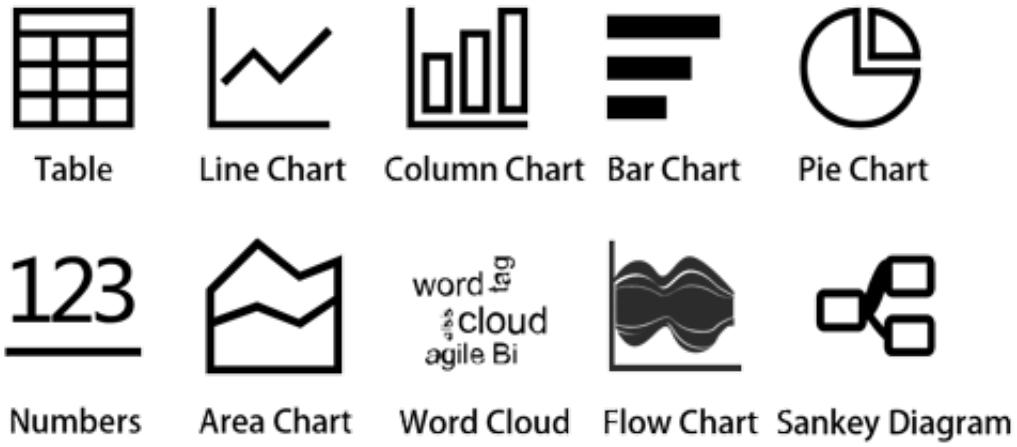
Prerequisites

1. Create an index and enable the analytics.
2. Log Service shows the graphs to you according to the statistical results only when you use the analysis statement for query.

Graph type

Currently, Log Service provides the following types of graphs.

Figure 7-1: Graph type



For how to use each type of graphs, see the following documents:

- *Table*
- *Line chart*

- *Column chart*
- *Bar chart*
- *Pie chart*
- *Single value chart*
- *Area chart*
- *Flow chart*
- *Sankey diagram*
- *Word cloud*

7.1.2 Table

Table, as the most common display type of data, is the most basic method to organize data. By organizing the data, table references and analyzes the data quickly. Log Service provides a function similar to the SQL aggregate computing. By default, the results obtained by using the query and analysis syntax are displayed in a table.

Basic components

- Header
- Row
- Column

Wherein:

- The number of `SELECT` items is the number of columns.
- The number of rows is determined by the number of logs after being computed in the current time interval. The default value is `LIMIT 100` .

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.
2. Click the Graph tab, the query results are displayed in a table  by default.

Example

The raw log is as follows.

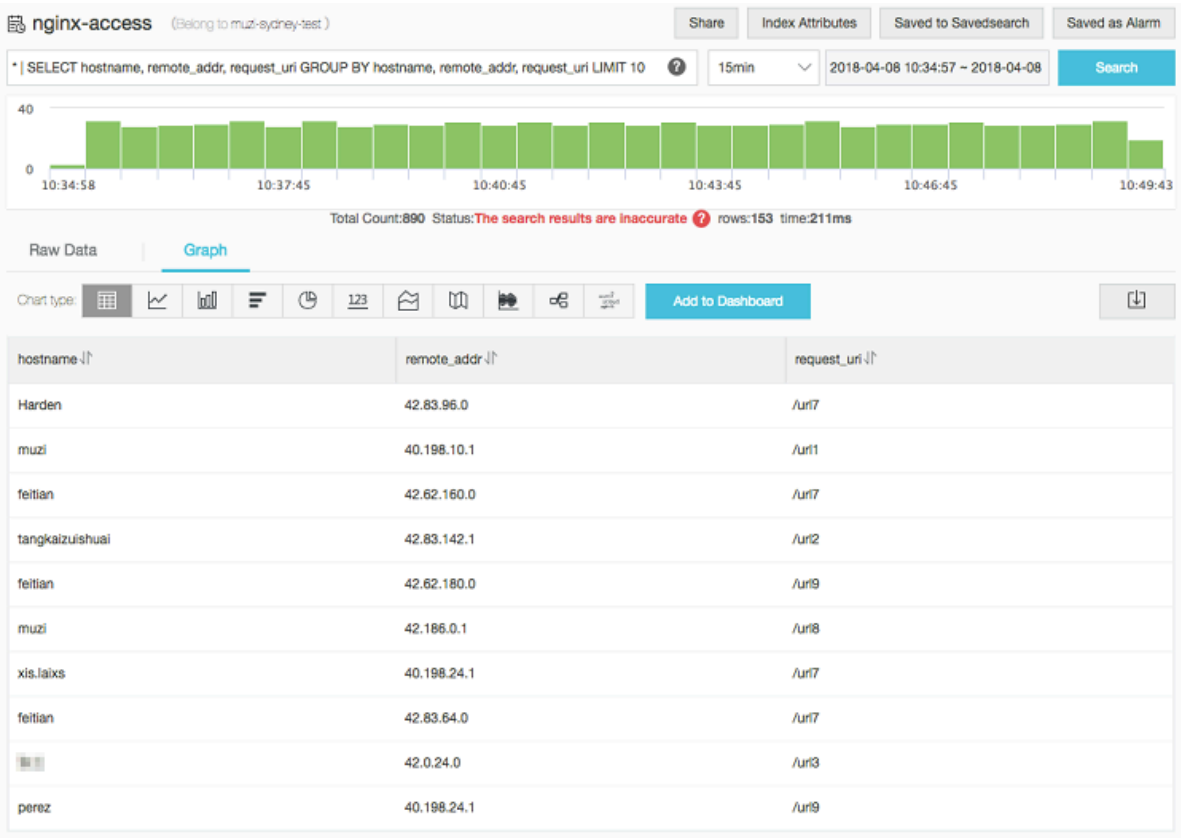
Figure 7-2: Original log



1. To obtain the columns hostname , remote_addr , and request_uri of the latest 10 logs, the statement is as follows:

```
* | SELECT hostname , remote_add r , request_ur i GROUP
 BY hostname , remote_add r , request_ur i LIMIT 10
```

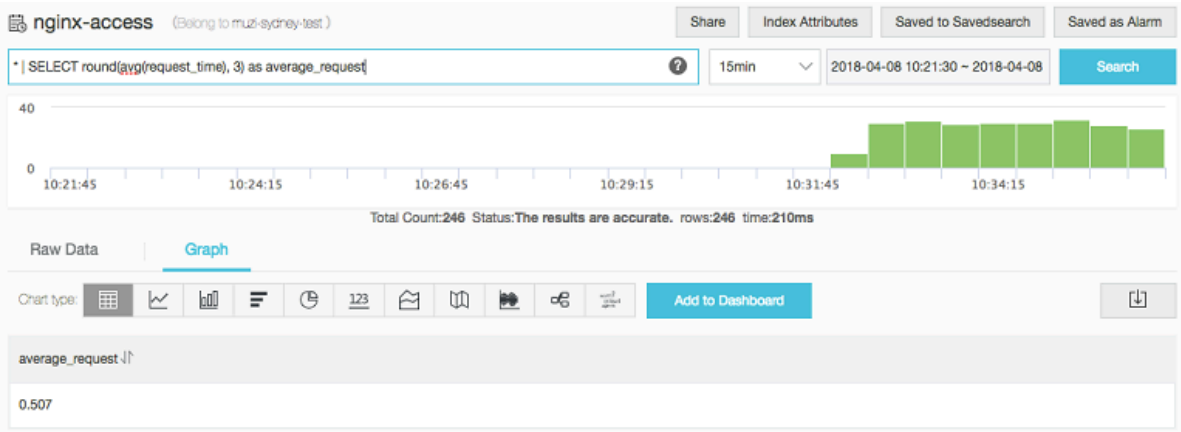
Figure 7-3: case 1



2. To compute a single data, for example, the average request_time (the average request time) in the current time interval, and retain three decimal places, the statement is as follows:

```
* | SELECT round ( avg ( request_ti me ), 3 ) as average_re
quest
```

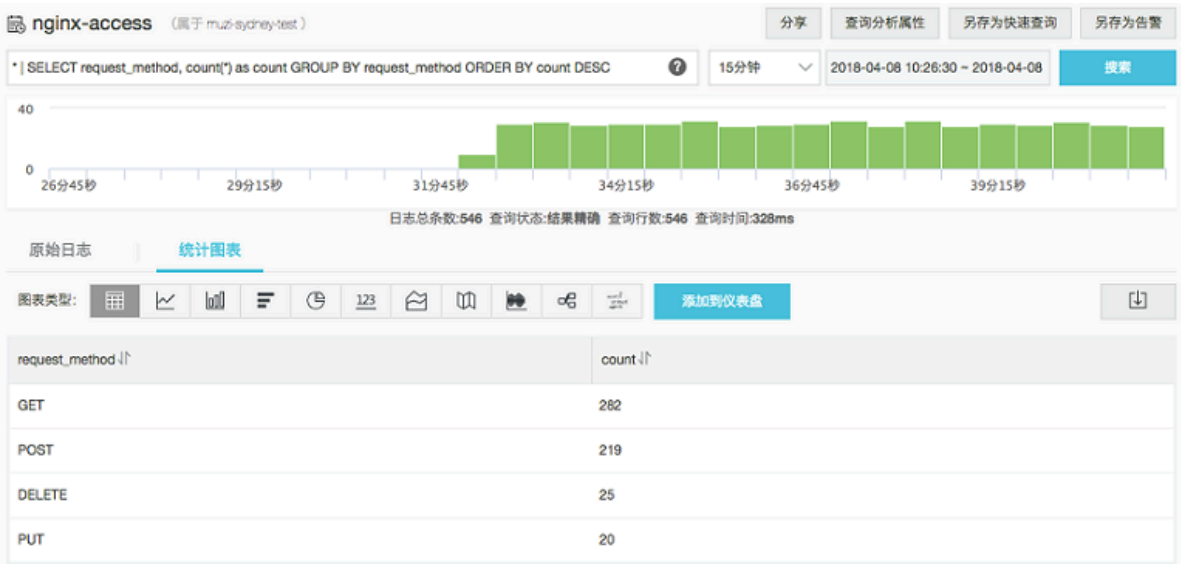
Figure 7-4: case 2



3. To compute grouped data, for example, the request_method distribution in the current time interval, and display the distribution in descending order, the statement is as follows:

```
* | SELECT request_me thod , count (*) as count GROUP BY
request_me thod ORDER BY count DESC
```

Figure 7-5: case 3



7.1.3 Line chart

The line chart, a graph for analyzing trend, is generally used to indicate the changes of a group of data on an ordered data type (successive time intervals in most cases) for analyzing the trend of data changes intuitively.

You can see the data changes in a period clearly by using the line chart. The changes are displayed mostly in the following aspects:

- Progressive increase or decrease
- Rate of increase or decrease
- Law of increase or decrease (such as periodic changes)
- Peak and valley

Therefore, the line chart is the best choice for analyzing the trend of data changes over time. You can also use multiple lines to analyze the changing trend of multiple groups of data in the same period, and then analyze the mutual effect (such as increasing or decreasing at the same time and being inversely proportional to each other) among data in different groups.

Basic components

- X axis
- Left Y axis
- Right Y axis (optional)
- Data point
- Changing trend line
- Legend

Configuration item

Configuration item	Meaning
X axis	Generally, the X axis is an ordered data type (time series).
Left Y axis	You can configure one or more columns of data to correspond to the value interval of the Y axis.

Configuration item	Meaning
Right Y axis	You can configure one or more columns of data to correspond to the value interval of the right Y axis (the layer of the right Y axis is higher than that of the left Y axis).
Column marker	Display a selected column in the left Y axis or right Y axis as a histogram.
Legend	The location where the legend is in the graph. You can configure the legend to the top, bottom, left, or right of the graph .
Padding	The distance between the coordinate axis and the graph boundary.

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.
2. Click the Graph tab and select the line chart  .
3. Configure the graph properties.

> Note:
> The number of data records for a single line must be greater than two in case that the data trend cannot be analyzed. We also recommend that you have no more than five lines in a line chart.

Example

Simple line chart

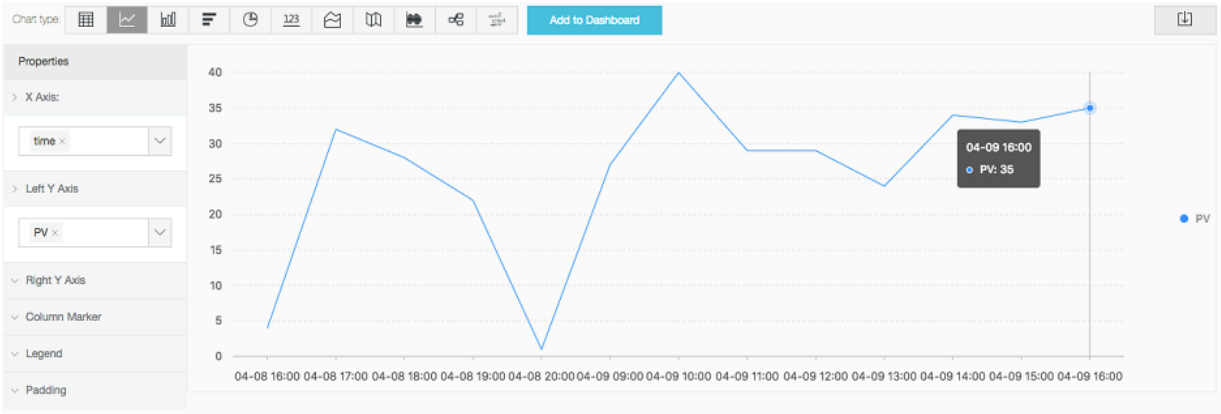
To query the access situation of the IP address `42 . 0 . 192 . 0` in the last day, the statement is as follows:

```
remote_add  r :  42 . 0 . 192 . 0  |  select   date_forma  t (
date_trunc (' hour ',  __time__ ), '% m -% d  % H :% i ')
```

```
as time , count ( 1 ) as PV group by time order by
time limit 1000
```

Select `time` as the X Axis, `PV` as the Left Y Axis, UV as the Right Y Axis, and PV as the Column Marker.

Figure 7-6: Simple line chart



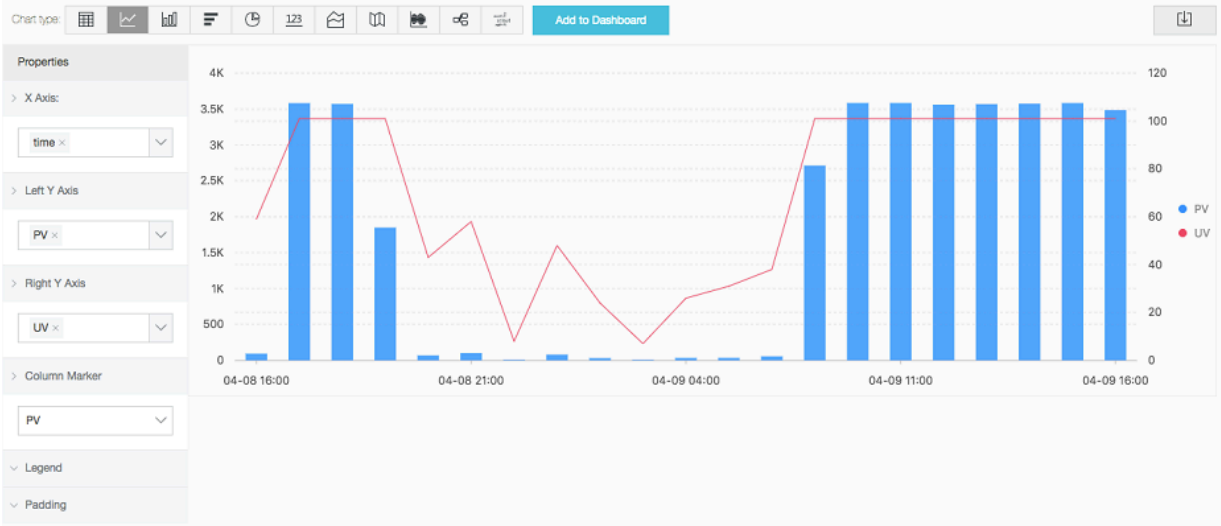
Line chart with both left Y axis and right Y axis

To query the access PVs and UVs in the last day, the statement is as follows:

```
* | select date_forma t ( date_trunc (' hour ', __time__ ), '% m
-% d % H :% i ') as time , count ( 1 ) as PV , approx_dis
tinct ( remote_add r ) as UV group by time order by
time limit 1000
```

Select `time` as the X Axis, `PV` as the Left Y Axis, `UV` as the Right Y Axis, and `PV` as the Column Marker.

Figure 7-7: Line chart with both left Y axis and right Y axis



7.1.4 Column chart

The column chart displays the numeric comparison among data types by using vertical or horizontal columns. The line chart describes the ordered data, while the column chart describes different types of data and counts the number in each data type.

You can also use multiple rectangular blocks to correspond to one type attribute in the grouping or stacked modes to analyze the differences of data types in different dimensions.

Basic components

- X axis (horizontal axis)
- Y axis (vertical axis)
- Rectangular block
- Legend

The column chart provided by Log Service uses the vertical columns by default, that is, the width of the rectangular block is fixed, and the height of the rectangular block indicates the numeric value. Use the grouped column chart to display the data if multiple columns of data are mapped to the Y axis.

Configuration items

Configuration items	Description
X axis	Generally, the X axis indicates the data types.
Y axis	You can configure one or more columns of data to correspond to the value interval of the Y axis.
Legend	The location where the legend is in the graph. You can configure the legend to the top, bottom, left, or right of the graph .
Padding	The distance between the coordinate axis and the graph boundary.

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.

2. Click the Graph tab and select the column chart (column).

3. Configure the graph properties.

> **Note:**
> Use the column chart if the number of data types is no more than 20. We recommend that you use `LIMIT` to control the number of data types in case that the horizontal width is so large that the analytical comparison is not intuitive. We also recommend that you have no more than five columns of data to map to the Y axis.

Example

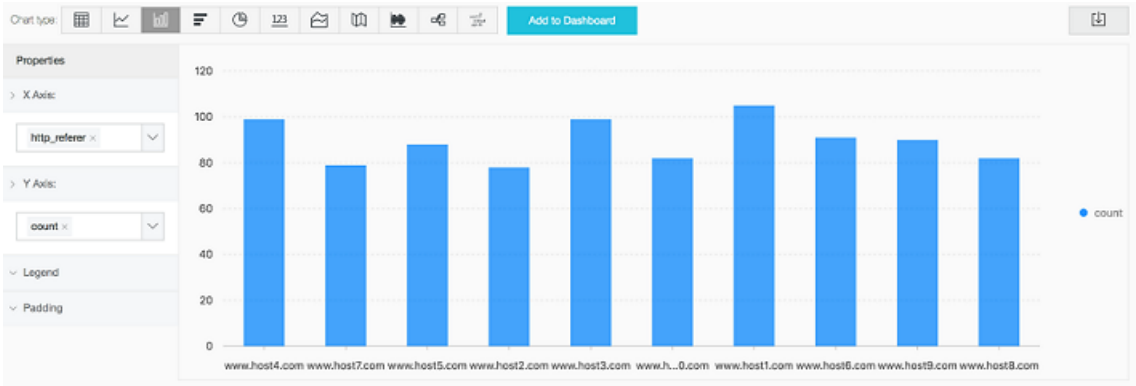
Simple column chart

To query the number of visits for each `http_refer er` in the current time interval, the statement is as follows:

```
* | select http_refer er , count ( 1 ) as count group by
 http_refer er
```

Select `http_refer er` as the X Axis and `count` as the Y Axis.

Figure 7-8: Simple column chart



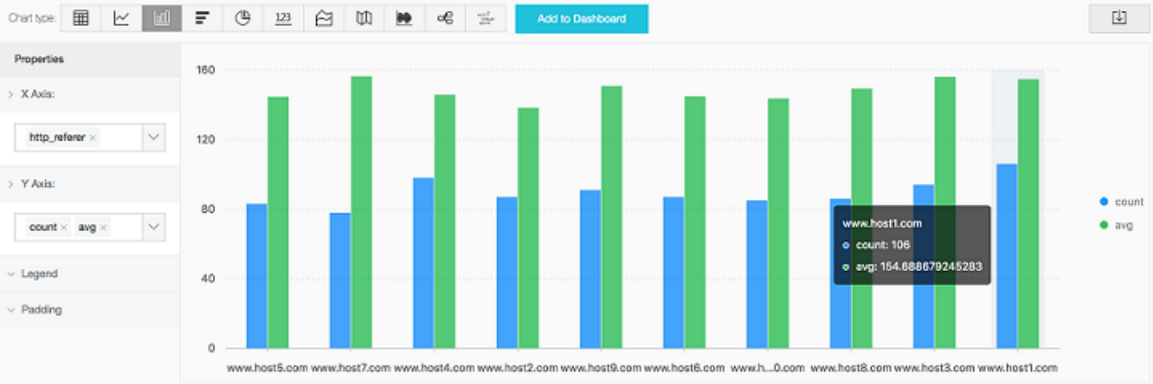
Grouped column chart

To query the number of visits and the average bytes for each `http_refer er` in the current time interval, the statement is as follows:

```
* | select http_refer er , count ( 1 ) as count , avg ( body_bytes _sent ) as avg group by http_refer er
```

Select `http_refer er` as the X Axis, and select `count` and `avg` as the Y Axis.

Figure 7-9: Grouped column chart



7.1.5 Bar chart

The bar chart is another form of column chart, that is, the horizontal column chart . Generally, the bar chart is used to analyze the top scenario and the configuration method is similar to that of the column chart.

Basic Components

- X axis (vertical axis)
- Y axis (horizontal axis)
- Rectangular block
- Legend

The height of the rectangular block in the bar chart is fixed and the width of the rectangular block indicates the numeric value. Use the grouped bar chart to display the data if multiple columns of data are mapped to the Y axis.

Configuration item

Table 7-1: Description

Description	Description
X axis	Generally, the X axis indicates the data types.
Y axis	You can configure one or more columns of data to correspond to the value interval of the Y axis.
Legend	The location where the legend is in the graph. You can configure the legend to the top, bottom, left, or right of the graph .
Padding	The distance between the coordinate axis and the graph boundary.

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.
2. Click the Graph tab and select the bar chart .
3. Configure the graph properties.

Note:

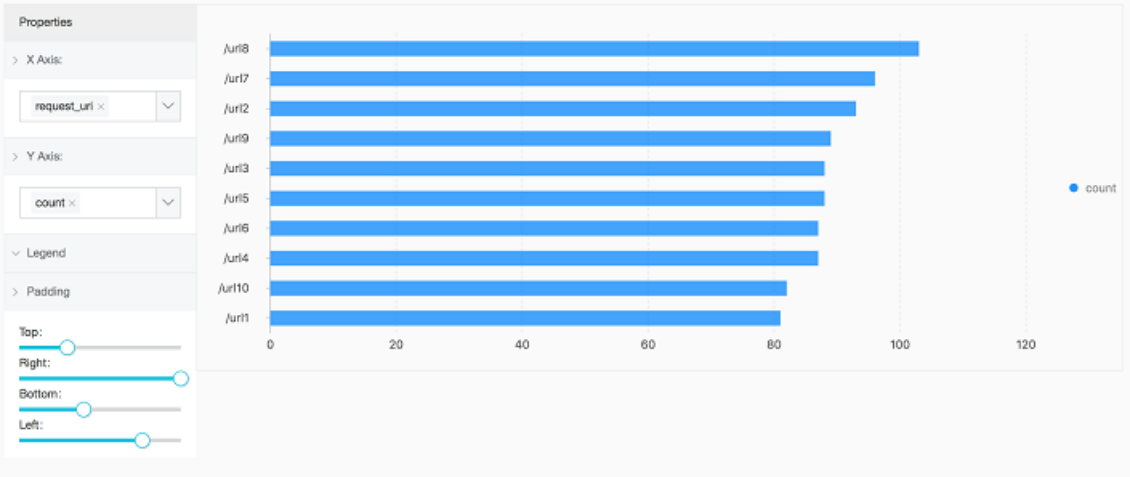
- Use the bar chart if the number of data types is no more than 20. We recommend that you use `LIMIT` to control the number of data types in case that the vertical height is so large that the analytical comparison is not intuitive, and use the `ORDER BY` syntax when analyzing the top scenario. We also recommend that you have no more than five columns of data to map to the Y axis.
- Supports grouped bar chart, but data in all groups of the bar chart must indicate the increase or decrease at the same time.

Simple bar chart

To analyze the top 10 request_uri with the largest number of visits, the statement is as follows:

```
* | select request_uri i , count ( 1 ) as count group by
 request_uri i order by count desc limit 10
```

Figure 7-10: Simple bar chart



7.1.6 Pie chart

The pie chart is used to indicate the ratios of different data types and compare different data types by using the radian. A pie is divided into multiple sections according to the ratios of different data types. The entire pie indicates the total amount of data, and each section (arc) indicates the ratio of a data type to the total amount of data. The sum of all the section (arc) ratios is 100%.

Basic components

- Sector
- Text percentage
- Legend

Configuration items

Configuration item	Description
Type	The data types.
Value column	The value corresponding to different types of data.

Configuration item	Description
Legend	The location where the legend is in the graph. You can configure the legend to the top, bottom, left, or right of the graph .
Padding	The distance between the coordinate axis and the graph boundary.
Pie chart type	Provides the pie chart (the default one), the cycle graph, and the Nightingale rose diagram.

Type

Log Service provides three types of pie charts: the default pie chart, the cycle graph, and the Nightingale rose diagram.

Cycle graph

Essentially, the cycle graph is a pie chart without the central part. Compared with the pie chart, the cycle graph has the following advantages:

- Supports displaying the total amount based on the original components, which provides you with more information.
- Comparing two pie charts is not intuitive. Two cycle graphs can be compared by using the ring length.

Nightingale rose diagram

Essentially, the Nightingale rose diagram is not a cycle graph, but a column chart in the polar coordinate system. The data types are divided by arcs and the radius of the arc indicates the data size. Compared with the pie chart, the Nightingale rose diagram has the following advantages:

- Use the pie chart if the number of data types is no more than 10, and use the Nightingale rose diagram if the number of data types is 11–30.
- The area is the square of radius. Therefore, the Nightingale rose diagram enlarges the differences among different types of data, and is especially applicable to comparing similar values.
- A circle has a period. Therefore, the Nightingale rose diagram can also be used to indicate the time concept in a period, such as the week or the month.

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.
2. Click the Graph tab and select the pie chart.
3. Instructions

> **Note:**
>
> - Use the pie chart and cycle graph if the number of data types is no more than 10. We recommend that you use LIMIT to control the number of data types in case that the number of sections with different colors is so large that the analysis is not intuitive.
> - We recommend that you use the Nightingale rose diagram or column chart if the number of data types is more than 10.

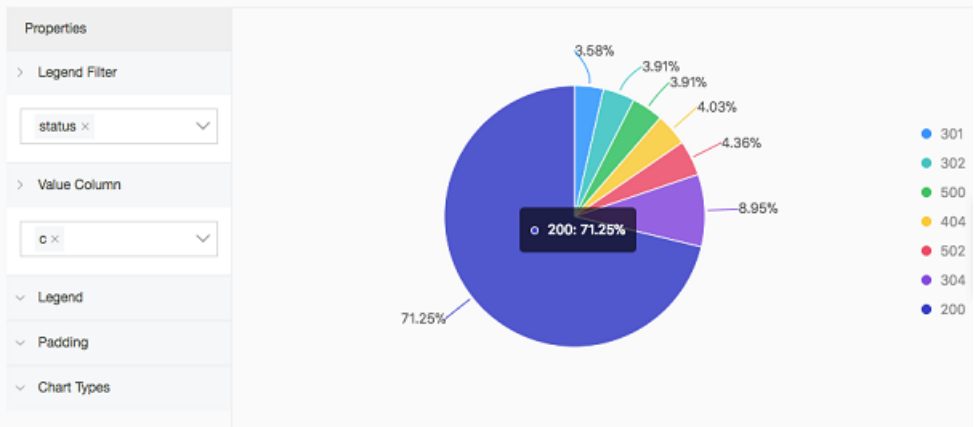
Example

Pie chart

Analyze the ratio of the access `status`:

```
* | select status , count ( 1 ) as c group by status
 order by c limit 10
```

Figure 7-11: Pie chart

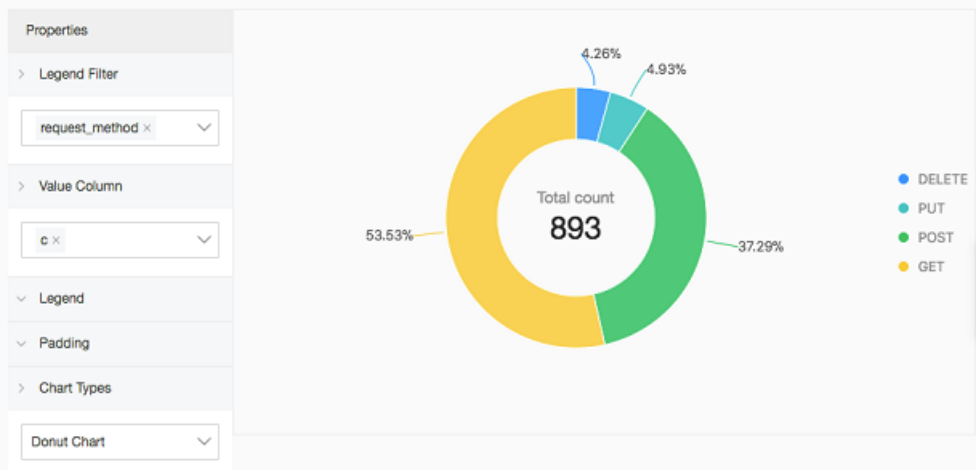


Cycle graph

Analyze the ratio of the access `request_me thod` :

```
* | select request_me thod , count ( 1 ) as c group by
 request_me thod order by c limit 10
```

Figure 7-12: Cycle graph

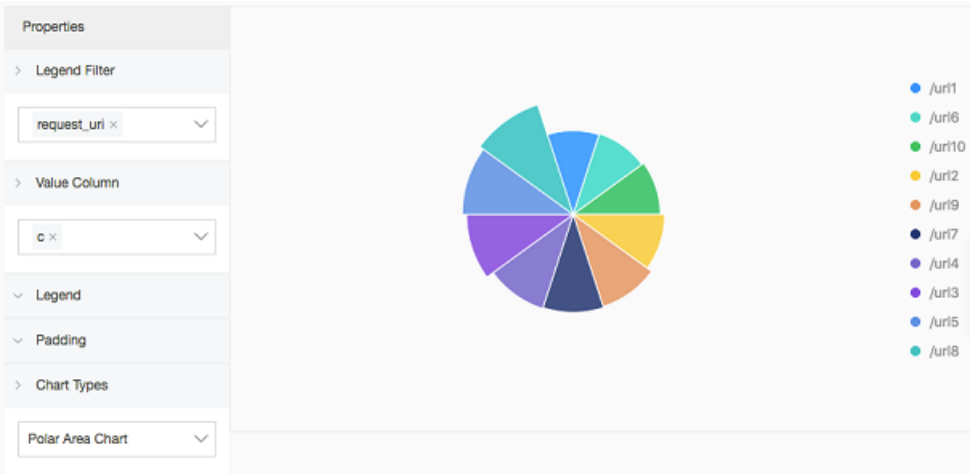


Nightingale rose diagram

Analyze the ratio of the access `request_ur i` :

```
* | select request_ur i , count ( 1 ) as c group by
 request_ur i order by c
```

Figure 7-13: Nightingale rose diagram



7.1.7 Area chart

The area chart is based on the line chart and has the section between the line and the coordinate axis in the line chart filled with color. The filled section is the area and the color highlights the trend better. The same as the line chart, the area chart emphasizes the number changes over time, and is used to highlight the trend of the total number. Both the line chart and the area chart are mostly used to indicate the trend and relationship, instead of the specific values.

Basic components

- X axis (horizontal axis)
- Y axis (vertical axis)
- Area block

Configuration items

Configuration item	Description
X axis	Generally, the X axis is an ordered data type (time series).
Y axis	You can configure one or more columns of data to correspond to the value interval of the Y axis.
Legend	The location where the legend is in the graph. You can configure the legend to the top, bottom, left, or right of the graph .
Padding	The distance between the coordinate axis and the graph boundary.

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.
2. Select the area chart ( ).
3. Configure the graph properties.

Note:

> The number of data records for a single area block in the area chart must be greater than two in case that the data trend cannot be analyzed. We also recommend that you have no more than five area blocks in an area chart.

Example

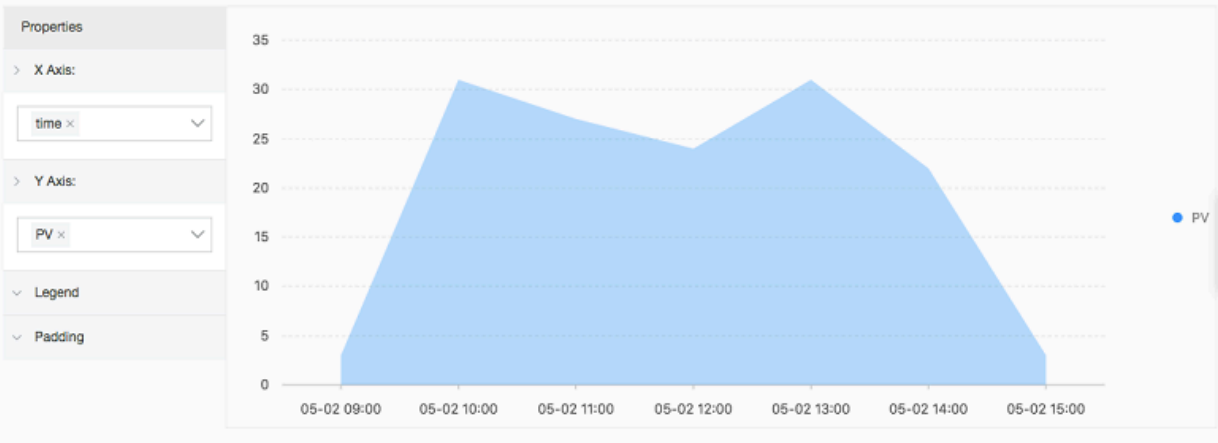
Simple area chart

The PV of IP `42 . 0 . 192 . 0` within the last day:

```
remote_add r : 42 . 0 . 192 . 0 | select date_forma t (
date_trunc (' hour ', __time__ ), '% m -% d % H :% i ') as time
, count ( 1 ) as PV group by time order by time
limit 1000
```

Select `time` as the X Axis and `PV` as the Y Axis.

Figure 7-14: Simple area chart



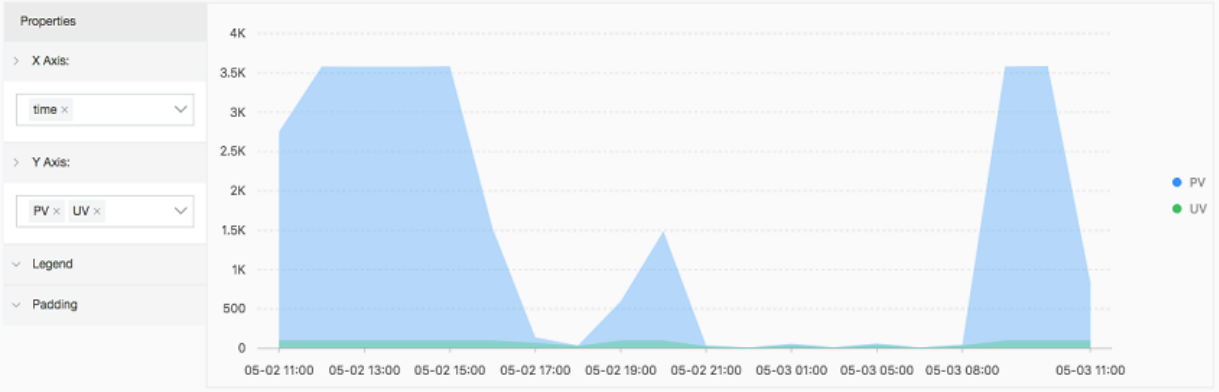
Stacked area chart

```
* | select date_forma t ( date_trunc (' hour ', __time__ ), '% m
-% d % H :% i ') as time , count ( 1 ) as PV , approx_dis
```

```
tinct ( remote_add r ) as UV group by time order by
time limit 1000
```

Select `time` as the X Axis. Select `PV` and `UV` as the Y Axis.

Figure 7-15: Stacked area chart



7.1.8 Single value chart

A single value chart highlights a single value. Types of single value charts includes:

- Rectangle frame: Shows general values.
- Dial: Shows how close the current value is to the configured threshold.
- Compare numb chart: Shows SQL query results of year-on-year function and period-over-period function. For information about analysis syntax, see *Interval-valued comparison and periodicity-valued comparison functions*.

A rectangle frame is displayed by default. A rectangle frame is the simplest and most direct data representation, which visually and clearly displays the data at a certain point. It is generally used to represent the key information at a certain point in time. To display the proportional type indicator, you can use a dial.

Basic components

- Text
- Unit (optional)
- Description (optional)
- Chart types

Configuration

- Rectangle frame configuration instructions:

Rectangle frame configuration	Description
Chart type	Rectangle frame
Value column	By default, the first line of data in this column is displayed.
Text	The attribute configurations related to the text, including: - Font size (12 px –100 px) - Unit - Unit font size (12 px–100 px) - Description - Description font size (12 px–100 px)
Color	Colors in the diagram, including: - Font color - Background color

- Dial configuration instructions:

Dial configuration category	Configuration	Description
Chart type	Dial	Displays query results in a dial.
Value column	Actual value	By default, the first line of data in this column is displayed.
	Unit	The unit of values in the dial.
	Dial maximum	The maximum value displayed on the dial. The default is 100.
	Colored regions	The dial is divided into several value regions. Each region is displayed in a different color. The maximum number of colored regions is 5. The default is 3.

Dial configuration category	Configuration	Description
	Region max value	The maximum value of each region on the dial. By default, the maximum value of the last region is the maximum value on the dial and you do not need to specify this value.  Note: By default, three colored regions divide the dial evenly. Changing the number of colored regions does not change the value range of each default colored region. Therefore, set the maximum value for each colored region based on your needs.
	Show title	You can add a single value chart of the dial type in the dashboard. By using Show Title, you can display or hide the title of a single value chart in the dial form on the dashboard page. The item is disabled by default, that is, the dial title is not displayed. Clicking the enable button does not display the title on the current page. The title is displayed on the dashboard page after you create a report or modify the current report.
Text	Font size	The font size of the text, in the range of 12 px–100 px.
	Description	The value description.
	Description font size	The font size of description content, in the range of 12 px–100 px.

Dial configuration category	Configuration	Description
Color	Region color	By default, the dial has three regions which are in colors of blue, yellow, and red respectively. If you change the Colored Regions to a value greater than 3, the added regions are blue by default. You can change the color of each region.
	Font color	Colors of values displayed in the dial.

- Compare numb chart configuration instructions

Configuration category	Configuration	Description
Chart types	`Compare numb chart`	Displays query results in a compare numb chart
Value column	`Show value`	A value displayed in the center of the compare numb chart. This value is generally set to the statistical result of the current time period in the compare numb function
	`Compare value`	A value used to compare with the threshold. This value is typically set to the comparison result between the current time period and the previous time period in the compare numb chart.
	`Trend comparison threshold`	A value used to measure the variation trend of a compare value.
Text	`Font size`	The font size of the show value, in the range of 12 px–100 px.
	Unit	The unit of the show value.
	Unit font size	The font size of the unit of the show value, in the range of 12 px–100 px.
	Compare unit	The unit of the compare value.

Configuration category	Configuration	Description
	Compare font size	The font size of the compare value and its unit, in the range of 12 px–100 px.
	`Descriptio n`	A description of the displayed value and its growth trends, which is displayed below the value.
	Description font size	The font size of the value description, in the range of 12 px–100 px.
Color	`Font color`	Font color of the show value
	Growth font color	The font color of the compare value that is greater than the threshold.
	Growth background color	The background color displayed when the compare value is greater than the threshold.
	Decrease font color	The font color of the compare value that is less than the threshold.
	Decrease background color	The background color displayed when the compare value is less than the threshold.
	Equal background color	The background color displayed when the compare value is equal to the threshold.

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and click Search.
2. Select the single value chart 123.
3. Select a chart type based on your needs, and configure chart properties.

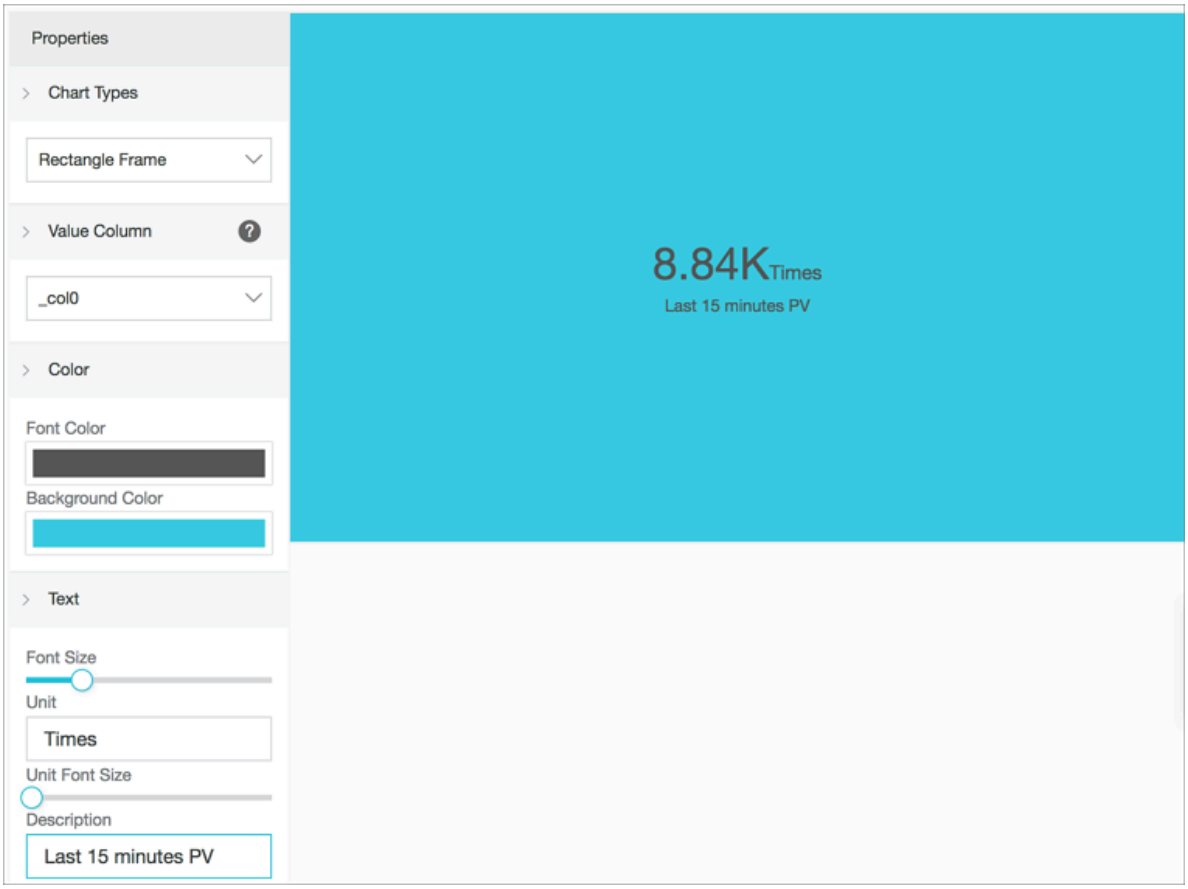
> Note:
> A Log Service single value chart automatically performs normalization based on the numerical size. For example, `230000` is processed as `230K`. To define your own numeric format, please define the format in real-time analysis through *Mathematical calculation functions*.

Examples

Execute the following query analysis statement to view the number of visits and display analysis results in diagrams:

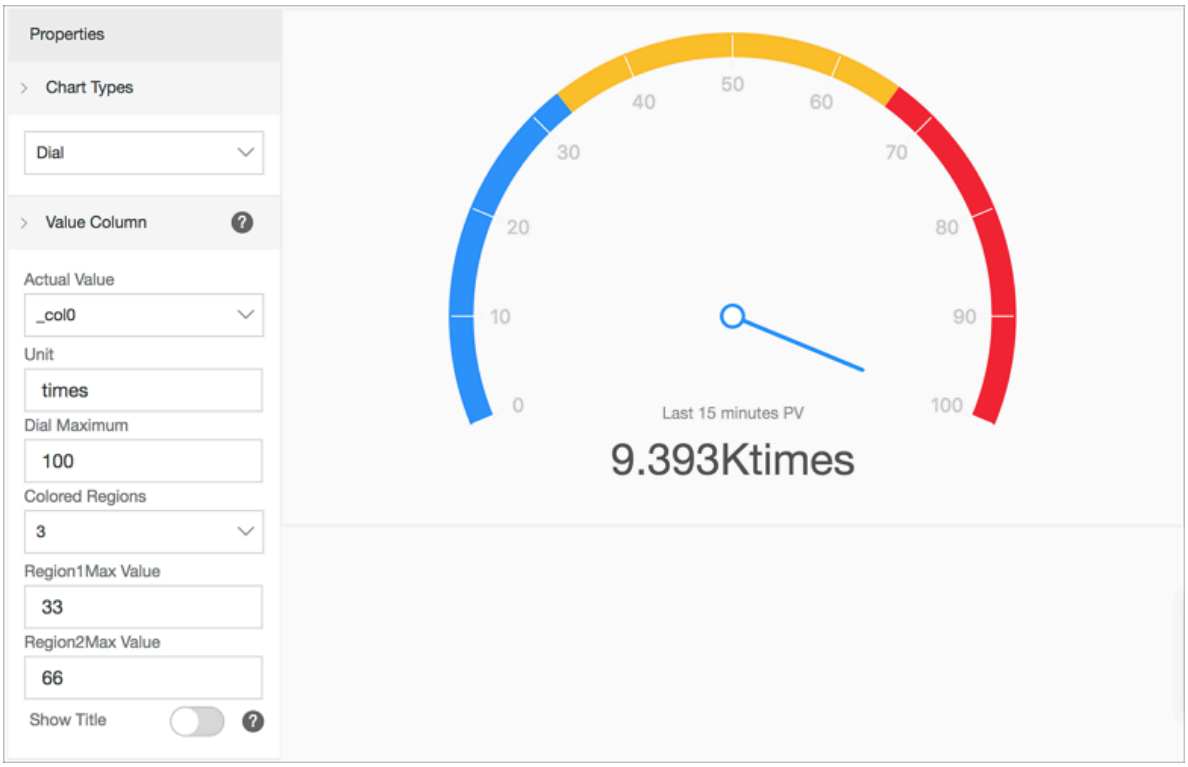
- Rectangle frame

```
* |  select   count ( 1 )  as   pv
```



- **Dial**

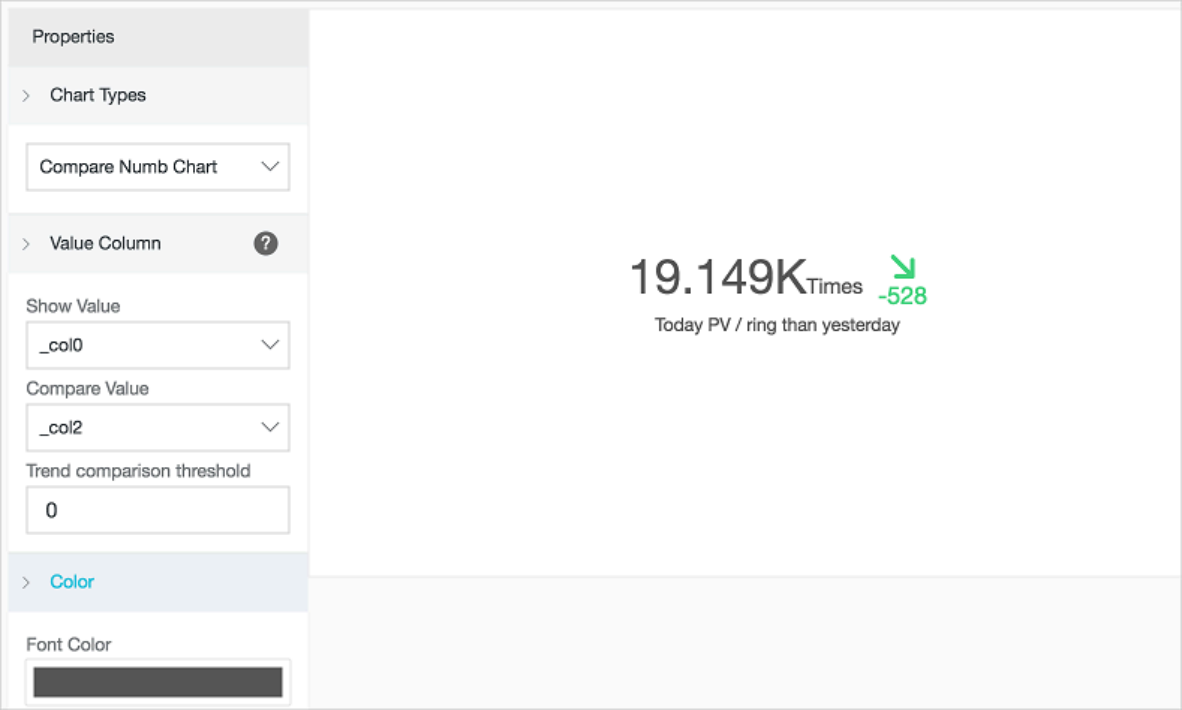
```
* | select count ( 1 ) as pv
```



- Compare numb chart

  View the comparison of today visits and yesterday visits:

```
* |  select   diff [ 1 ], diff [ 2 ],  diff [ 1 ]- diff [ 2 ]  from
  ( select   compare (  pv  ,  86400 )  as   diff   from  ( select
 count ( 1 )  as   pv   from   log ))
```



7.1.9 Flow chart

The flow chart, also known as ThemeRiver, is a stacked area chart around the central axis. The banded branches with different colors indicate different types of information. The band width indicates the corresponding numeric value. Besides, the centralized time attribute of the original data maps to the X axis, which forms a three-dimensional relationship.

You can switch a flow chart to a line chart or column chart. Note that the column chart is displayed in the stacked form by default, and the start point of each data type is at the top of the last column.

Basic components

- X axis (horizontal axis)
- Y axis (vertical axis)
- Band

Configuration item

Configuration item	Description
X axis	Generally, the X axis is an ordered data type (time series).
Y axis	You can configure one or more columns of data to correspond to the value interval of the Y axis.
Aggregate column	The information requires to be aggregated in the third dimension.
Legend	The location where the legend is in the graph. You can configure the legend to the top, bottom, left, or right of the graph .
Padding	The distance between the coordinate axis and the graph boundary.
Chart type	Provides the area chart (the default one), line chart, and column chart (stacked).

Procedure

1. On the query page, enter the query statement in the search box, select the time interval, and then click Search.
2. Click the Graph tab and select the flow chart  .
3. Configure the graph properties.

Example

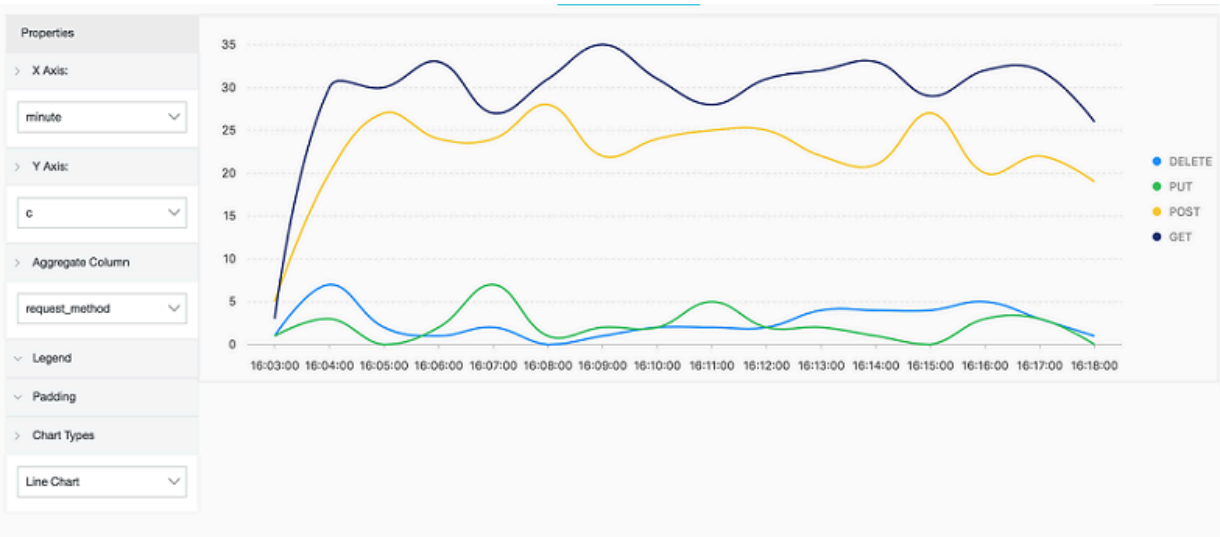
The flow chart is applicable to displaying the three-dimensional relationship (time-type-value).

```
* |  select   date_forma  t ( from_unixt  ime ( __time__  -  __time__
 %  60 ), '% H :% i :% S ')  as   minute ,  count ( 1 )  as   c ,
```

```
request_me thod group by minute , request_me thod order
by minute asc limit 100000
```

Select `minute` as the X Axis, `c` as the Y Axis, and `request_me thod` as the Aggregate Column.

Figure 7-16: Flow chart



7.1.10 Tree map

A tree map is a rectangle chart that contains rectangle blocks in a tree structure. The area of each rectangle block in a tree map is proportional to the amount of data it represents. The larger the area is, the greater the proportion of data it represents.

Components

Rectangle blocks are generated from data calculations and distributed in the chart.

Configuration

Configuration	Description
Legend filter	Field that indicates a data type.
Value column	Value field. The greater the value of a data type, the larger the corresponding rectangle block will be.
Padding	The spacing between any two adjacent sides of different rectangle blocks. The value range of this field is 0–100 px.

Procedure

1. Enter a query statement, select a time interval, and then click Search & Analysis.

2. Select the tree map  .

3. Configure the chart properties.

Example

Analyze the distribution of the hostnames in the Nginx logs.

```
* | select hostname , count ( 1 ) as count group by
 hostname order by count desc limit 1000
```

Select `hostname` from the Legend Filter drop-down list and select `count` from the Value Column drop-down list.



7.2 Dashboard

7.2.1 Dashboard overview

This topic describes the functions of charts and dashboards provided by Log Service for real-time data analysis.

A chart provides the same functionality as a query and analysis statement. One or multiple charts can be added to and subsequently viewed from a dashboard. When you open or refresh a dashboard, each chart in the dashboard automatically runs the query and analysis statement that it represents.

Log Service also provides the *Console embed* function that allows you to embed a dashboard into an external web page and choose to view the dashboard from either the Log Service console, or the newly embedded web page. Additionally, Log Service

also provides the *Drill-down analysis* function. With this function, you can set drill-down analysis configurations for a chart when you add the chart to a dashboard, and then use the function to obtain more detailed analysis results.

Limits

- Up to 50 dashboards can be created for each project.
- Each dashboard can contain up to 50 analysis charts.

Try out the dashboard function

Account: sls-reader1

Password: pnX-32m-MHH-xbm

Open the trial dashboard console.

Functions

A dashboard can be either in display mode or editing mode.

- ***Display mode***

  When you select display mode for a dashboard, you can set the parameters for displaying the analysis results as follows:

  - Settings for displaying a dashboard.

    You can set the global query time range for the dashboard, set alert notifications for all charts, set whether to automatically refresh the dashboard page, choose to expand the dashboard to full screen display, and set the display style of all chart titles.
  - Settings for displaying a chart.

    You can view analysis results of the chart, set the query time range for the chart, set alert notifications for the chart, download the chart and the chart log, and check whether *drill-down analysis* is set.

- ***Editing mode***

  When you select editing mode for a dashboard, you can set multiple dashboard parameters as follows:

  - Set a dashboard.
    - Add elements to the dashboard. For example, you can add *markdown charts*, customized charts, texts, icons, and other elements to the dashboard.
    - Add a line to connect two charts. Lines can automatically be adjusted according to chart location.
    - Add a *filter*. Note that in editing mode you can add a filter, but you need to switch to display mode to filter the chart data.
    - Show or hide grid lines.
  - Set a chart.

    You can modify the query statement represented by a chart and the chart properties, or *set a drill-down analysis*.

7.2.2 Create and delete a dashboard

This topic describes how to create and delete a dashboard.

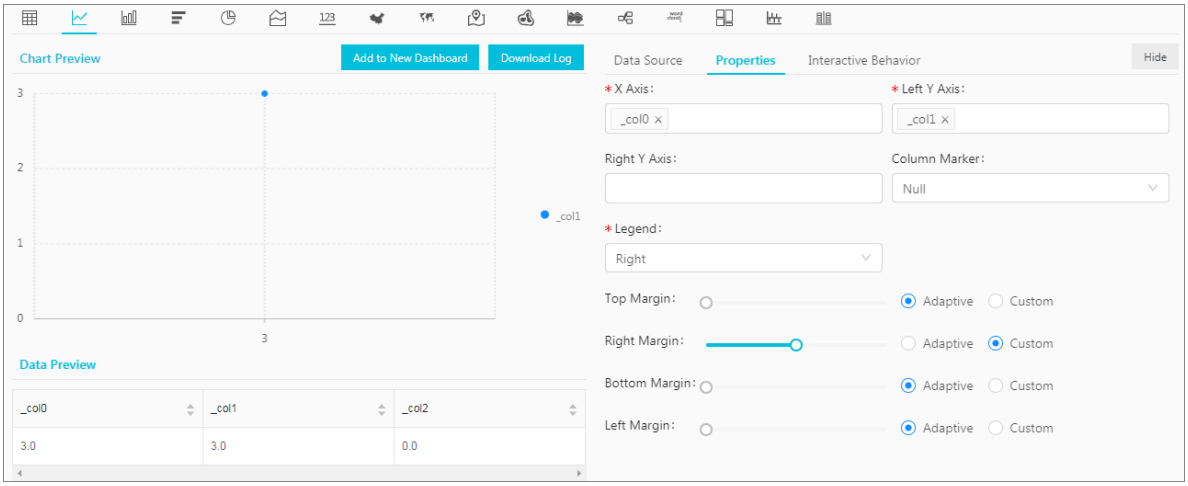
Prerequisites

- Logs are collected.
- *Indexes* are enabled and set.

Create a dashboard

1. Log on to the *Log Service console*, and then click the target project name.
2. On the Logstores page, click Search in the LogSearch column.
3. In the search box, enter a query and analysis statement, and click Search & Analysis.

4. Click Properties to set the chart properties.



5. Optional. Specify a placeholder variable.

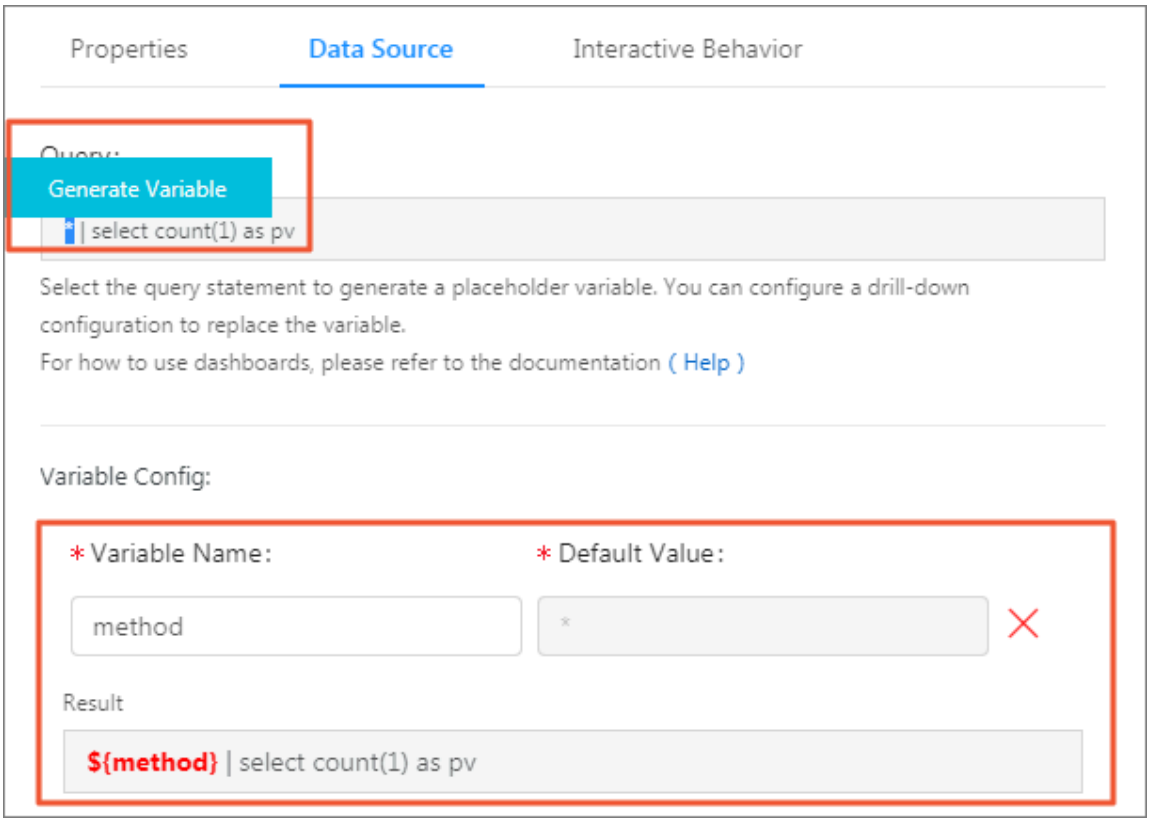
> **Note:**
>
> If you specify a placeholder variable for the query statement, and a drill-down event hat results in a jump to the current dashboard is triggered by any other chart, the system performs the following:
>
> a. It replaces the placeholder variable of the query statement with the chart value that triggered the drill-down event.
> b. It refreshes the current dashboard and updates with the new query statement.
>
> For more information, see *Set a drill-down analysis*.

To specify a placeholder variable, follow these steps:

a. Click the Data Source tab, and select part of the query statement.
b. Click Generate Variable.
c. Set Variable Config.

Configuration	Description
Variable Name	Placeholder variable name. If the placeholder variable name is the same as the variable specified in the chart that triggered the drill-down event, the placeholder variable will be replaced with the chart value.
Default Value	Default value of the placeholder variable in the current dashboard.

Configuration	Description
Result	Query statement that has the specified variable.



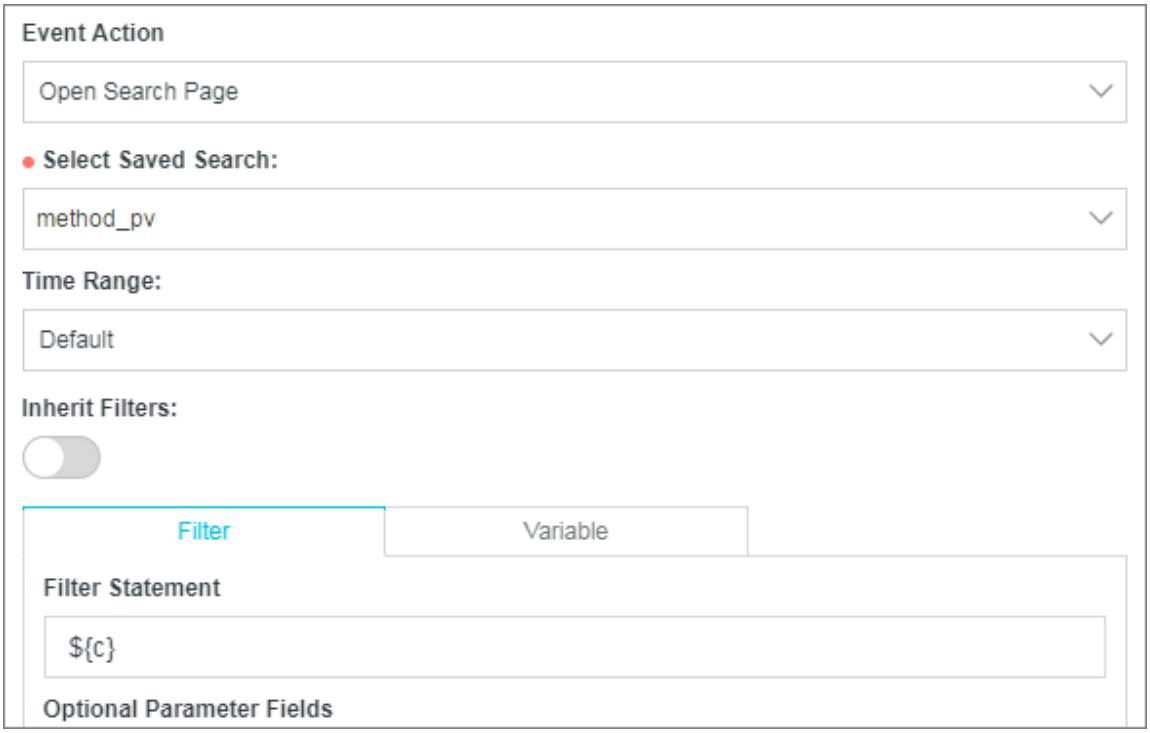
6. Optional. Set *a drill-down analysis*.

   To set a drill-down analysis, follow these steps:

   a. Click the Interactive Behavior tab.

   b. Select an Event Action.

   c. Set parameters related to the selected event action.



7. Click Add to New Dashboard.

8. Set the dashboard and chart names.

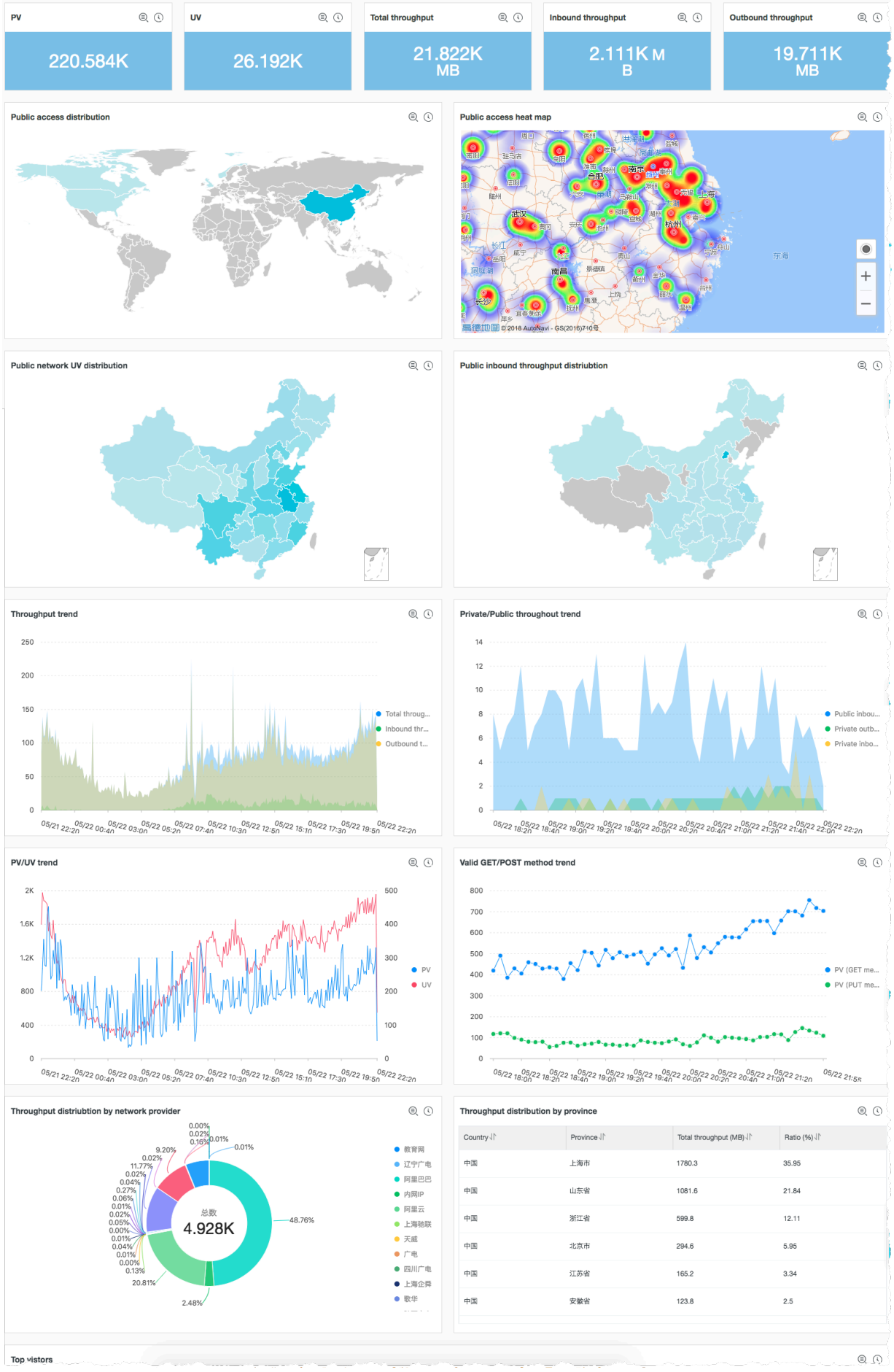
Configuration	Description
Operation	Available actions are: · Add to Existing Dashboard: add the chart to an existing dashboard. · Create Dashboard: create a new dashboard and then add the chart to the new dashboard.
Dashboards	Select an existing dashboard name. Note: This parameter is required only when you set the Operation parameter as Add to Existing Dashboard.

Configuration	Description
Dashboard Name	Enter a dashboard name. Note: This parameter is required only when you set the Operation parameter as Create Dashboard.
Chart Name	Enter a chart name. The chart name is displayed as the chart title in the dashboard.

9. Click OK.

   You can add up to fifty analysis charts to a dashboard.

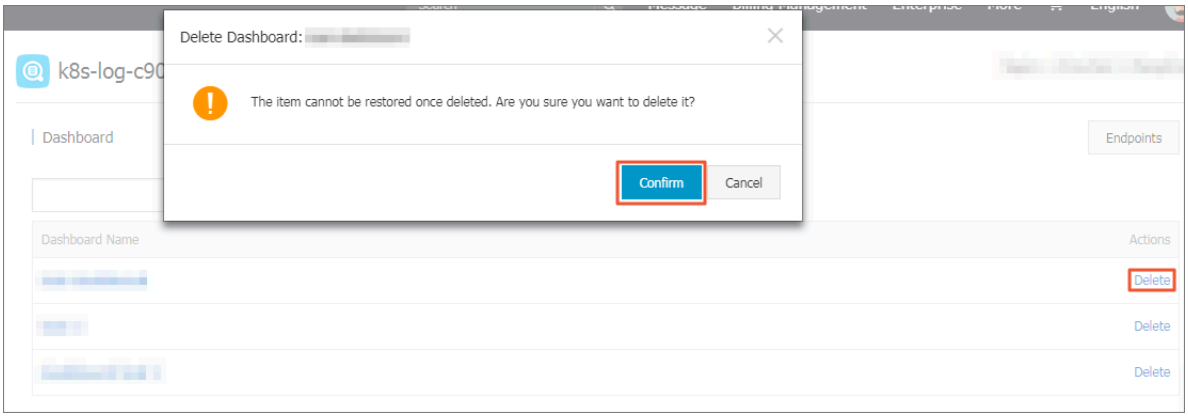
The following figure shows a dashboard that contains multiple analysis charts.


PV
220.584K
UV
26.192K
Total throughput
21.822K MB
Inbound throughput
2.111K MB
Outbound throughput
19.711K MB
Public access distribution
Public access heat map
Public network UV distribution
Public inbound throughput distriubtion
Throughput trend
Total throug...
Inbound thr...
Outbound t...
Private/Public throughout trend
Public inbou...
Private outb...
Private inbo...
PV/UV trend
PV
UV
Valid GET/POST method trend
PV (GET me...
PV (PUT me...
Throughput distriubtion by network provider
总数
4.928K
48.76%
20.81%
2.48%
Throughput distribution by province
Country
Province
Total throughput (MB)
Ratio (%)
中国 上海市 1780.3 35.95
中国 山东省 1081.6 21.84
中国 浙江省 599.8 12.11
中国 北京市 294.6 5.95
中国 江苏省 165.2 3.34
中国 安徽省 123.8 2.5
Top visitors

Delete a dashboard

> **Note:**
> A deleted dashboard cannot be recovered.

1. In the left-side navigation pane of the Logstores page, click Dashboard.
2. Click Delete on the right of the target dashboard.
3. In the displayed dialog box, click Confirm.



7.2.3 Set the display parameters for a dashboard

This topic describes how to set the display parameters for a dashboard in Log Service. The display mode is the default viewing mode of a dashboard.

To open your target dashboard, perform either of the following two operations:

- In the left-side navigation pane of the Logstores page, click Dashboard, and then click the target dashboard name.
- In the left-side collapsed navigation pane of the Search & Analysis page, the Saved Search page, or any other page in the Log Service console, move your pointer over the pane to show the items, and then click the target dashboard name.

Available settings

- Settings for displaying a dashboard

  When display mode is selected for a dashboard, the following available function buttons are located in the upper-right corner of the dashboard (left to right): Please select, Edit, Subscribe, Alerts, Refresh, Share, Full Screen, Title Configuration, and Reset Time.

- Settings for displaying a chart

  When display mode is selected for a dashboard, the list hidden in the upper-right corner of a chart provides functions for you to set parameters related to chart analysis results.

  > Note:
  > Different elements in a dashboard have different function lists.

Set the query time range for a dashboard

All charts in a dashboard use the query time range that is set for the dashboard. To set a query time range exclusive to a single chart, follow the instructions described in *Set the query time range for a chart*.

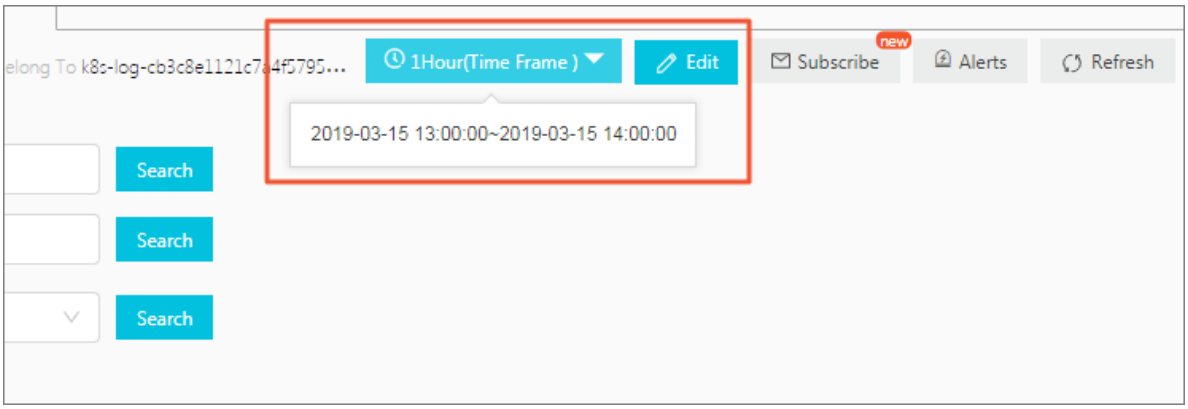
> Note:
> A custom set query time range for a dashboard is a temporary setting that is not saved by the system. This means that when you re-open the dashboard to view charts, the system displays the query and analysis results of the default query time range.

1. Click Please Select.
2. Select a query time range.

   Available types of query time ranges for a dashboard are as follows:

   - Relative: indicates to query the logs in an exact time range (accurate to seconds) of one minute, five minutes, fifteen minutes, or other time length that starts from the current time point. For example, if the current time point is 19:20:31 and you set this parameter to one hour, then the charts in the dashboard query logs generated from 18:30:31 to 19:20:31.
   - Time Frame: indicates to query the logs in a whole time range at a one-minute, five-minute, fifteen-minute period, or any other time length period that starts from the current time point. For example, if the current time point is 19:20:31 and you set this parameter to one hour, then the charts in the dashboard query logs generated from 18:00:00 to 19:00:00.
   - Custom: indicates to query the logs in a customized time range.

3. Move your cursor to Please Select to verify that the time range that you set has taken effect.



Switch to editing mode

To enter editing mode, click Edit. For more information, see *Edit a dashboard*.

Set an alert notification

To create or modify an alert notification, choose Alerts > Create or choose Alerts > Modify in the upper-right corner. An alert must be associated with at least one chart.

For more information, see *Set an alert*.

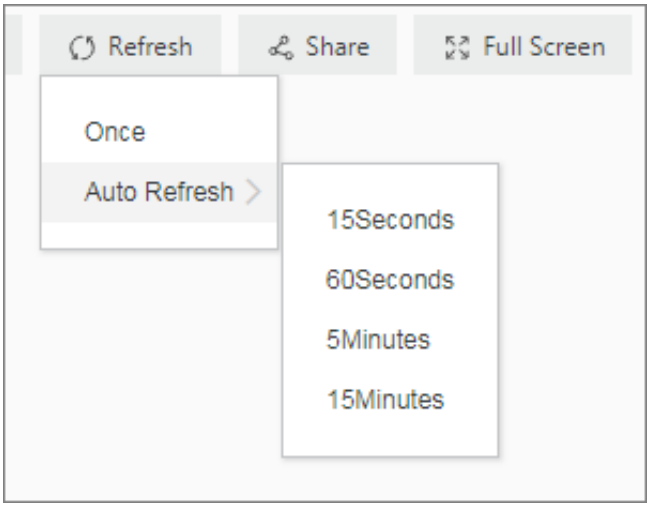
Set the refresh page frequency

You can manually refresh a dashboard, or set a time interval at which the dashboard automatically refreshes.

- To manually refresh the dashboard, choose Refresh > Once.
- To set the dashboard to refresh automatically at specified time intervals, choose Refresh > Auto Refresh. Then, select an interval.

  A dashboard can be automatically refreshed at 15-second intervals, 60-second intervals, 5-minute intervals, or 15-minute intervals.

> Note:
> If no activity is detected within your browser and the screen goes to sleep, the time interval at which the dashboard refreshes may be affected.



Share a dashboard

To share a dashboard with other users, click Share to copy the link of the dashboard page and then send the link to users that have the permission to view the dashboard. Your current dashboard settings are saved in the shared dashboard page (such as the query time range and the style to display chart titles).

> **Note:**
> Before you share a dashboard, you must grant the target user or users the permission to view the dashboard.

Display a dashboard in full screen

Click Full Screen. We recommend that you perform this operation if you want to focus on data or make a presentation.

Set the chart title format

Click Title Configuration, and then select one of the following formats:

- Single-line Title and Time Display
- Scrolling Title and Time Display
- Alternate Title and Time Display
- Title Only
- Time Only

Reset the query time range

To restore the default query time ranges of all the charts in a dashboard, click Reset Time.

Select chart view

- View analysis details of a chart

  To view analysis details of a chart (such as the query statement associated with a chart and the chart properties) , choose More Option > View Analysis Details in the upper-right corner of the chart.
- Set the query time range for a chart

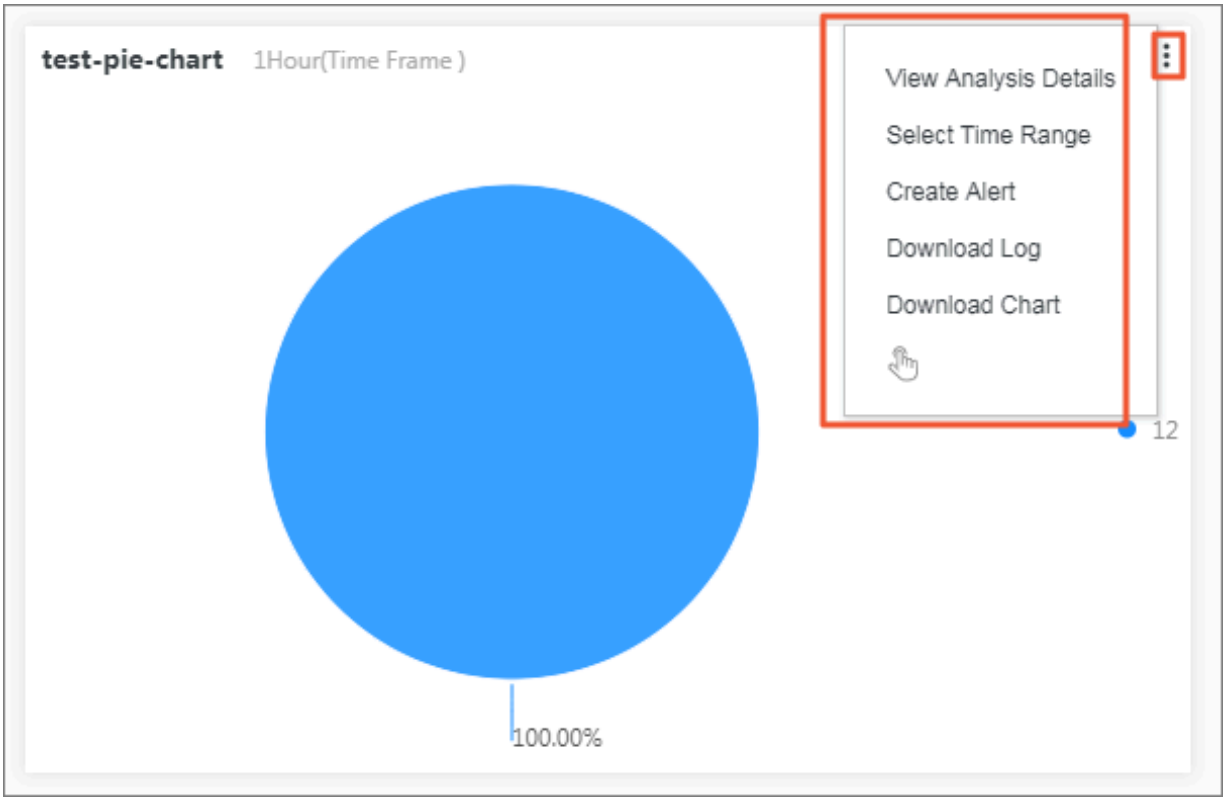
  To set the query time range for a chart, choose More Option > Select Time Range in the upper-right corner of the chart. The setting only takes effect for the current chart.
- Set an alert notification for a chart

  To set an alert notification for a chart, choose More Option > Create Alert in the upper-right corner of the chart. For more information, see *Set an alert*.
- Download logs

  To download logs, choose More Option > Download Log in the upper-right corner of the chart. The raw log analysis results are downloaded as a .csv file.
- Download a chart

  To download a chart, choose More Option > Download Chart in the upper-right corner of the chart. The chart is downloaded as a .psd file.
- Check whether a drill-down analysis is set for a chart

  To check whether drill-down analysis is set for a chart, move your cursor to the More Option button in the upper-right corner of the chart, and check the color of the icon at the bottom of the hidden list. A red icon indicates that a drill-down analysis is set for the chart. A gray icon indicates that no drill-down analysis is set for the chart.



7.2.4 Edit a dashboard

This topic describes how to edit a dashboard.

You can perform the following editing operations in a dashboard:

- Set dashboard parameters.
  - In the upper-left corner, you can click the current dashboard name to modify the dashboard name.
  - Add elements to the dashboard. For example, you can add *markdown charts*, customized charts, texts, icons, and other elements as needed.
  - Add a line to connect two charts. Lines are automatically adjusted according to the chart location.
  - Add a *filter*. A filter can filter specific chart data when the dashboard is in the displaying mode.
  - Set the dashboard to show grid lines.
  - Set chart properties by using the tools in the menu bar.
- Set a chart.

  You can modify the query statement represented by a chart and the chart properties, or *set a drill-down analysis*.

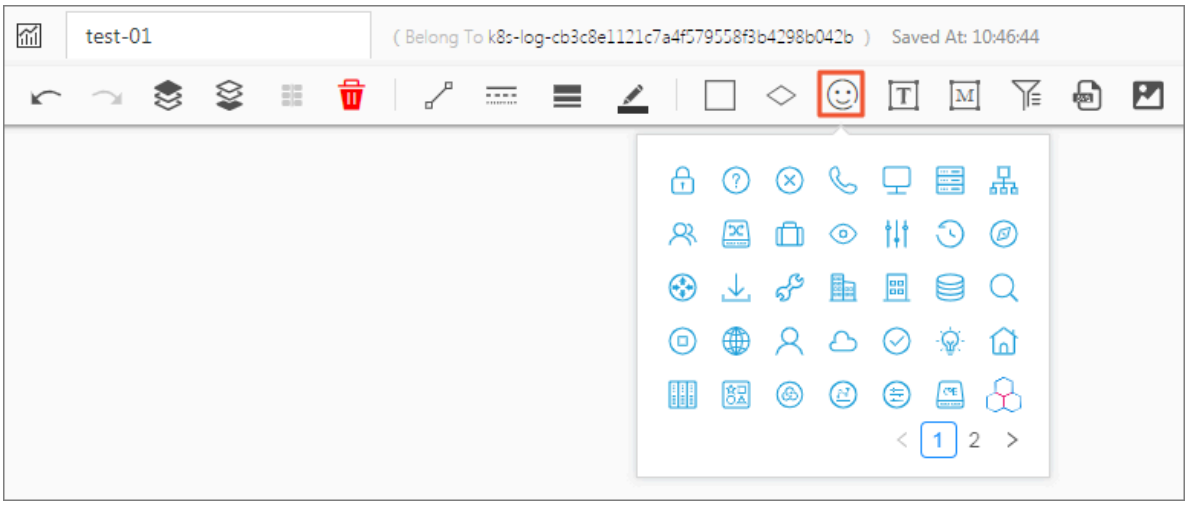
Note:
You must click Save in the upper-right corner after you make any modifications to a dashboard in editing mode. Otherwise, your changes will not be saved when you exit editing mode.

Add an element to a dashboard



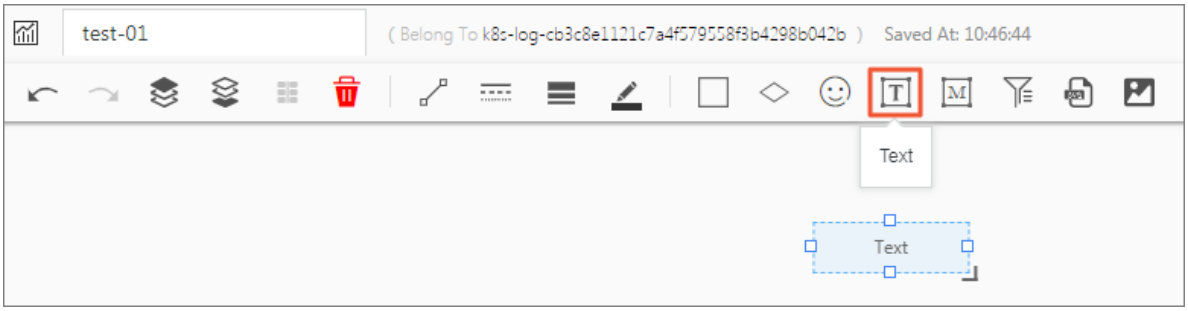
- Icons

  To add an icon to the dashboard, click the icon tool in the ribbon, select the target icon, and then drag the icon to where you want to position it in the dashboard.



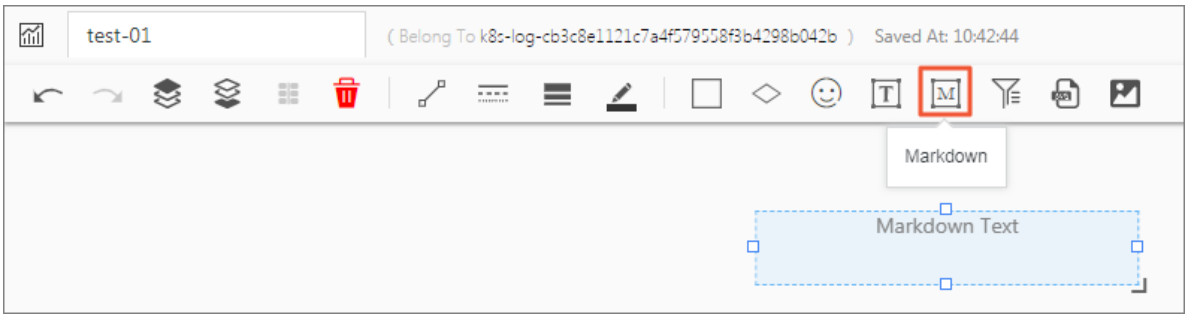
- Text

  To add text to the dashboard, drag the text icon to the target position in the dashboard. Then, double click the text box to add text.



- ***Markdown chart***

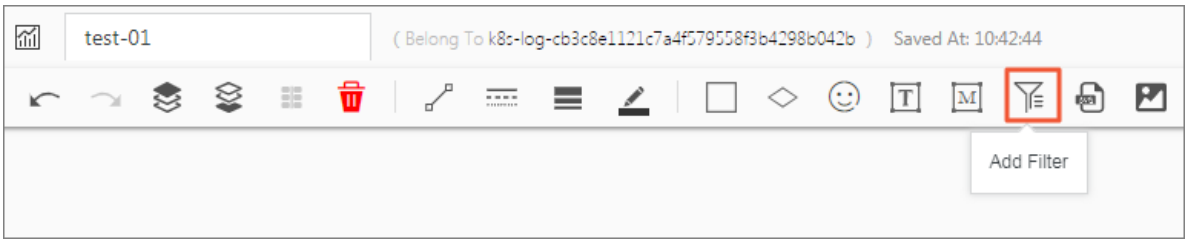
  To add a markdown chart to the dashboard, first drag the markdown tool icon from the ribbon to the target position in the dashboard. To edit the markdown box, move your pointer to the upper-right corner of the markdown box, click the More Options icon and then click Edit.



- ***Filter***

  A filter can be used to narrow down your query scope or replace variables in the dashboard.

  To add a filter to the dashboard, click the filter tool icon in the ribbon, and then set a filter in the displayed page. By default, a filter is added to the upper-left corner of the dashboard. To modify the filter settings, move your pointer to the upper-right corner of the filter box, click the More Options icon and then click Edit.



- SVG file

  To add an SVG file to the dashboard, click the SVG tool icon in the ribbon, and then click the gray area in the displayed dialog box to select an SVG file from your local directory or drag and drop an SVG file to the gray area in the displayed dialog box.

  Note:
  The maximum size of an SVG file is 10 kb.

- Image from the Web

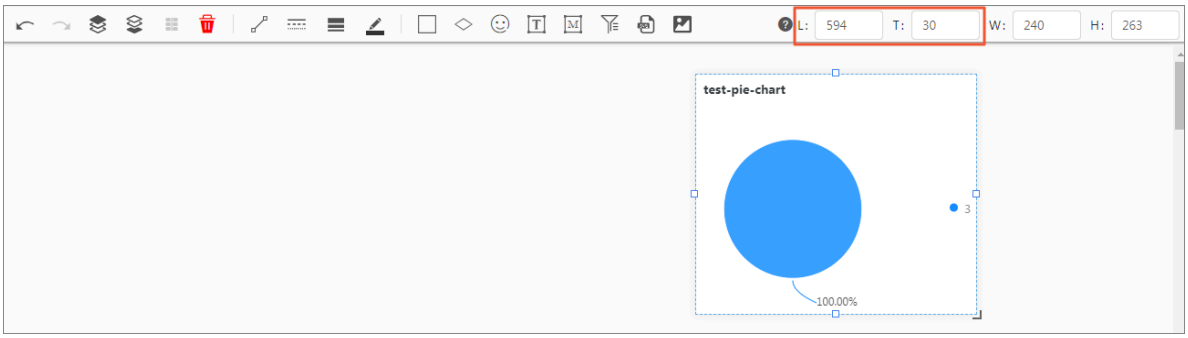
  To add an image from a website to the dashboard, click the Image icon in the ribbon, enter or paste the URL of the image in the displayed dialog box, and then click OK.

Set the dashboard layout

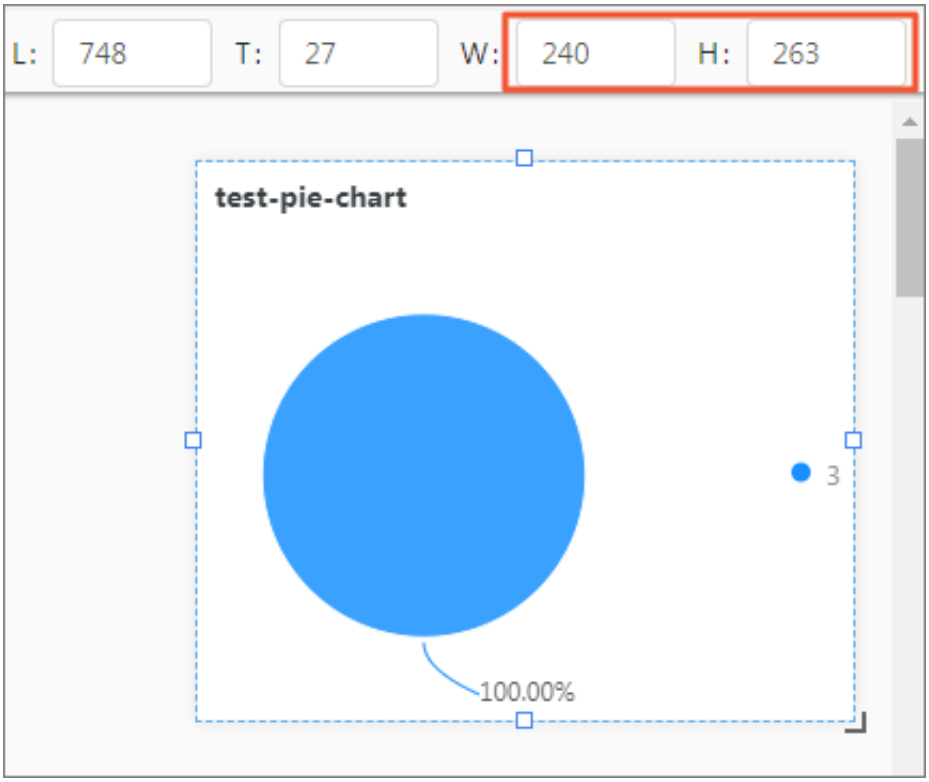
In editing mode, all the charts and elements in the dashboard can be dragged to any position or resized while maintaining proportions (except for lines used to connect charts and elements). The limit of the dashboard on the horizontal display is 1000 pixels (the vertical display has no pixel limit). o accurately set the position of a chart and the spacing between charts, we recommend that you click Show Grid Lines in the upper-right corner before setting the dashboard layout.

You can perform the following operations:

- Adjust chart position
  - Directly drag the target chart.
  - Select the target chart, and then set the L value and the T value in the ribbon.



- Adjust the width and height of a chart
  - Select the target chart, and then drag the lower-right corner of the chart to resize the chart.
  - Select the target chart, and then set the L value and the T value in the ribbon.

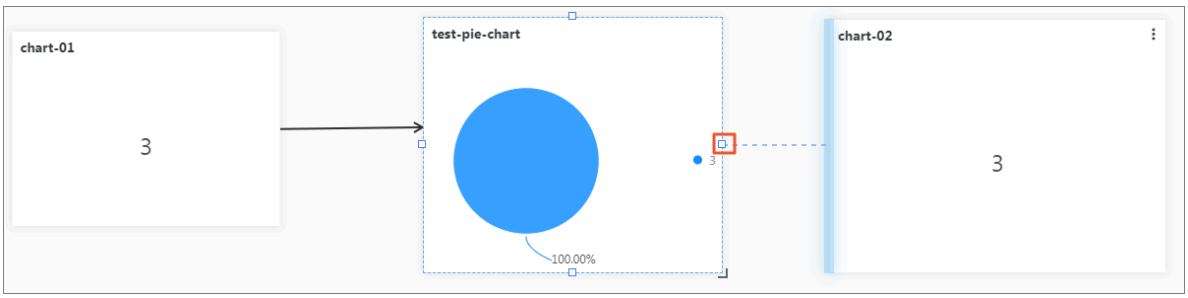


- Add a line to connect two charts

  To connect a source chart with a destination chart by using a line, follow these steps:

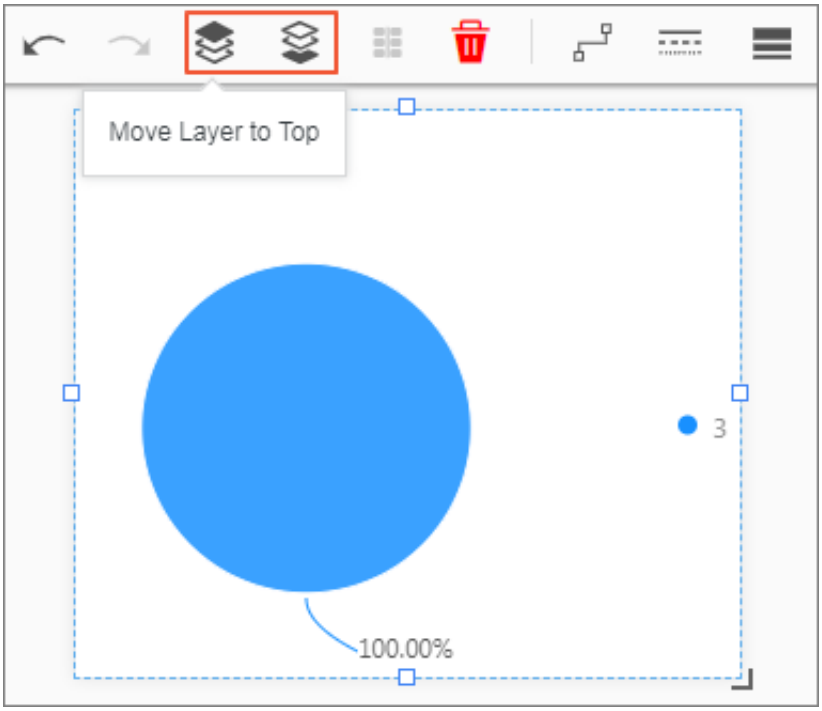
  1. Select the source chart.
  2. Click and hold a sizing handle on the outline of the source chart and drag it to one side of the destination chart.
  3. Release your cursor when the target side of the destination chart becomes blue.

  After, when you adjust the position and size of a chart that is connected with any other chart through a line, the line moves in proportion.



- Set the chart layer

  To set the chart layer, select the target chart, and then click the Move Layer to Top icon or Move Layer to Bottom icon in the ribbon.



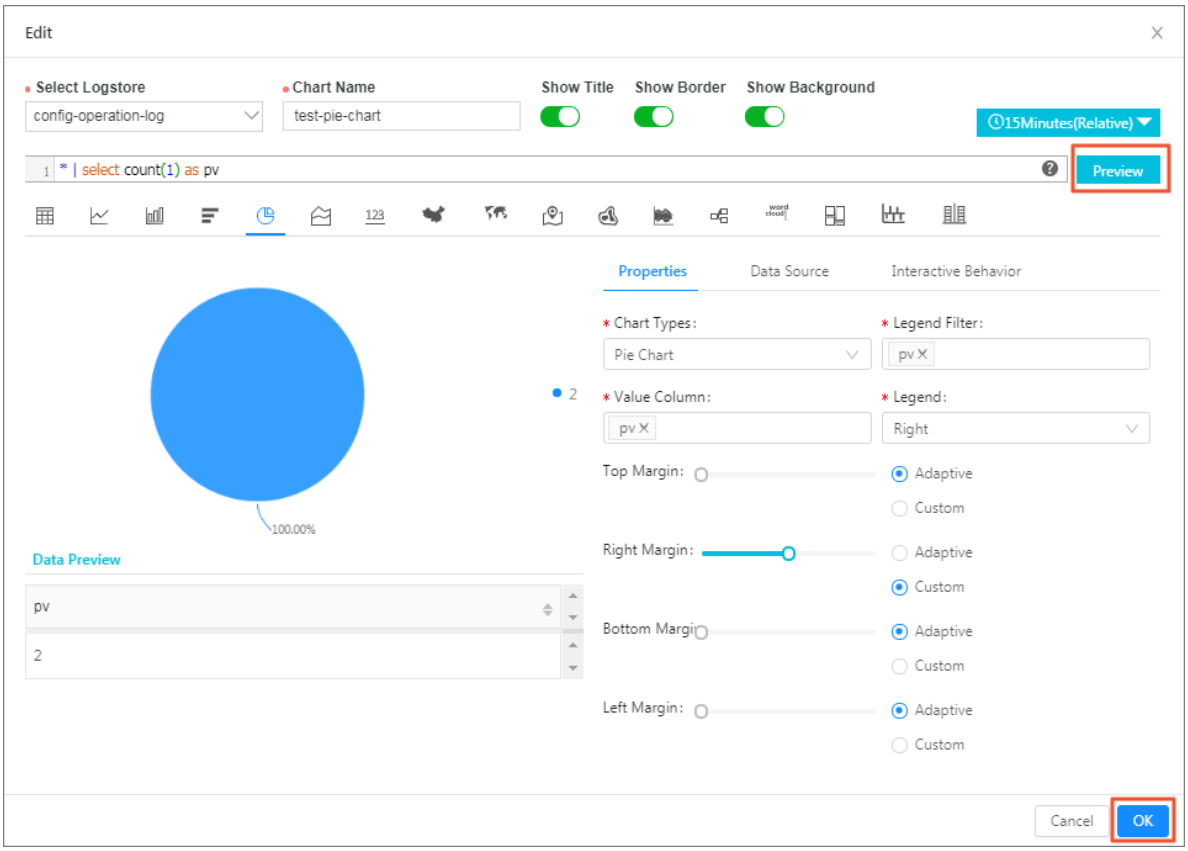
Modify a chart

In editing mode, you can modify a chart in a dashboard by performing the following operations:

- Edit a chart.

  You can modify the query statement represented by a chart and the chart properties, *set a drill-down analysis*, and perform other operations.

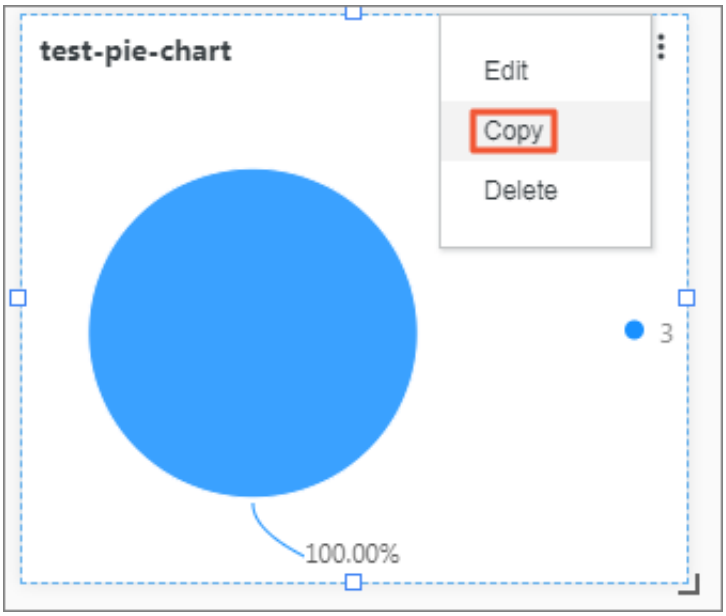
  1. In the upper-right corner of the dashboard, click Edit.
  2. In the upper-right corner of the target chart, click the More Options icon and then click Edit.
  3. In the displayed dialog box, modify the query statement, or set Properties, Data Source or Interactive Behavior.
  4. Click Preview, and then click OK.
  5. In the upper-right corner of the dashboard, click Save.



- Copy a chart

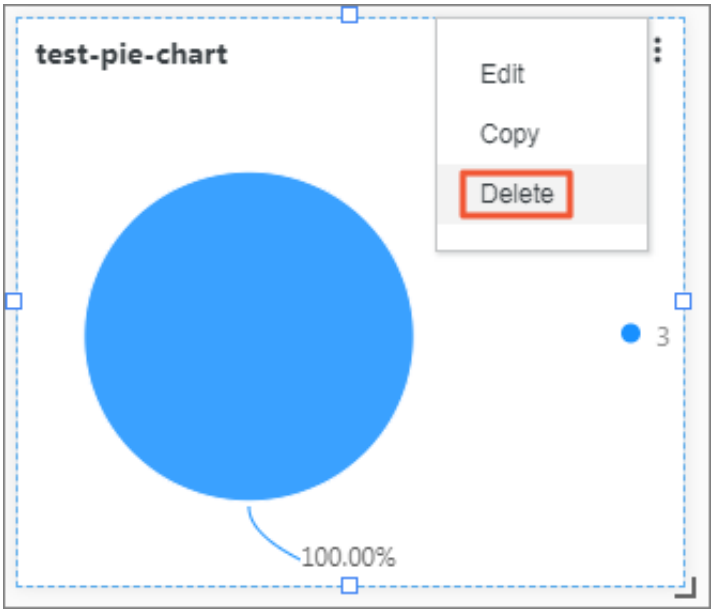
  You can create a copy of a chart to preserve all current settings of the chart.

  1. In the upper-right corner of the dashboard, click Edit.
  2. In the upper-right corner of the target chart, click the More Optionsicon and then click Copy.
  3. Drag the replicate chart to a new position and release.
  4. In the upper-right corner of the dashboard, click Save.



- Delete a chart

  1. In the upper-right corner of the dashboard, click Edit.
  2. In the upper-right corner of the target chart, click the More Options icon and then click Delete.
  3. In the upper-right corner of the dashboard, click Save.



7.2.5 Subscribe to dashboard snapshots

This topic describes how to subscribe to dashboard snapshots in Log Service. Specifically, you can set for the system to send notifications about dashboard snapshots to specific email addresses or send notifications as DingTalk chatbot messages.

Limits

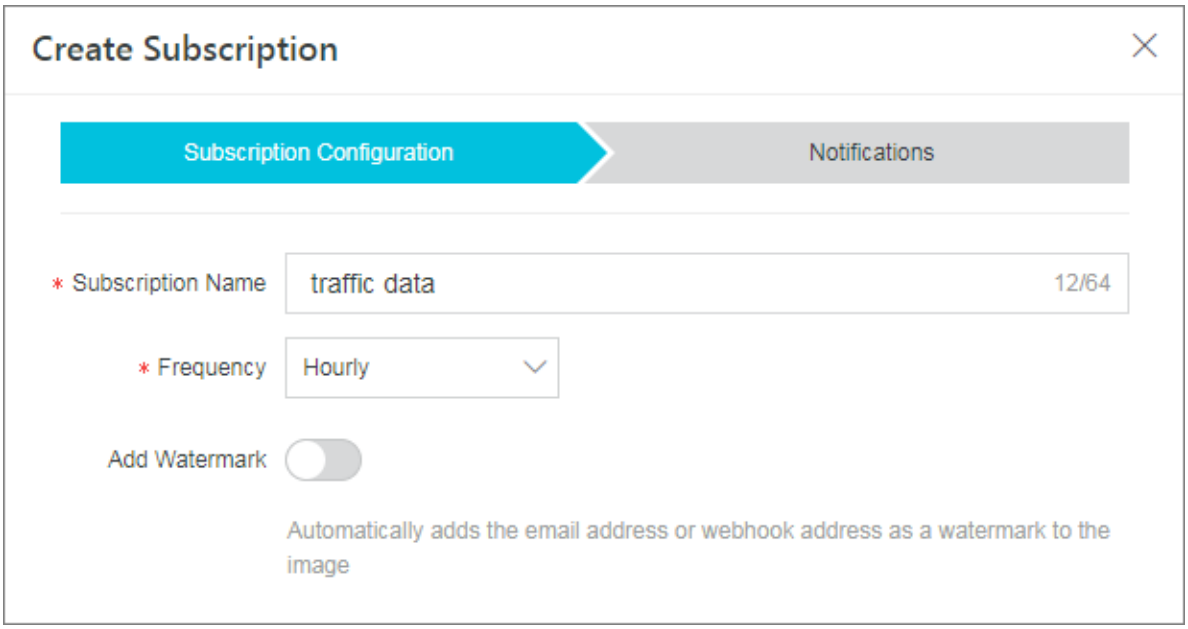
- You can create a maximum of one subscription for each dashboard.
- In each day, Log Service can send up to fifty emails for each account.
- The interval specified by a Cron expression must be at least one minute. However, we recommend that you set an interval of one hour or longer if you choose to use a Cron expression.
- The maximum number of subscriptions and alert notifications that you can create for a Log Service project is 100. If you need to increase one or both of these quotas, open a ticket.

Create a subscription

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane, click Dashboard.
3. Click the target dashboard.
4. In the upper-right corner of the page, click Subscribe.
5. Set a subscription, and then click Next.

Configuration	Description	Example
Subscription Name	A subscription name must be 1 to 64 characters in length.	Dashboard subscription
Frequency	The frequency at which dashboard snapshot notifications are sent. Available values are: · Hourly · Daily · Weekly · Fixed Interval · Cron Cron indicates a Cron expression that can be used to specify an interval in minutes at least. However, we recommend that you set an interval longer than one hour if you use a Cron expression to set this parameter.	The `* 0 / 1 * * *` Cron expression indicates that dashboard snapshot notifications are sent at one-hour intervals starting from 00:00.

Configuration	Description	Example
Add Watermark	Add watermarks to dashboard snapshots. The watermark is the email address or the access_token of the WebHook.	-

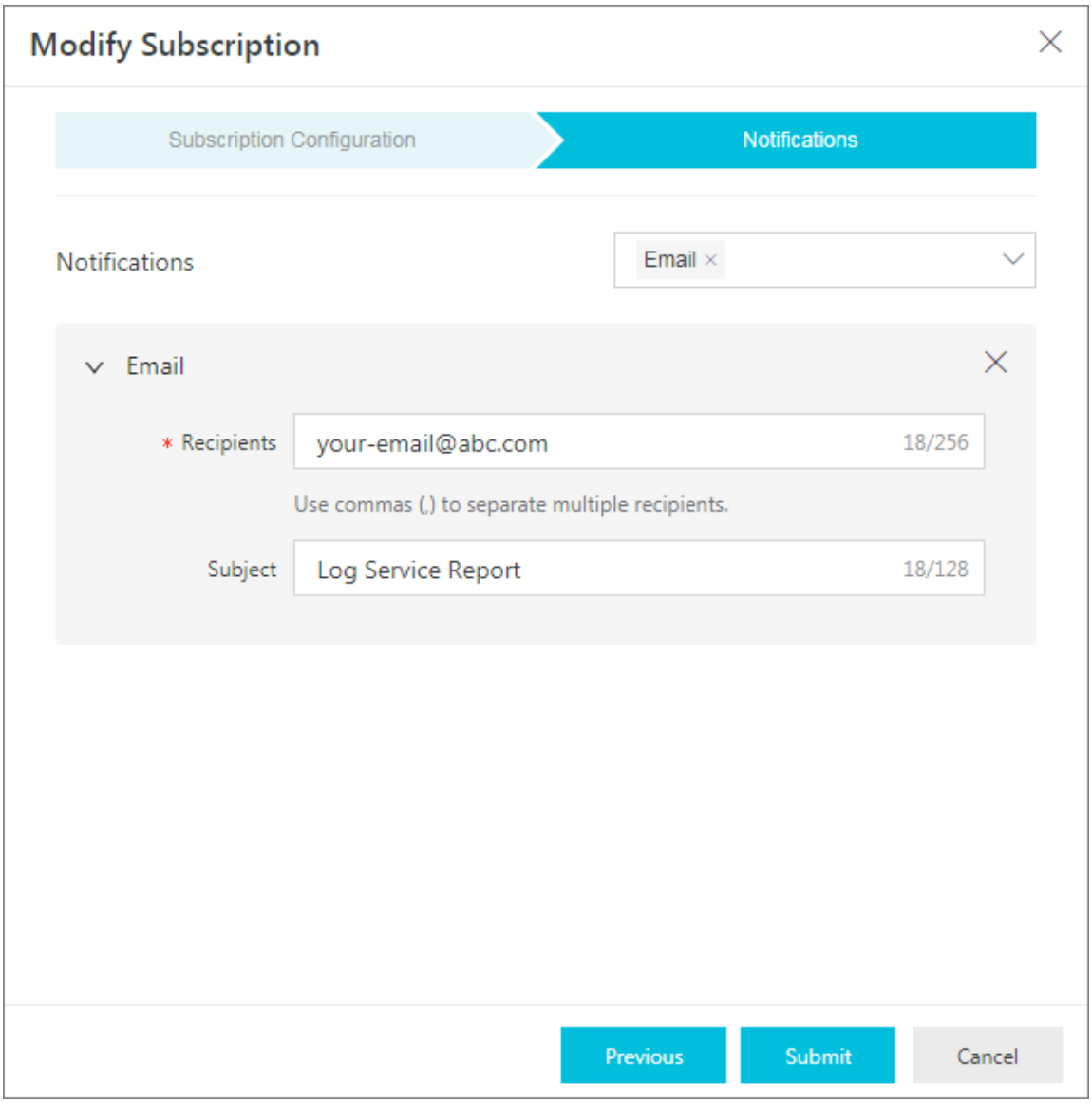


6. Set notifications.

   Dashboard snapshot notifications can be sent to a specific email address or sent as a DingTalk chatbot message.

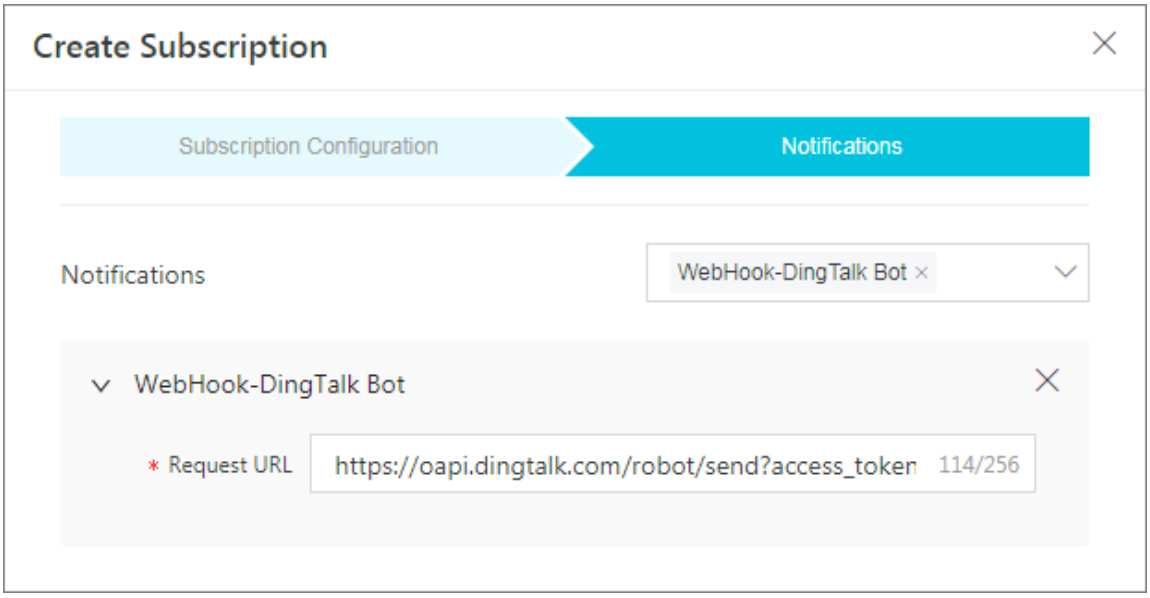
   - Email

     In the Recipients box, enter an email address. You can also set the email Subject. If you do not set any email subject, Log Service uses `Log Service Report` as the default subject.



   - DingTalk chatbot

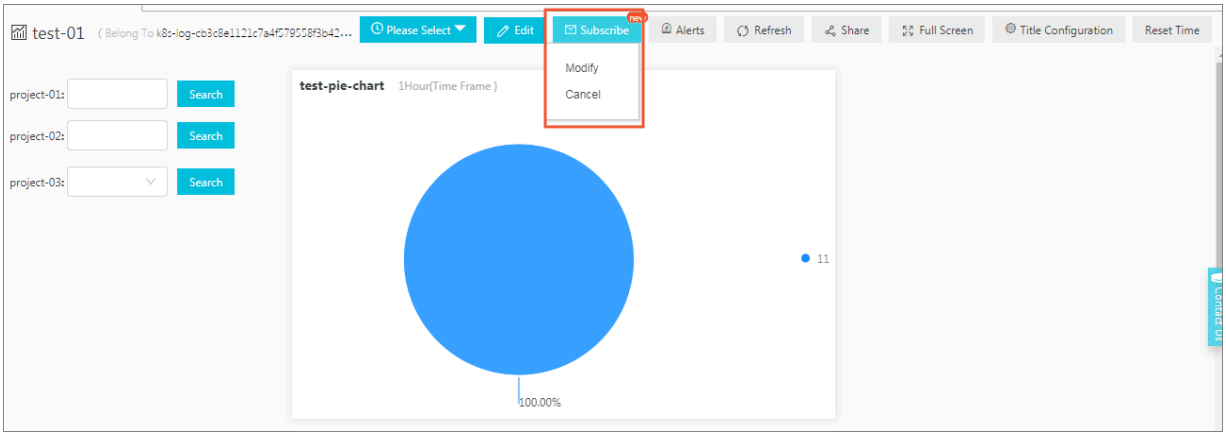
     In the Request URL box, enter the DingTalk chatbot address. For more information, see *Set a DingTalk robot*.



Modify or cancel a subscription

In the upper-right corner, choose Subscribe > Modify or choose Subscribe > Cancel according to your requirements.

Note that after you cancel a subscription, Log Service immediately stops sending dashboard snapshot notifications.



7.2.6 Drill-down analysis

Log Service analysis charts provide the drill-down function in addition to the basic data visualization functions. When you add a chart to the dashboard, you can modify the configurations in the drill-down list to present the data in a more powerful way.

Drilling is an essential function for data analysis. It allows you to view more detailed information by moving to different layers of data. Drilling includes rolling up and drilling down. By rolling up, you move to higher data layers with more summarized information. By drilling down, you move to deeper data layers to reveal more detailed

information. By drilling down data layer by layer, you can view data more clearly, extract more value out of data, and make more accurate decisions faster based on the data.

Log Service supports drill-down analysis for analysis charts in the dashboard. After you set the dimension and layer of a drilldown, you can jump to the analysis page of a deeper dimension by clicking a data point in the dashboard. Analysis charts in the dashboard are actually results of query statements. If you configure a drill-down analysis for the request status table and add it to the dashboard, you can click a request status type in the dashboard to view logs of the request status.

Limits

In Log Service, that charts that support drill-down analysis include:

- Table
- Line chart
- Column chart
- Bar chart
- Pie chart
- Single value chart
- Area chart
- Tree map

Prerequisites

1. You have enabled and configured an index.
2. You have configured a saved search, dashboard, and custom link to jump to.
3. Configure a placeholder variable of statements in the saved search and dashboard to be jumped to. For more information, see *Saved search* and *Create and delete a dashboard*.

Procedure

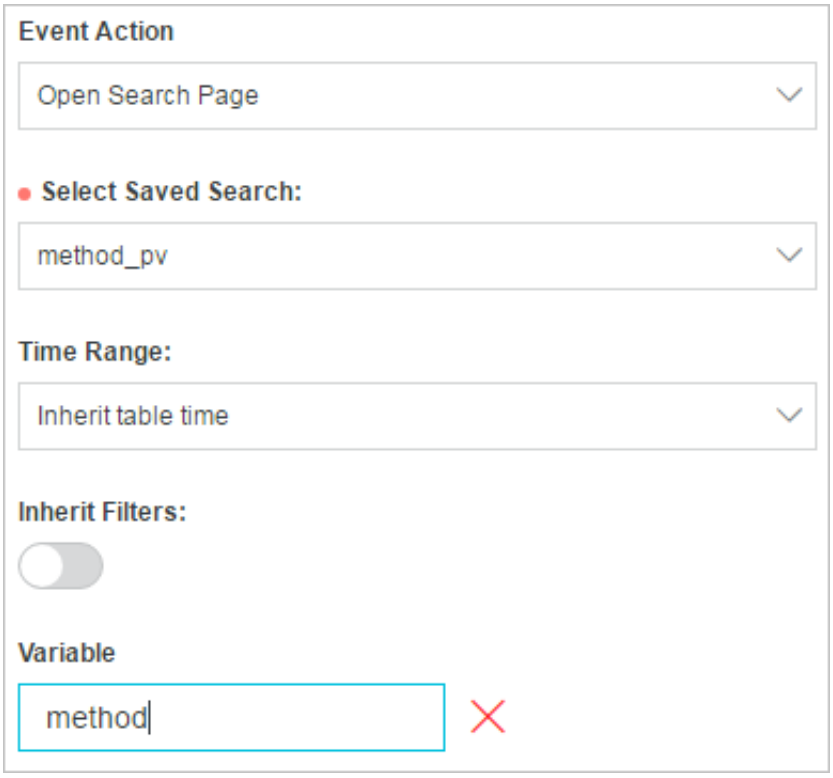
1. Click Search in the LogSearch column in the Logstores list.
2. Enter your query and analysis statement, set the time range, and click Search & Analysis.
3. On the Graph tab, select Chart type and configure Properties of the chart.

4. Click Drilldown on the right side of the Properties column, and configure a drill-down event.

   By default, the drilldown configuration is disabled. A drill-down event is triggered by a single clicking. A drill-down event is an event triggered by clicking the analysis chart on the dashboard page. After you configure a drill-down event and click the chart data in the dashboard, your current page jumps to the corresponding page according to your configured drill-down event. Choose one of the following four options.

   - Disable: Disables the drill-down function.
   - Open Search Page: Enables drill down. The drill-down event is to open the search page.

     When you click a value in the chart, the system replaces the placeholder configured in the saved search statement with the chart value you clicked, and then performs a deeper query according to the chart value.



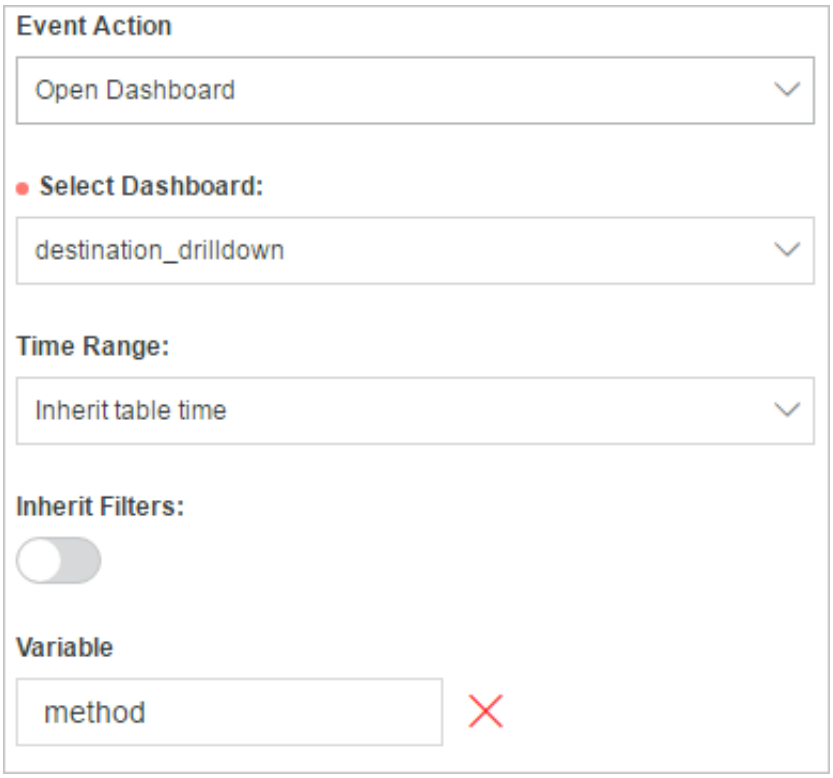
Configuration	Description
Saved Search	Name of the saved search to be jumped to. For information about configuring a saved search, see *Saved search*.

Configuration	Description
Time Range	Configure the time range for the saved search to be jumped to. The default is Inherit table time. That is, after you jump to the saved search by clicking the chart in the dashboard, the time range of the query statement is the table time configured in the dashboard by default when the event is triggered.
Inherit Filters	If you turn on the Inherit Filters switch, the system synchronizes filtering conditions added in the dashboard to the saved search, and adds the filtering conditions before the query statement by using `AND` .
Variable	Click Add Variable to enter a placeholder variable name. When the variable in the saved search matches the name of the added variable, the variable in the query statement is replaced with the chart value that triggers the drill-down event. This flexibly changes the dimension of the saved search.  Note: To add a variable, you must first configure a placeholder variable of the query statement in the saved search to which your page will jump.

- Open Dashboard: Enables the drill-down function. The drill-down event is to open a dashboard.

  The chart in the dashboard is the chart-form result of the query statement . You need to pre-configure a placeholder of the query statement in the dashboard chart to be jumped to. When you click a chart value in the upper

layer dashboard, the system replaces the pre-configured placeholder with the chart value, and then performs a deeper query according to the chart value.

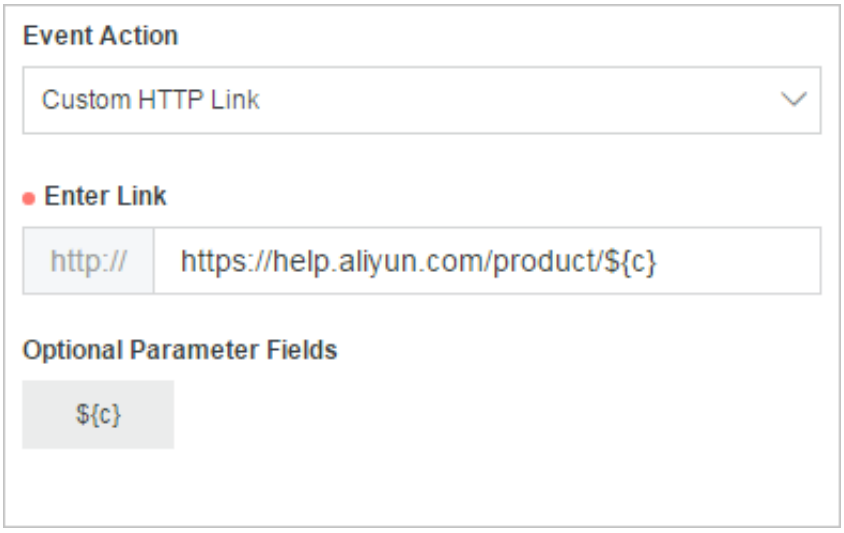


Configuration	Description
Dashboard	Name of the dashboard to be jumped to. For information about configuring a dashboard, see *Create and delete a dashboard*.
Time Range	Configure the time range of the dashboard to be jumped to. The default is Inherit table time. That is, after you jump to the configured dashboard by clicking the chart in the current dashboard, the time range of the configured dashboard is the time configured in the current dashboard chart by default where the drill-down event is triggered.
Inherit Filters	If you turn on the Inherit Filters switch, the system synchronizes filtering conditions added in the dashboard where an event is triggered to the dashboard to be jumped to. The filtering conditions are added before the query statement by using `AND`.

Configuration	Description
Variable	Click Add Variable to enter a placeholder variable name. When the query statement variable of the analysis chart in the dashboard to be jumped to matches the name of the added variable, the query statement variable of the analysis chart is replaced with the chart value that triggers the drill-down event. This flexibly changes the query statement of the analysis chart in the target dashboard. Note: To add a variable, you must first configure a placeholder variable of the query statement in the dashboard to jump to.

- Custom HTTP link: Enables the drill- down function. The drill-down event is to open a custom HTTP link.

  The part of path in the HTTP link that is the hierarchical path of the destinatio n file to be accessed. After you add optional parameter fields to the part of path in a custom HTTP link and click the chart content of the dashboard, the system replaces the added parameter fields with the chart value to jump to the relocated HTTP link.



Configuration	Description
Link	Destination address to be jumped to.
Optional Parameter Fields	By clicking an optional parameter variable, you can replace part of the HTTP link with the chart value that triggers a drill-down event.

5. Click Add to New Dashboard, configure the dashboard, and click OK.

   You can then view the analysis chart on the dashboard page, and click the chart to view deeper analysis results.

Example

For example, you can store collected Nginx access logs in the Logstore named accesslog, display the common analysis scenarios of Nginx logs in the dashboard named RequestMethod, and display the trend of PV distribution over time in the dashboard named destination_drilldown. You can configure drill-down analysis for the table of request methods, add it to the RequestMethod dashboard, and configure the drill-down event to jump to the destination_drilldown dashboard. In the RequestMethod dashboard, click each request method to jump to the destinatio n_drilldown dashboard to view the corresponding PV trend.

The procedure is as follows:

1. Configure a dashboard to be jumped to.

   a. Filter logs according to request types and view the PV changes over time.

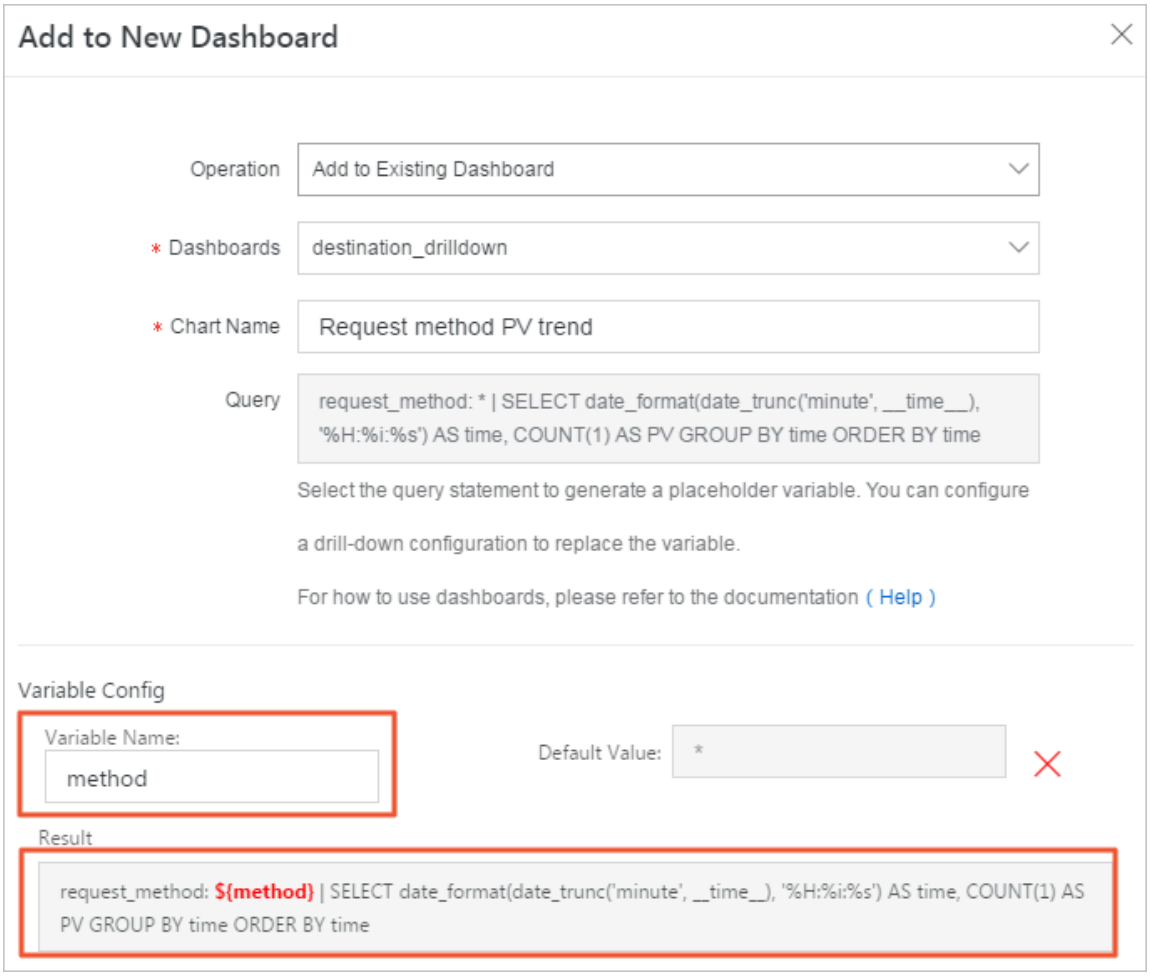
      Query statement:

      ```
      request_me  thod : * |  SELECT   date_forma  t ( date_trunc ('
      minute ',  __time__ ), '% H :% i :% s ')  AS   time ,  COUNT ( 1
      )  AS   PV   GROUP   BY   time   ORDER   BY   time
      ```

   b. Use a line chart to display the query result and save the line chart to the dashboard.

      When saving the chart to the dashboard, configure `*` as a placeholder named method. If the variable of the drill-down event that jumps to this saved search is

also named method, you can replace `*` with your clicked chart value to perform a query and analysis again.

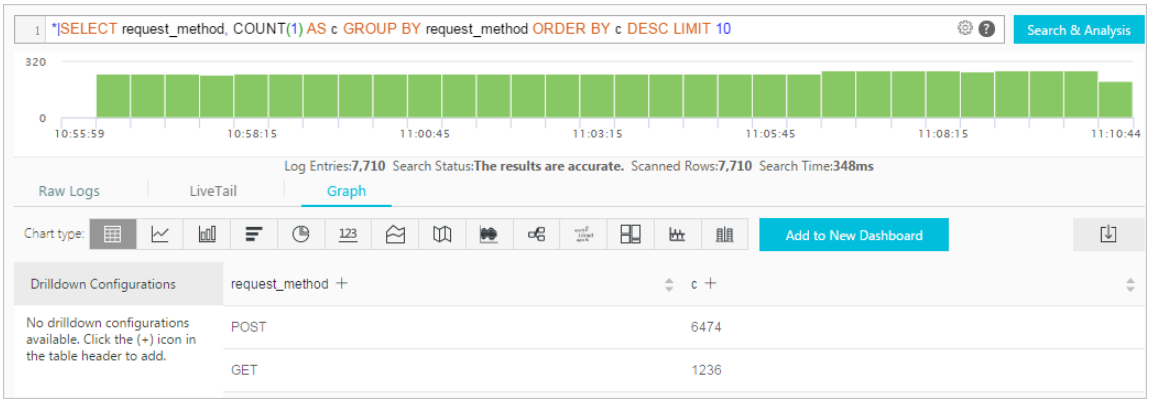


2. Configure a chart that triggers drill-down analysis, and add the chart to the dashboard.

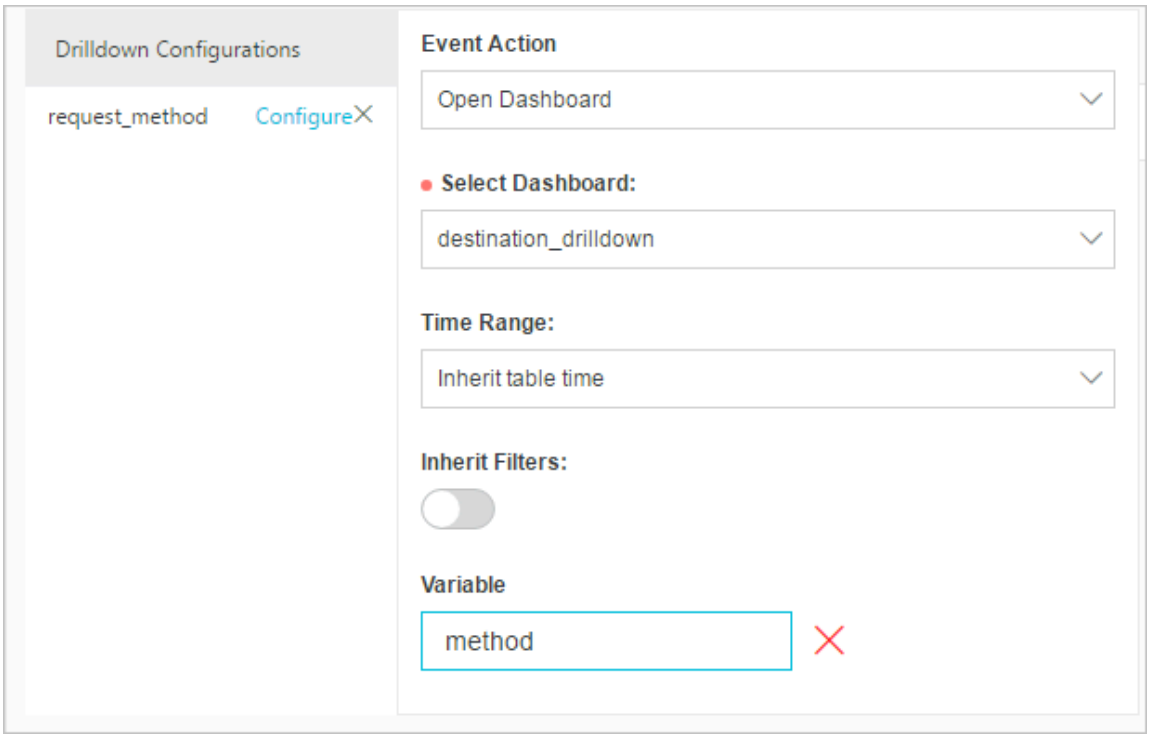
   a. On the search page, use SQL to analyze the number of logs of each request method in the Nginx access logs, and display the result in a table.

```
*| SELECT request_me thod , COUNT ( 1 ) AS c GROUP BY
 request_me thod ORDER BY c DESC LIMIT 10
```

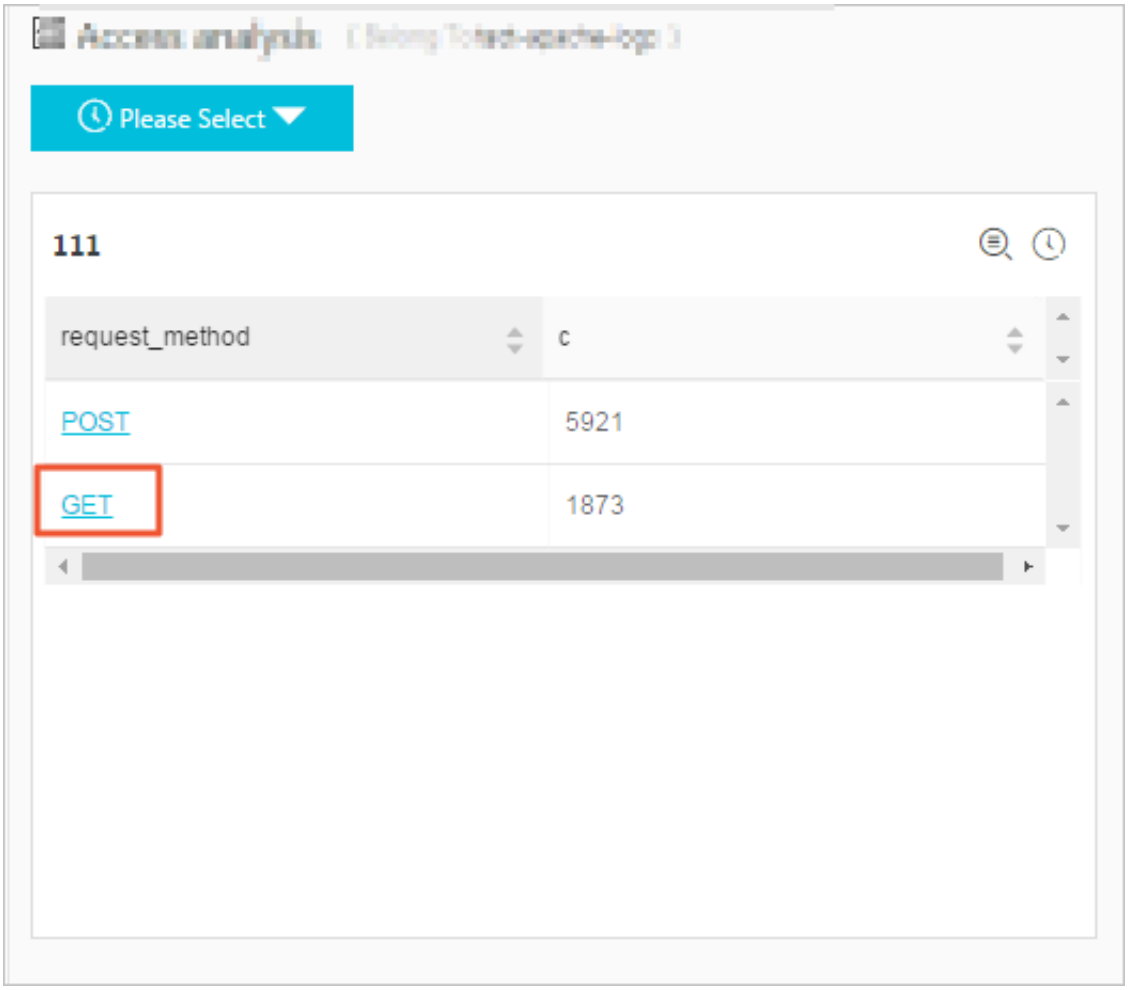
   Query result:



   b. Configure drill-down analysis for the `request_me thod` column in the table:

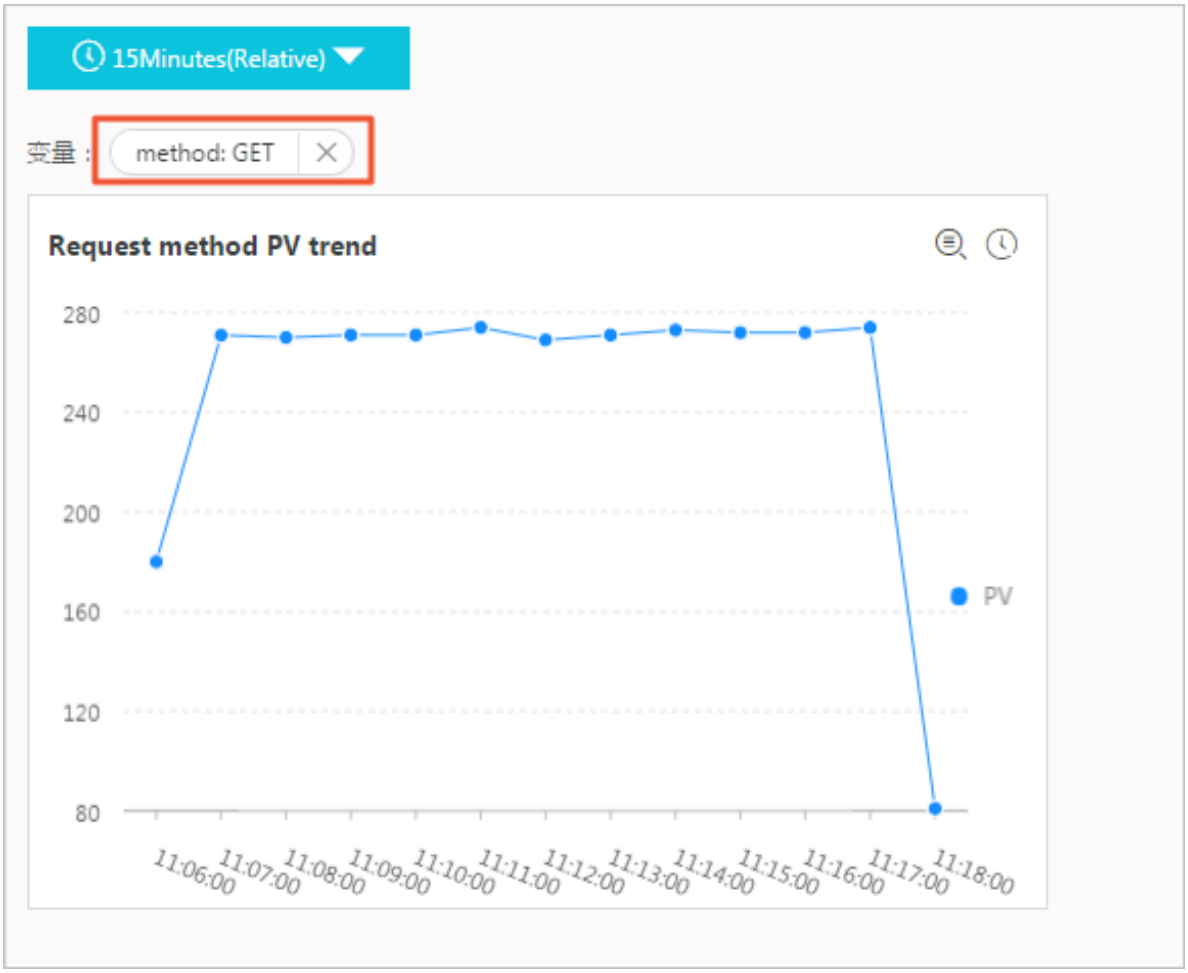


3. Click the GET request in the RequestMethod dashboard.



4. Jump to the destination_drilldown dashboard.

   The page automatically jumps to the dashboard configured in *1*. The `*` in the query statement is replaced with `GET`, the chart value you clicked on. The dashboard then shows changes of the GET request PV over time.



7.2.7 Set and use a filter in a dashboard

This topic describes how to set and use a filter in a Log Service dashboard. A filter in a dashboard can help you refine search results or replace placeholder variables with specified values across the whole dashboard.

Overview

Each chart in a Log Service dashboard functions as a search-and-analysis statement. By setting a filter in a dashboard, you are performing an action for each search-and-analysis statement that is indicated by a corresponding chart in the dashboard at one time. To do so, add one or more conditions for filtering the result of each search-and-analysis statement, or replace placeholder variables in each search-and-analysis

statement with a specified value. You can set the two following types of filters in a dashboard.

- Filter type: Add a key value as a condition before the search-and-analysis statement`[ search query ]` to filter the result of the search-and-analysis statement. Then, the new search-and-analysis statement can be executed as `key : value AND [ search query ]`. This indicates to search for logs containing `key : value` in the result of the original search-analysis.

  For this type of filters, you can add multiple key values. A filter that contains multiple key values can filter out the logs that contain any of the key values from the search result of the search-and-analysis statement.
- Replace Variable type: Set a value to replace the placeholder variables in the search-and-analysis statements that are indicated by corresponding charts.

Components

Each filter contains the following two core components:

- The key value component, which indicates a filter operation.
- The list item component, which corresponds to the key.

Prerequisites

- An index is enabled and configured. For more information, see *Enable and set indexes*.
- A dashboard is created. For more information, see *Dashboard*.
- A placeholder variable in the target search-and-analysis statement is set. This is so that you can use a filter to replace the placeholder variable in the search-and-analysis statement with a specific value.

Procedure

1. Log on to the *Log Service console*, and then click the target project name.
2. Find the target Logstore, and then click Search in the LogSearch column.
3. In the left-side navigation pane, click the name of the target dashboard.
4. In the progress bar of the dashboard, click Edit.

5. Click the icon  , and then set the filter configuration parameters:

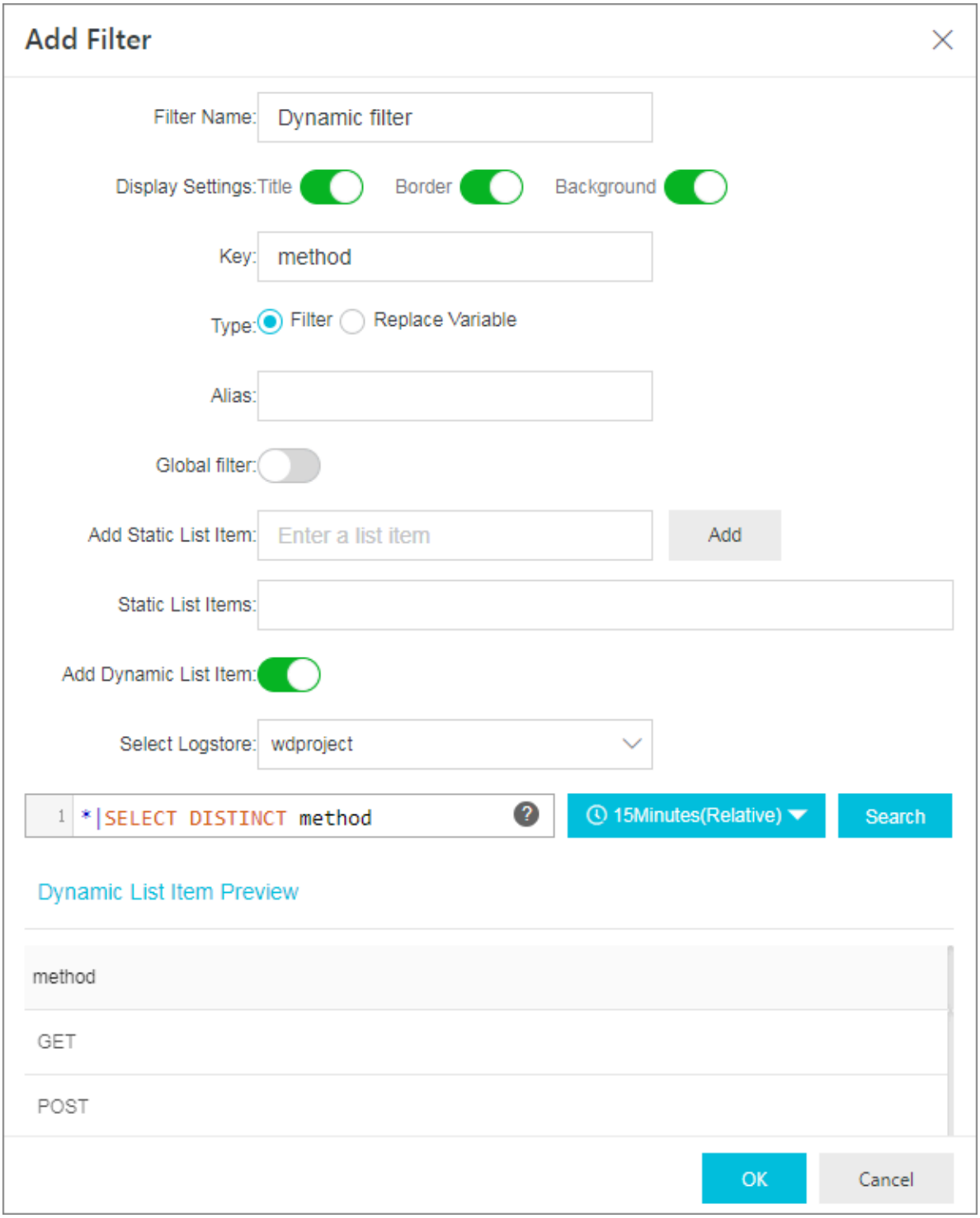
   - Chart Name: the filter name.
   - Title: Turn on this switch to display the filter title in the dashboard.
   - Border: Turn on this switch to display the filter borders.
   - Background: Turn on this switch to add a white background for the filter.
   - Key: There are Filter type and Replace Variable type keys. The key place holder variable must be as the one set in Prerequisites.
     - For the Filter type, set this parameter as a condition to filter the execution result of the search-and-analysis statement.
     - For the Replace Variable type, set this parameter as a placeholder variable.
   - Type: Select Filter or Replace Variable.
     - If you select Filter, a List Item refers to the value of the condition used to filter the results of a search-and-analysis statement.
     - If you select Replace Variable, a List Item refers to the value that replaces a specific variable placeholder.

     For both of these types, you can set more than one value. Furthermore, after you set a filter containing multiple values in a dashboard, you can select a value as needed when viewing the dashboard.
   - Alias: Set the column alias.

     This parameter is only required by the Filter type. After you set a column alias for a filter, the alias is shown in the filter in the dashboard.
   - Global filter: Set whether to filter a specific value globally (that is, applied to all fields).

     By default, this parameter is disabled. Furthermore, you can only set this parameter for the Filter type.
   - List Item: There are two types Static List Item and Dynamic List Item.
     - Static List Item: You need to manually add list items to the filter. To add items, in the Add Static List Item box, enter list items and click Add.
     - Dynamic List Item: Set the system to dynamically display list items in the filter. The list items are the results of the search-and-analysis statement that you set. To enable the Add Dynamic List Item switch, enter a search-and-

analysis statement, and then click Search. Then, the dynamic list items are displayed.



6. Click OK.

Examples

In the case that you have collected Nginx logs, you can analyze the Nginx logs by one of the two following methods. (For information about how to collect Nginx logs, see *Collect and analyze Nginx access logs*.)

- Example 1: Analyze the logs by using different time granularities

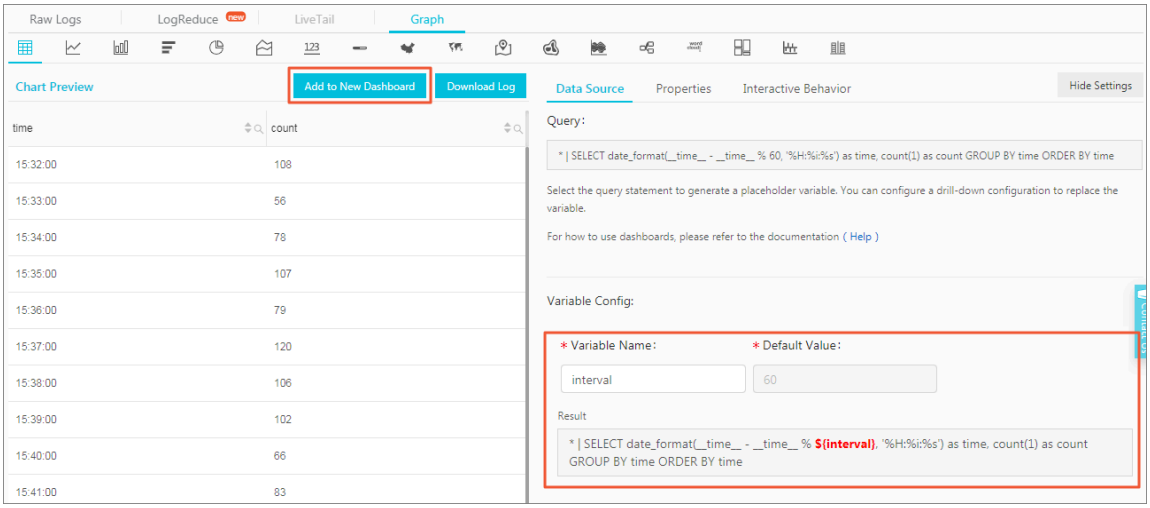
  You can use a search-and-analysis statement to view the PV in terms of different time granularities, such as the second or minute granularity. To change the time granularity to be used, you must change the value of `__time__ - __time__ % 60`. Changing the search-and-analysis statement is the typical method, which

is inefficient for multiple times of operations. If you set a filter, you can more quickly replace the target placeholder variable with a specific value.

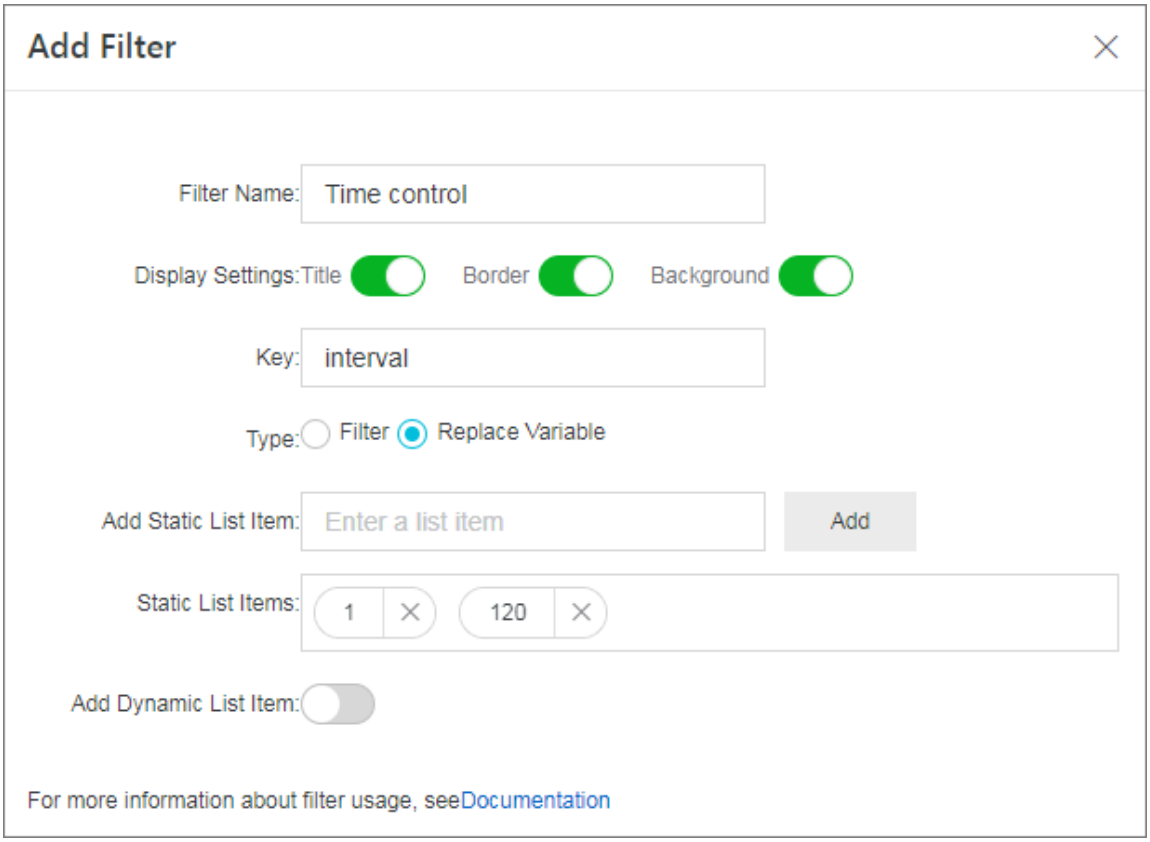
1. Use the following search-and-analysis statement to view the PV each minute:

```
* | SELECT date_format ( __time__ - __time__ % 60 , '% H :% i :% s ') as time , count ( 1 ) as count GROUP BY time ORDER BY time
```

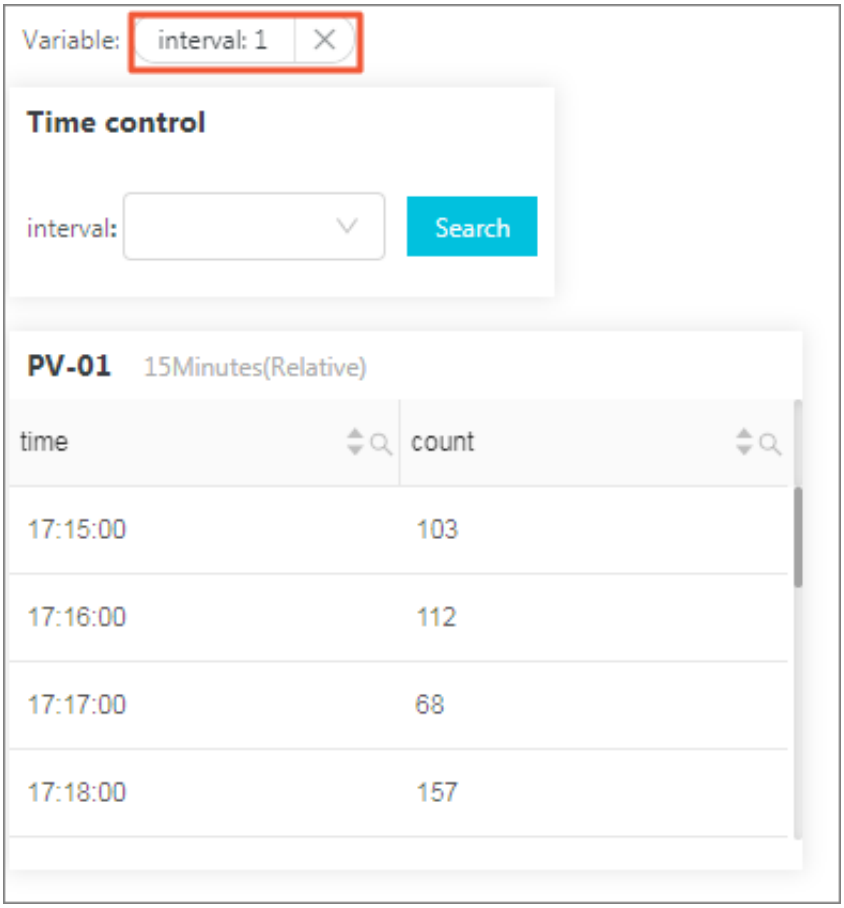
2. Add the search-and-analysis statement as a chart to the target dashboard, select `60` in the statement to generate a placeholder variable, and set the variable as `interval`. The default value of this variable is 60.



3. Set a filter in the target dashboard.
   - Type: Select Replace Variable.
   - Key: Enter `interval`.
   - Static List Item: Add `1` (each second) and `120` (every two minutes) as the static list items.



4. In the target dashboard, select `1` from the Search drop-down list of the filter to view the data recorded in seconds.



Note:

The search-and-analysis statement that has a new placeholder variable value is then changed to the following:

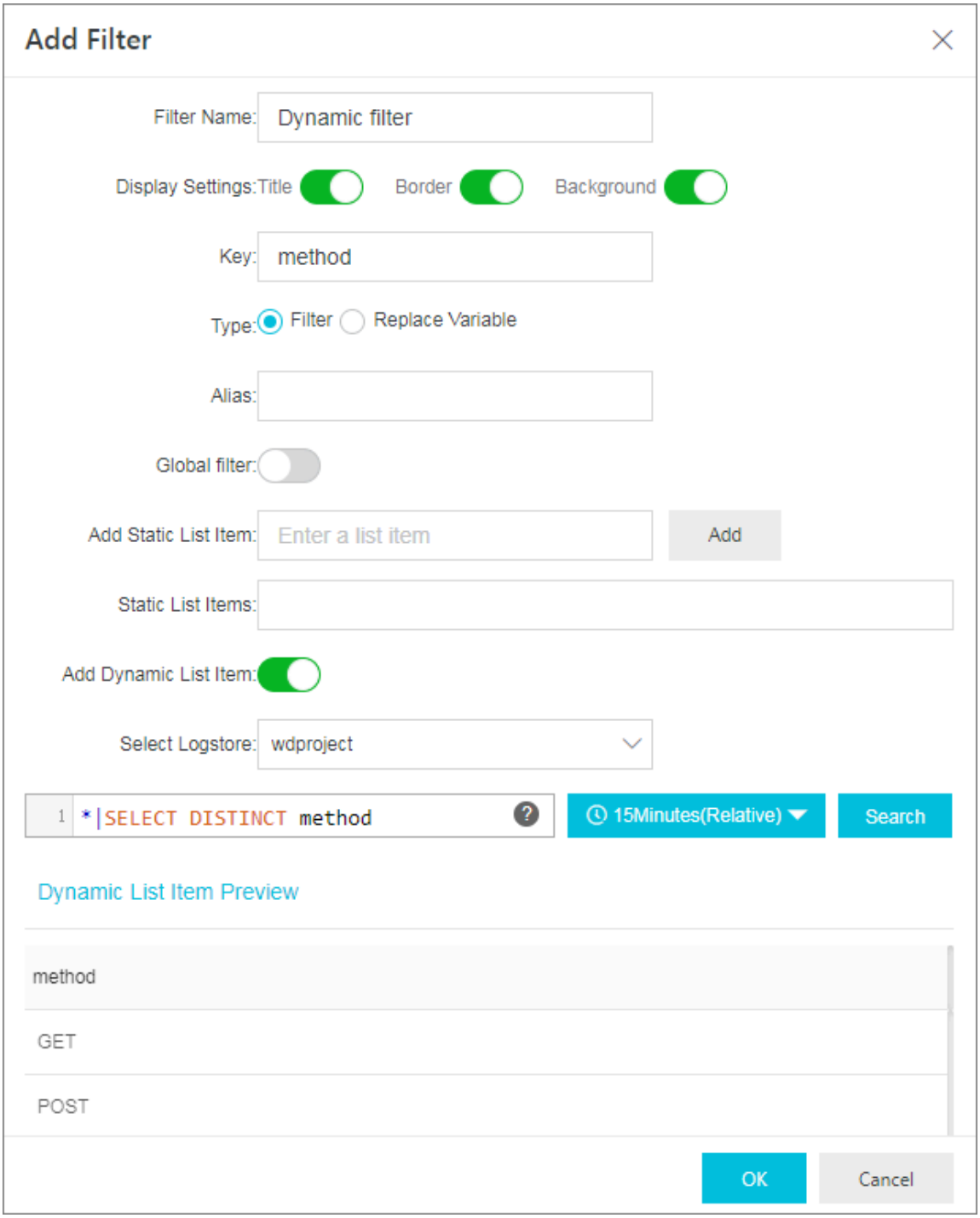
```
* | SELECT date_forma t ( __time__ - __time__ % 1 , '% H
 :% i :% s ') as time , count ( 1 ) as count GROUP BY
  time ORDER BY time
```

- **Example 2: Dynamically switch filter methods**

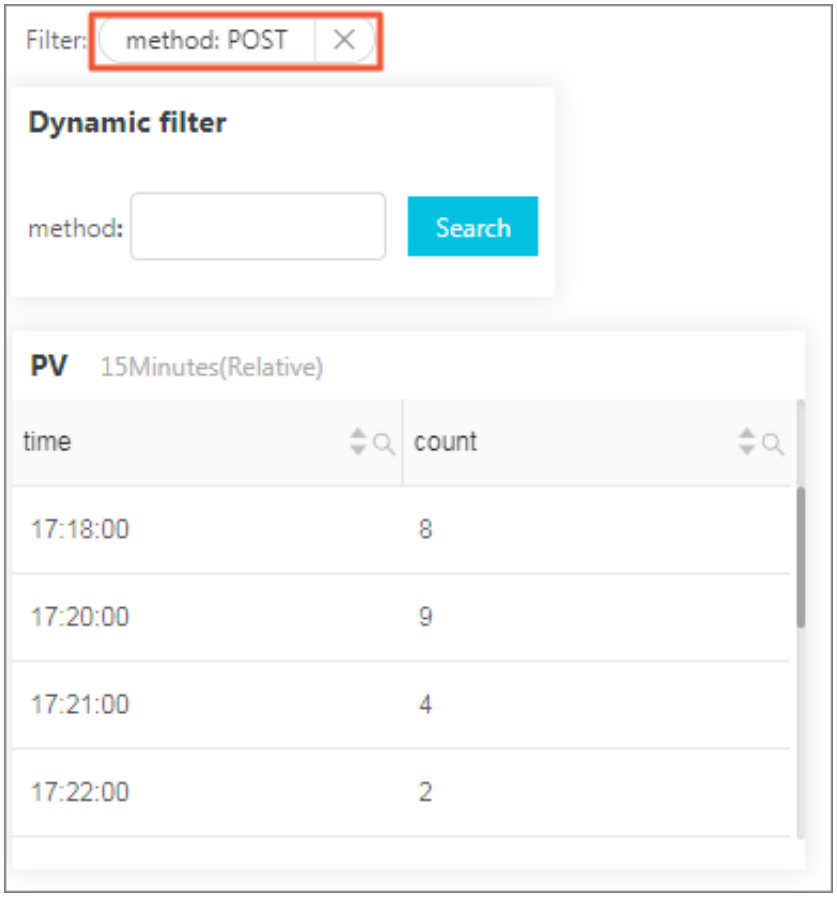
  By adding dynamic list items to a filter, you can dynamically switch different request methods. In example 1, the search-and-analysis statement that starts with `*` means no condition is set to filter the results of the statement with the statement

applies to all logs. You can add a new filter to view the PV data of different request methods.

1. Set a filter in the target dashboard.
   - Type: Select the Filter type.
   - Key: Enter `method`.
   - Select Logstore: Select the Logstore where the target dashboard belongs.
   - Add Dynamic List Item: Enable this switch, and enter a search-and-analysis statement.



2. In the target dashboard, select `POST` from the Search drop-down list of the filter.



Note:

The chart only displays the data of the accesses that use the `POST` method.

The search-and-analysis statement is then changed to the following:

```
(*) and ( method : POST ) | SELECT date_forma t ( __time__
  - __time__ % 60 , '% H :% i :% s ') as time , count ( 1
 ) as count GROUP BY time ORDER BY time
```

7.2.8 Markdown chart

With Log Service, you can add a markdown chart to the dashboard. In the markdown chart, you can insert images, links, videos, and other elements to make your dashboard page more friendly.

By adding multiple analysis charts to the dashboard when querying and analyzing log data, you can quickly view multiple analysis results and monitor the status of multiple services in real time. With Log Service, you can also add a markdown chart to the dashboard. The markdown chart is edited by using the markdown language. You can insert images, links, videos, and other elements to the markdown chart to make your dashboard page more friendly.

Markdown charts are created according to different requirements. To optimize the dashboard information expression, you can insert text such as background information, chart descriptions, page notes, and extension information in a markdown chart. To easily switch to other query pages, you can insert saved searches or dashboard links of other projects in a markdown chart. To enrich your dashboard information and make your dashboard functions more flexible, you can insert custom images in a markdown chart.

Scenarios

By using a markdown chart, you can customize links that redirect to other dashboards of the current project. You can also insert an image to go with each link to make it easier to tell them apart. You can also insert a markdown chart to describe the parameters in a chart.

Figure 7-17: Scenarios



Prerequisites

1. Log data is collected.
2. A dashboard is configured.

Procedure

1. On the Dashboard page, click Edit in the upper-right corner.
2. Click Create Markdown.

3. In the displayed page, configure markdown chart properties.

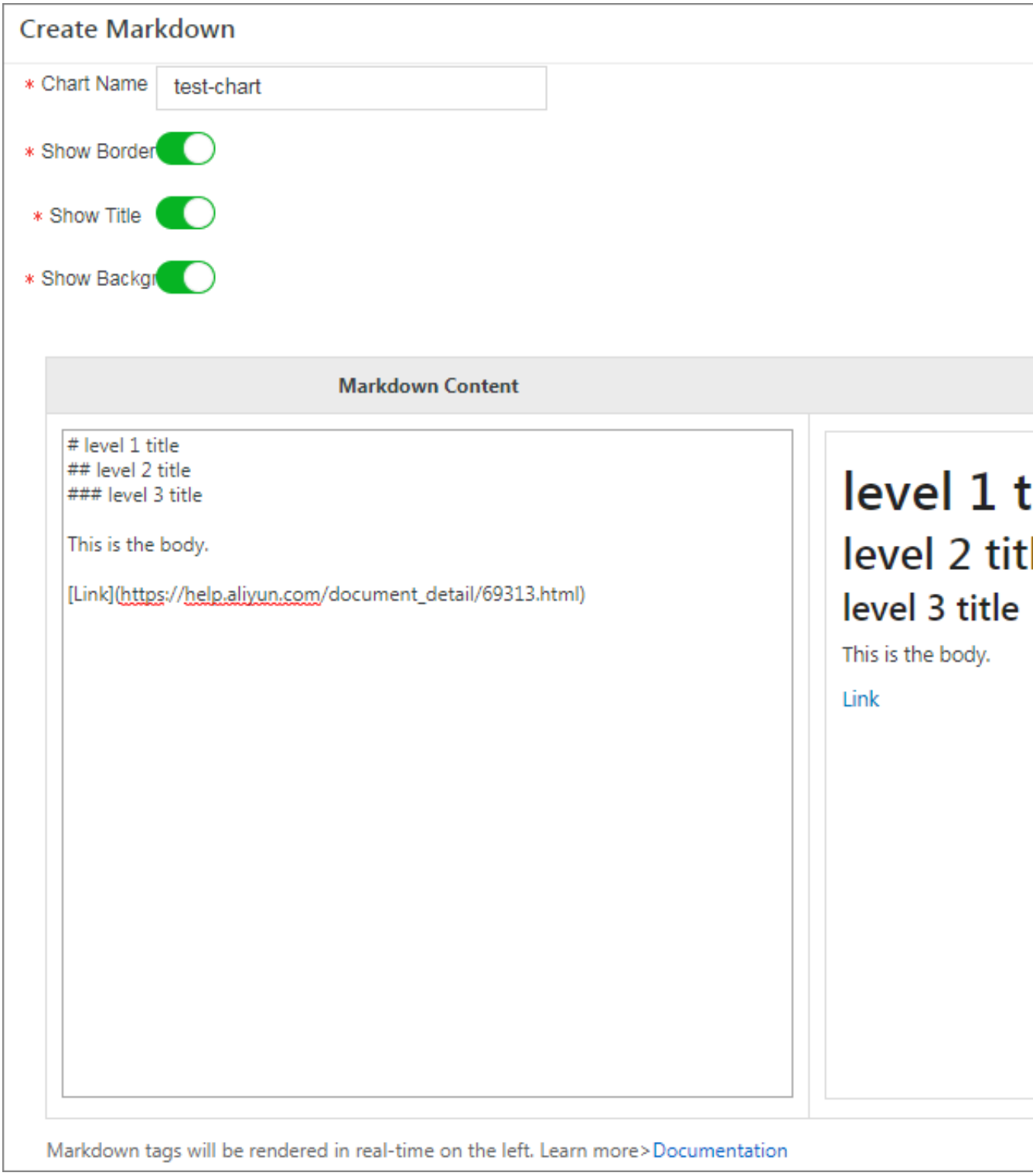
Configuration item	Description
Chart name	Name of your markdown chart.
Show border	Turn on the Show Border switch to add borders for your markdown chart.
Show title	Turn on the Show Title switch to display your markdown chart title in the dashboard.
Show background	Turn on the Show Background switch to add white background for your markdown chart.

4. Edit the Markdown Content.

   In the Markdown Content area, enter markdown statements. The Show Chart section on the right displays the preview in real time. Modify the markdown statements according to the preview content.

5. After you complete the configuration, click OK.

Figure 7-18: Create a markdown chart



After you complete the configuration, the created markdown chart is displayed under the current dashboard.

Modify a markdown chart

- Modify the chart location and size
  1. On the Dashboard page, click Edit in the upper-right corner.
  2. Drag the markdown chart to adjust its location, and drag the lower-right corner of the chart to adjust its size.
  3. Click Create in the upper-right corner.
- Modify the chart title
  1. On the Dashboard page, click Edit in the upper-right corner.
  2. Enter a new title in the chart title box.
  3. On the Dashboard page, click Save in the upper-right corner, and click OK in the displayed dialog box.
- Modify the chart content
  1. On the Dashboard page, click Edit in the upper-right corner.
  2. Click Edit in the upper-right corner of the markdown chart.
  3. Modify the chart configuration and click OK.
- Delete a chart
  1. On the Dashboard page, click Edit in the upper-right corner.
  2. Click Delete in the upper-right corner of the markdown chart.
  3. On the Dashboard page, click Save in the upper-right corner, and click OK in the displayed dialog box.

Common markdown syntax

- Title

  Markdown statement:

```
#  Level   1   title
##  Level   2   title
```

```
###  Level   3   title
```

Figure 7-19: Title preview



- **Link**

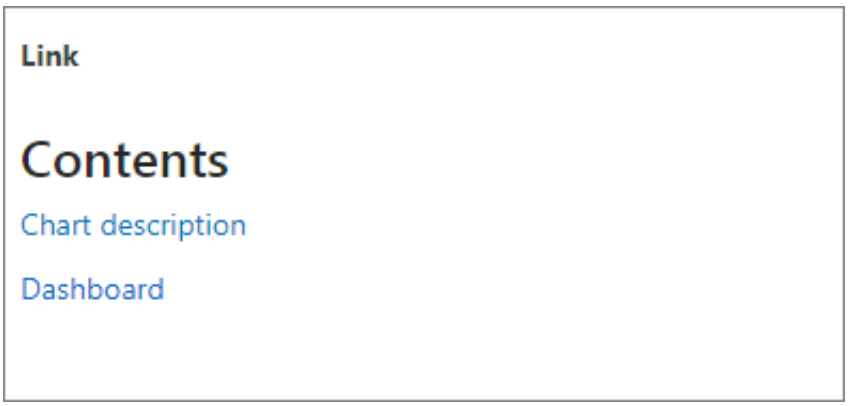
  Markdown statement:

```
###  Contents

[ Chart   descriptio  n ]( https :// www . alibabaclo  ud . com /
 help / doc - detail / 69313 . html )

[ Dashboard ]( https :// www . alibabaclo  ud . com / help / doc -
 detail / 59324 . html )
```

Figure 7-20: Link preview



- **Image**

  Markdown statement:

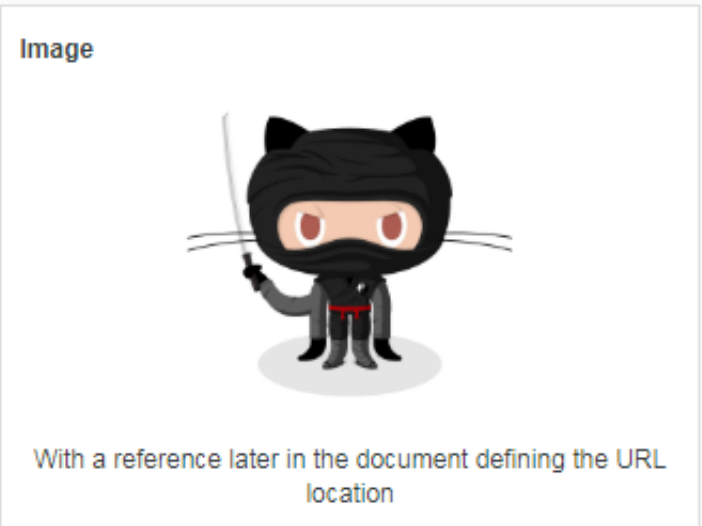
```
< div   align = center >

![  Alt   txt ][ id ]
```

```
 With   a   reference   later   in   the   document   defining   the
   URL   location

[ id ]:  https :// octodex . github . com / images / dojocat . jpg
 " The   Dojocat "
```

Figure 7-21: Image preview



- **Special mark**

  Markdown statement:

```
---

 __Advertis  ement  :) __

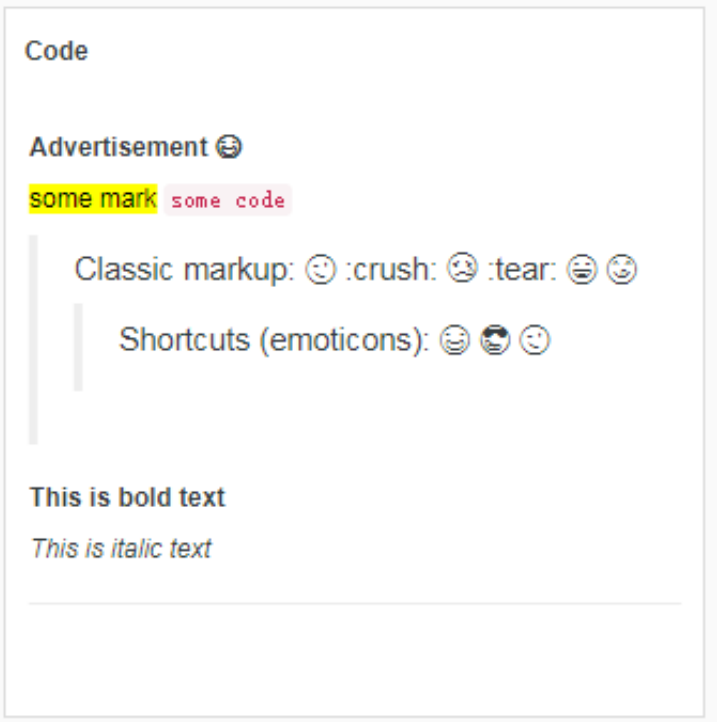
== some   mark ==   ` some   code `
>  Classic   markup : : wink : : crush : : cry : : tear : : laughing
 : : yum :
>>  Shortcuts  ( emoticons ): :-)   8 -) ;)

 __This   is   bold   text__

* This   is   italic   text *
```

```
---
```

Figure 7-22: Special mark preview



For more information about markdown syntax, see *Markdown syntax*.

7.3 Other visualization methods

7.3.1 Console sharing embedment

After configuring the collection and query analysis functions for Log Service, you may want to directly use the log query analysis and dashboard functions, or share these log-related functions with other users. In this case, using RAM for sharing may generate management costs for many subaccounts. To avoid this, Log Service allows you to log on to embedded pages through a single point for integrated query analysis and dashboard.

Context

Benefits

You can embed a specific Logstore query page and dashboard page into a self-built website. This gives you access to the analysis and visualization features of Log Service without logging on to Alibaba Cloud.

- The independent query page and dashboard page can be easily embedded into any website.
- You can generate a logon link by using the security token service (STS) and control the operation permissions, such as ready-only permission, by using remote access management (RAM).

Procedure

1. Log on to your self-built website.

   After logon, the Web server STS obtains a temporary identity for you.

   - For more information on STS, see *Overview*.
   - Grant the user access to specified Logstores. For details, see *Grant RAM sub-accounts permissions to access Log Service*.

2. Request Alibaba Cloud logon service for the logon token.

   After getting the temporary AccessKey pair and security token from STS, call the logon service interface to obtain the logon token.

   > **Note:**
   > The security token returned by STS may contain special characters. When the token contains special characters, encode them with URL-encoding method before using the token.

   Request example:

```
http :// signin . aliyun . com / federation ? Action = GetSigninT
oken
                  & AccessKeyI  d =< Temporary   Access   Key
pair   returned   by   the   STS >
                  & AccessKeyS  ecret =< Temporary   secret
returned   by   the   STS >
                  & SecurityTo  ken =< Security   token   returned
  by   the   STS >
```

```
                    & TicketType = mini
```

3. Generate a logon-free link.

   a) Generate an access link along with the link to the embedded page after getting the logon token.

   The token is valid for three hours. Therefore, we recommend you generate a new logon token and redirect each access request to an embedded link to your self-built website through a 302 message.

   Request example:

```
http :// signin . aliyun . com / federation ? Action = Login
                          & LoginUrl =< Address   to   which   a
  logon   request   is   redirected   upon   a   logon   failure
,  which   is   usually   configured   to   the   URL   on   your
  self - built   website   through   a   302   message ;>
                          & Destinatio  n =< Log   Service
  page   to   be   accessed .  Pages   for   query   and
dashboard   are   supported .>
                          & SigninToke  n =< Logon   token
obtained >
```

   b) Embedded page.

   - A complete page for query and analysis (multiple tags are allowed):

```
https :// sls4servic  e . console . aliyun . com / next /
project /< Project   name >/ logsearch /< Logstore   name >?
hideTopbar = true & hideSideba  r = true
```

   - Query page:

```
https :// sls4servic  e . console . aliyun . com / next /
project /< Project   name >/ logsearch /< Logstore   name >?
isShare = true & hideTopbar = true & hideSideba  r = true
```

   - Dashboard page:

```
https :// sls4servic  e . console . aliyun . com / next /
project /< Project   name >/ dashboard /< Dashboard   name >?
isShare = true & hideTopbar = true & hideSideba  r = true
```

   The sample code in Java, PHP, and Python is as follows:

   - *Java*:

```
< dependency >
                    < groupId > com . aliyun </ groupId >
                    < artifactId > aliyun - java - sdk -
 sts </ artifactId >
                    < version > 3 . 0 . 0 </ version >
                    </ dependency >
                    < dependency >
                    < groupId > com . aliyun </ groupId >
```

```
                    < artifactId > aliyun - java - sdk -
core </ artifactId >
                    < version > 3 . 5 . 0 </ version >
                    </ dependency >
                    < dependency >
                    < groupId > org . apache . httpcompon
ents </ groupId >
                    < artifactId > httpclient </
artifactId >
                    < version > 4 . 5 . 5 </ version >
                    </ dependency >
                    < dependency >
                    < groupId > com . alibaba </ groupId >
                    < artifactId > fastjson </ artifactId
>
                    < version > 1 . 2 . 47 </ version >
                    </ dependency >
```

- *PHP*
- *Python*

7.3.2 Use JDBC to count and visualize logs

MySQL is a popular relational database. Many softwares support obtaining MySQL data by using MySQL transport protocol and SQL syntax. You can connect to MySQL if you know SQL syntax. Log Service provides MySQL protocol to query and analyze logs. You can use a standard MySQL client to connect to Log Service and use the standard SQL syntax to compute and analyze logs. Clients that support the MySQL transport protocol include MySQL client, JDBC, and Python MySQLdb.

Using bike sharing logs as an example, the following section describes how to use JDBC to connect to Log Service and read log data, the MySQL protocol and SQL syntax to compute logs, and DataV to visualize log data or computation results on a big screen.

JDBC scenarios:

- Use a visualization tool such as DataV, Tableau, or Kibana to connect to Log Service through the MySQL protocol.
- Use libraries such as JDBC in Java or MySQLdb in Python to access Log Service and process query results in the program.

Data example

A bike sharing log contains the user's age, gender, battery usage, vehicle ID, operation latency, latitude, lock type, longitude, operation type, operation result, and unlocking

type. Data is stored in `Logstore : ebike` of project: `project : trip_demo` . The region where the project resides is cn-hangzhou.

A sample log is as follows:

```
Time  : 10 - 12   14 : 26 : 44
__source__ :  11 . 164 . 232 . 105
__topic__ :  v1
age :  55
battery :  118497 . 673842
bikeid :  36
gender :  male
latency :  17
latitude :  30 . 2931185245
lock_type :  smart_lock
longitude :  120 . 052840484
op :  unlock
op_result :  ok
open_lock :  bluetooth
userid :  292
```

Prerequisites

Log indexing and analysis functions have been enabled for each column of Logstore through the console or API.

JDBC statistics

1. Create a Maven project and add JDBC dependency in pom dependency.

```
< dependency >
 < groupId > MySQL </ groupId >
 < artifactId > mysql - connector - java </ artifactId >
 < version > 5 . 1 . 6 </ version >
</ dependency >
```

2. Create a Java class and use JDBC in code for query.

```
/**
*  Created   by   mayunlei   on   2017 / 6 / 19 .
*/
 import   com . mysql . jdbc .*;
 import   java . sql .*;
 import   java . sql . Connection ;
 import   java . sql . Statement ;
/**
*  Created   by   mayunlei   on   2017 / 6 / 15 .
*/
 public   class   jdbc  {
  public   static   void   main ( String   args []){
      // Input   your   configurat  ion   here .
      final   String   endpoint  = " cn - hangzhou - intranet . sls
 . aliyuncs . com ";// Log   Service   intranet   or   VPC   domain
   name
      final   String   port  = " 10005 "; // The   MySQL   protocol
   port   of   Log   Service .
      final   String   project  = " trip - demo ";
      final   String   logstore  = " ebike ";
```

```
     final   String   accessKeyI  d  = "";
     final   String   accessKey  = "";
     Connection   conn  =  null ;
     Statement   stmt  =  null ;
     try  {
        // Step   1 :  Load   the   JDBC   driver .
         Class . forName (" com . mysql . jdbc . Driver ");
        // Step   2 :  Create   a   link .
         conn  =  DriverMana  ger . getConnect  ion (" jdbc : mysql
://"+ endpoint +":"+ port +"/"+ project , accessKeyI  d , accessKey
);
        // Step   3 :  Create   a   statement .
         stmt  =  conn . createStat  ement ();
        // Step   4 :  Define   query   statements .  Query   the
  number   of   logs   that   are   generated   on   October   11
,  2017   and   meet   the   condition   op  = " unlock ",  and
query   the   average   operation   latency .
         String   sql  = " select   count ( 1 )  as   pv , avg (
latency )  as   avg_latenc  y   from  "+ logstore +" " +
                " where   __date__  >= ' 2017 - 10 - 11   00 : 00 :
00 ' " +
                "  and   __date__  < ' 2017 - 10 - 12   00 : 00 :
00 '" +
                "  and   op  =' unlock '";
        // Step   5 :  Execute   query   conditions .
         ResultSet   rs  =  stmt . executeQue  ry ( sql );
        // Step   6 :  Extract   the   query   result .
         while ( rs . next ()){
            // Retrieve   by   column   name
             System . out . print (" pv :");
            // Obtain   pv   from   the   result .
             System . out . print ( rs . getLong (" pv "));
             System . out . print (" ;  avg_latenc  y :");
            // Obtain   avg_latenc  y   in   the   result .
             System . out . println ( rs . getDouble (" avg_latenc
y "));
             System . out . println ();
         }
         rs . close ();
    }  catch  ( ClassNotFo  undExcepti  on   e ) {
         e . printStack  Trace ();
    }  catch  ( SQLExcepti  on   e ) {
         e . printStack  Trace ();
    }  catch  ( Exception   e ) {
         e . printStack  Trace ();
    }  finally  {
         if  ( stmt  ! =  null ) {
             try  {
                 stmt . close ();
             }  catch  ( SQLExcepti  on   e ) {
                 e . printStack  Trace ();
             }
         }
         if  ( conn  ! =  null ) {
             try  {
                 conn . close ();
             }  catch  ( SQLExcepti  on   e ) {
                 e . printStack  Trace ();
             }
         }
    }
}
```

```
}
```

Use DavaV to access and display data

Visualized large-screen DataV displays data and connects to Log Service to read log data or display log computation results.

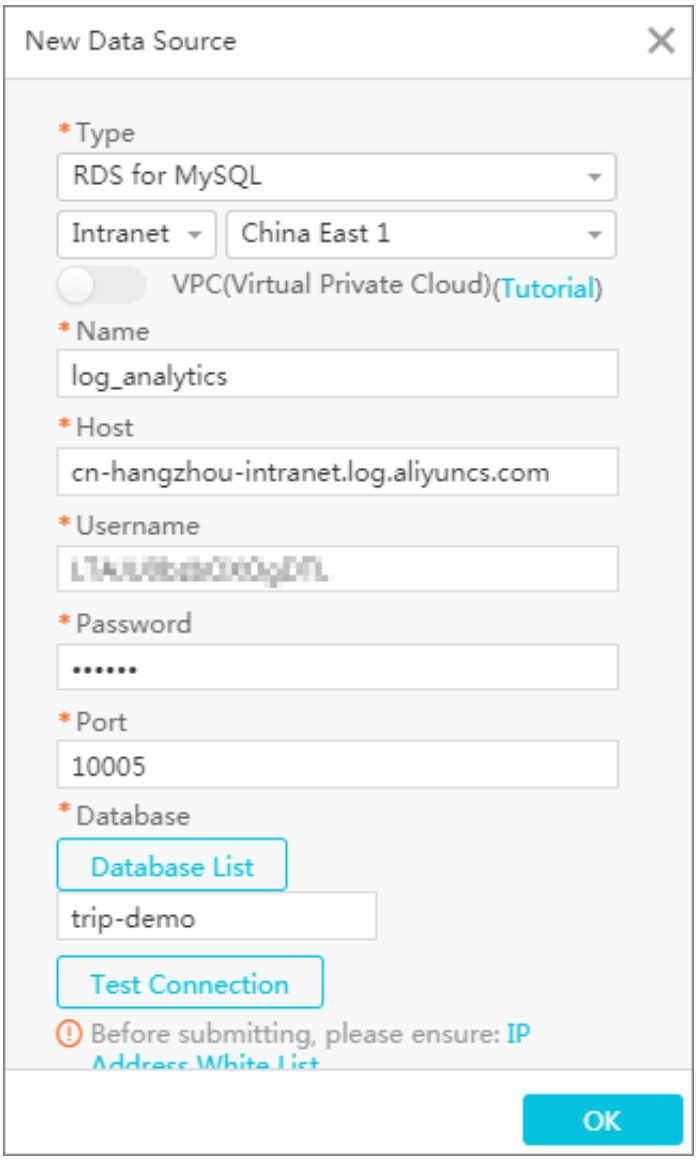
1. **Create data sources**

   You can select MySQL for RDS or Log Service as a data source as per your needs. The following section uses the MySQL protocol as an example to describe how to connect to Log Service.

   As shown in the figure, select the corresponding region and the intranet, and enter an AccessKey for the username and password. The AccessKey can be of a main

account or a sub-account that has the read permission to Log Service. Set the port number to 10005 and the database name to the project name.

Figure 7-23: Editing data

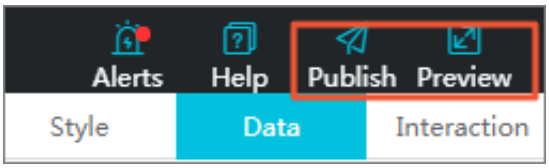


2. Creates a view.

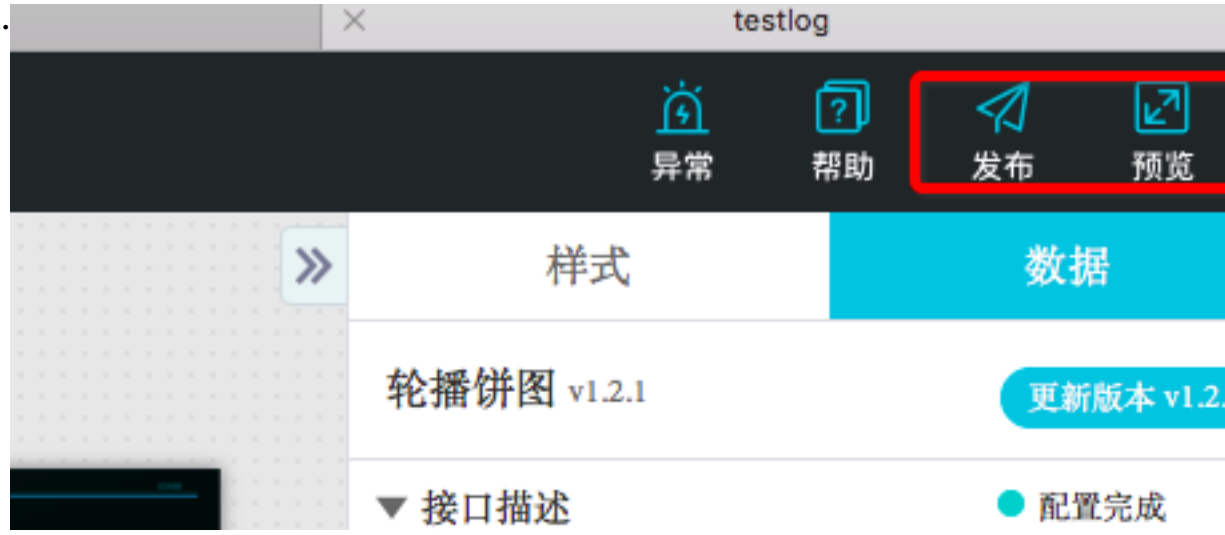
   Select a service template in the view and click any view on the large screen. Right-click the view to modify data or the data source of the view.

   As shown in the figure, select the database created in preceding steps as the data source, enter the queried SQL, and enter mappings between the query results and view fields in preceding field mappings.

   Figure 7-24: Select the database



3. Preview the view and publish.



Click the Preview button to preview the view.

Figure 7-25: Preview

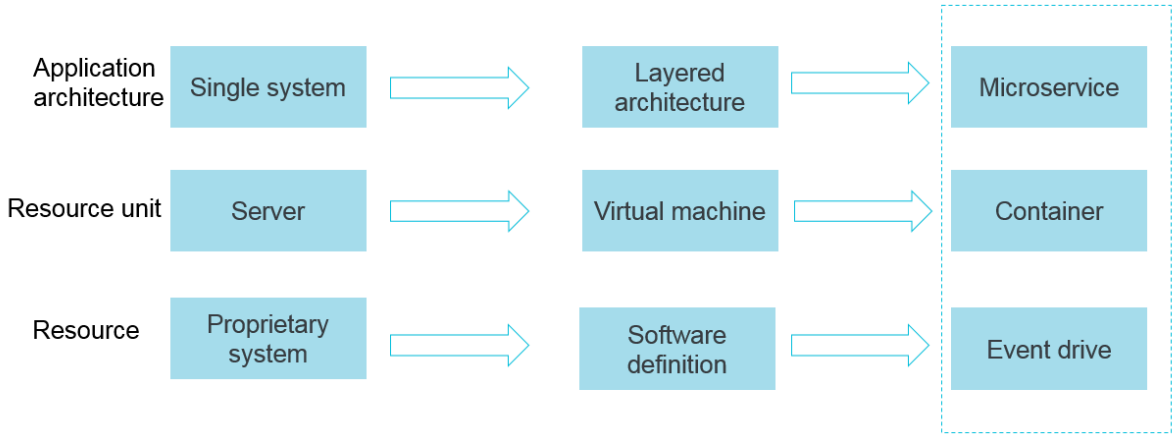


7.3.3 OpenTracing implementation of Jaeger

The advent of containers and serverless programming methods greatly increased the efficiency of software delivery and deployment. In the evolution of the architecture, there have been two changes:

- The application architecture is changing from a single system to microservices. Then, the business logic changes to the call and request between microservices.
- In terms of resources, traditional physical servers are fading out and changing to the invisible virtual resources.

Figure 7-26: Architectural Evolution



The preceding two changes show that behind the elastic and standardized architecture, the Operation & Maintenance (O&M) and diagnosis requirements are becoming more and more complex. To respond to these changes, a series of DevOps-oriented diagnosis and analysis systems have emerged, including the centralized log system (logging), the centralized measurement system (metrics), and the distributed tracing system (tracing).

Logging, metrics, and tracing

See the following characteristics of logging, metrics, and tracing:

- Logging is used to record discrete events.

  For example, the debugging or error information of an application, which is the basis of problem diagnosis.
- Metrics is used to record data that can be aggregated.

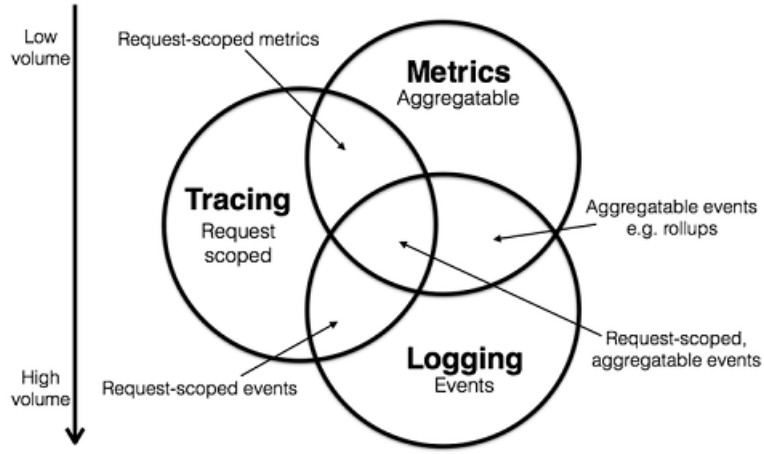
  For example, the current depth of a queue can be defined as a metric and updated when an element is added to or removed from the queue. The number of HTTP

requests can be defined as a counter that accumulates the number when new requests are received.

- Tracing is used to record information within the request scope.

For example, the process and consumed time for a remote method call, This is the tool we use to investigate system performance issues. Logging, metrics, and tracing have overlapping parts as follows.

Figure 7-27: Logging, metrics and tracing



We can use this information to classify existing systems. For example, Zipkin focuses on tracing. Prometheus begins to focus on metrics and may integrate with more tracing functions over time, but has little interest in logging. Systems such as ELK and Alibaba Cloud Log Service begin to focus on logging, continuously integrate with features of other fields, and are moving to the center in the preceding figure.

For more information about the relationship among *Metrics, tracing, and logging*. In what follows, we will introduce tracing systems.

Technical background of tracing

Tracing technology has been existing since the 1990s. However, the article "Dapper, a Large-Scale Distributed Systems Tracing Infrastructure" of Google brings this field into the mainstream. The more detailed analysis of sampling is in the article "Uncertainty in Aggregate Estimates from Sampled Distributed Traces". After these articles were published, a group of excellent tracing software programs were developed.

Among them, the popular ones are as follows:

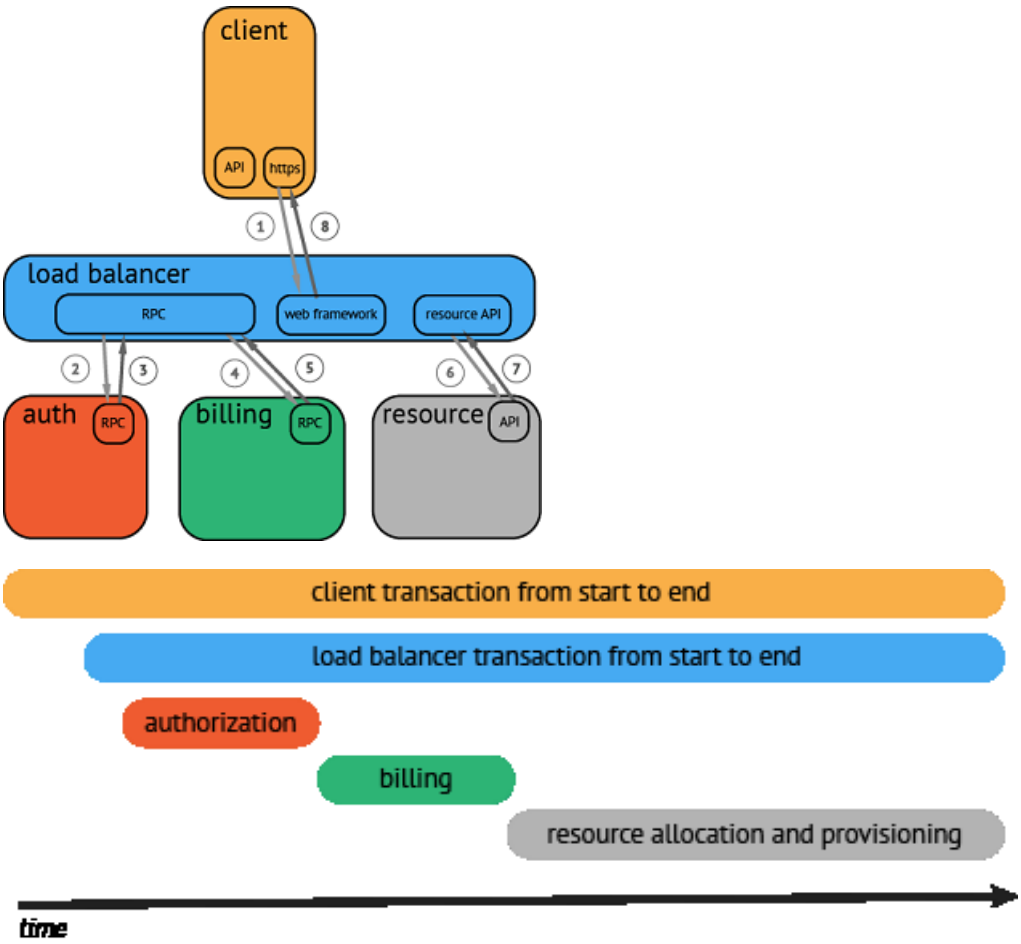
- Dapper (Google): Foundation for all tracers

- StackDriver Trace (Google)
- Zipkin (Twitter)
- Appdash (golang)
- EagleEye (Taobao)
- Ditecting (Pangu, the tracing system used by Alibaba Cloud cloud products)
- Cloud Map (Ant tracing system)
- sTrace (Shenma)
- X-ray (AWS)

Distributed tracing systems have developed rapidly, with many variants. However, they generally have three steps: code tracking, data storage, and query display.

An example of a distributed call is given in the following figure. When the client initiates a request, it is first sent to the load balancer and then passes through the authentication service, billing service, and finally to the requested resources. Finally, the system returns a result.

Figure 7-28: Example of distributed calls

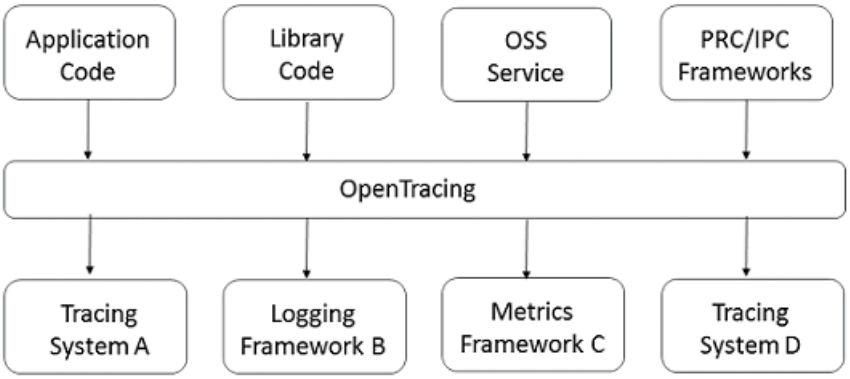


After the data is collected and stored, the distributed tracing system generally presents the traces as a timing diagram containing a timeline. However, in the data collection process, generally you have to make great changes to switch the tracing system because the system has to intrude on user codes and APIs of different systems are not compatible.

OpenTracing

The *OpenTracing* specification is created to address the problem of API incompatibility between different distributed tracing systems. OpenTracing is a lightweight standardization layer and is located between applications/class libraries and tracing or log analysis programs.

Figure 7-29: OpenTracing



Advantages

- OpenTracing already enters CNCF and provides unified concept and data standards for global distributed tracing systems.
- OpenTracing provides APIs with no relation to platforms or vendors, which allows developers to conveniently add (or change) the implementation of tracing system.

data model

In OpenTracing, a trace (call chain) is implicitly defined by the span in this call chain . Specifically, a trace (call chain) can be considered as a directed acyclic graph (DAG) composed of multiple spans. The relationship between spans is called References.

For example, the following trace is composed of eight spans:

```
In   a   single   trace ,  there   are   causal   relationsh  ips
between   spans
       [ Span   A ] ←←←( the   root   span )
            |
    +------+------+
```

```
    |             |
 [ Span   B ] [ Span   C ] ←←←( Span   C   is   a   child   node   of
  Span   A ,  ChildOf )
    |             |
[ Span   D ] +---+-------+
             |           |
          [ Span   E ] [ Span   F ] >>> [ Span   G ] >>> [ Span   H ]
                                        ↑
                                        ↑
                                        ↑
                        ( Span   G   is   called   after   Span   F
,  FollowsFro  m )
```

Sometimes, as shown in the following example, a timing diagram based on a timeline can better display the trace (call chain):

```
 The   time   relationsh  ip   between   spans   in   a   single
 trace .
--|-------|-------|-------|-------|-------|-------|-------|->  time
 [ Span   A ·····································]
   [ Span   B ·····························]
      [ Span   D ·····················]
    [ Span   C ·····················]
         [ Span   E ·····] [ Span   F ··] [ Span   G ··] [ Span   H
 ··]
```

each SPAN contains the following states:

- An operation name.
- A start timestamp.
- A finish timestamp.
- Span tag. A collection of span tags composed of key-value pairs. In a key-value pair , the key must be a string and the value type can be string, boolean, or numeric value.
- Span log. A collection of span logs. Each log operation contains one key-value pair and one timestamp. In a key-value pair, the key

must be a string and the value can be of any type. However, note that not all tracers that support OpenTracing must support all value types.

- SpanContext. The context of the span.
- References (relationship between spans). Zero or multiple related spans (establish the relationship between spans based on the SpanContext).

Each SpanContext contains the following statuses:

- Any OpenTracing implementation must transmit the current call chain status (for example, trace and span IDs) across process boundaries based on a unique span.

- Baggage items, which are the data that accompanies a trace. This is a collection of key-value pairs stored in a trace and also must be transmitted across process boundaries.

For more information about OpenTracing data models, see the OpenTracing semantic standards.

Function implementation

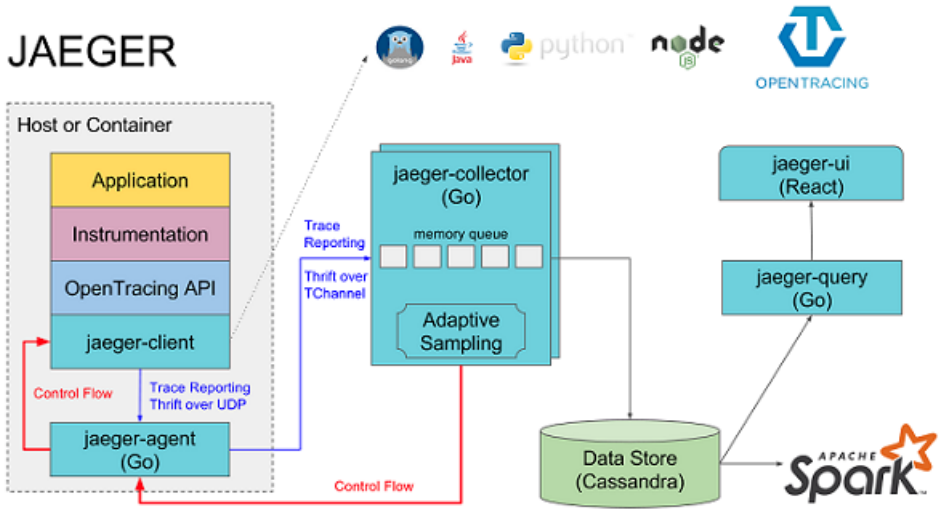
Supported tracer implementations lists all OpenTracing implementations. Of these implementations, *Jaeger* and *Zipkin* are the most popular ones.

Jaeger

Jaeger is an open-source distributed tracing system released by Uber. It is compatible with OpenTracing APIs.

Architecture

Figure 7-30: Architecture



As shown in the preceding figure, Jaeger is composed of the following components.

- Jaeger client: Implements SDKs that conform to OpenTracing standards for different languages. Applications use the API to write data. The client library transmits trace information to the jaeger-agent according to the sampling policy specified by the application.
- Agent: A network daemon that monitors span data received by the UDP port and then sends the data to the collector in batches. It is designed as a basic component

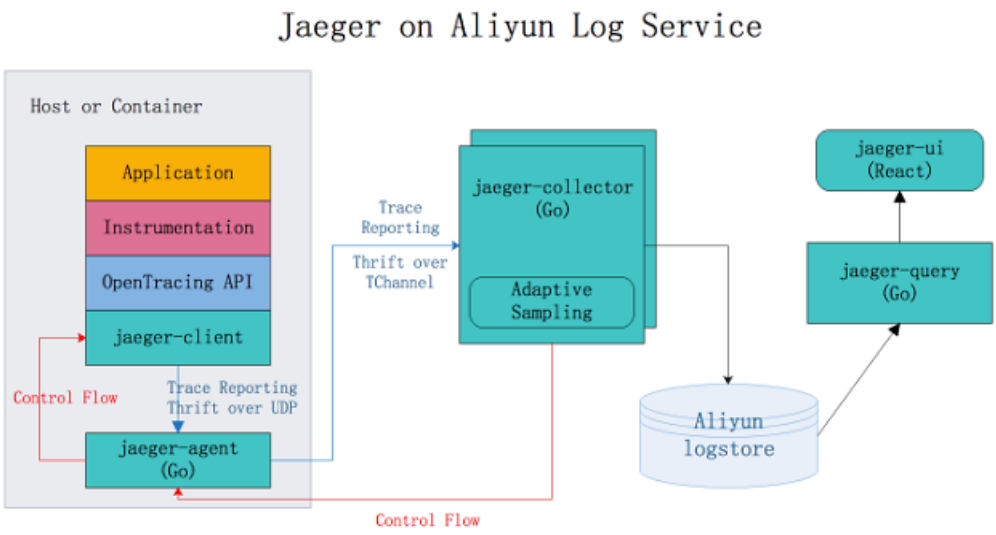
and deployed to all hosts. The agent decouples the client library and collector, shielding the client library from collector routing and discovery details.

- Collector: Receives the data sent by the jaeger-agent and then writes the data to backend storage. The collector is designed as a stateless component. Therefore, you can simultaneously run an arbitrary number of jaeger-collectors.
- Data store: The backend storage is designed as a pluggable component that supports writing data to Cassandra and Elasticsearch.
- Query: Receives query requests, retrieves trace information from the backend storage system, and displays it in the UI. Query is stateless and you can start multiple instances. You can deploy the instances behind load balancers such as Nginx.

Jaeger on Alibaba Cloud Log Service

is based on Jaeger, which allows you to persistently store collected tracing data in Log Service, and query and display the data by using the native Jaeger interfaces.

Figure 7-31: Jaeger



Advantages

- The native Jaeger only supports the persistent storage of data to Cassandra and Elasticsearch. You must maintain the stability of the backend storage system and adjust the storage capacity on your own. Jaeger on Alibaba Cloud Log Service leverages the capability of Log Service to process massive volumes of data and allows you to enjoy the convenience in the distributed tracing field without worrying about backend storage system problems.

- The Jaeger UI only provides query and trace display functions, without sufficient support for problem analysis and troubleshooting. By using Jaeger on Alibaba Cloud Log Service, you can take advantage of the powerful query and analysis capabilities of Log Service to quickly analyze the problems in the system.
- Compared to Jaeger using Elasticsearch as the backend storage, Log Service supports the Pay-As-You-Go billing method and the cost is only 13% of the Elasticsearch cost. For more information, see Compare LogSearch/Analytics with ELK in log query and analysis.

serviceLatency The backend latency (in milliseconds). Procedure

For more information, see *GitHub*.

Configuration example

HotROD is an application composed of multiple microservices and uses the OpenTracing API to record trace information.

Follow these steps to use Jaeger on Alibaba Cloud Log Service to diagnose problems in HotROD. The video contains the following:

1. Configure Log Service.
2. Run Jaeger by running the docker-compose command.
3. Run HotROD.
4. Use the Jaeger UI to retrieve the specified trace information.
5. Use the Jaeger UI to view detailed trace information.
6. Use the Jaeger UI to locate application performance bottlenecks.
7. Use the Log Service console to locate the performance bottleneck of the applicatio n.
8. The application calls the OpenTracing API.

Configuration tutorial

http://cloud.video.taobao.com//play/u/2143829456/p/1/e/6/t/1/50081772711.mp4

You can use the following query statements in this example:

- Count the average latency and request counts of frontend service HTTP GET/ dispatch operations every minute.

```
process . serviceNam  e : " frontend "  and   operationN  ame : "
HTTP   GET  / dispatch " |
select   from_unixt  ime (  __time__  -  __time__  %  60 )  as
time ,
```

```
truncate ( avg ( duration )/ 1000 / 1000 )  as   avg_durati  on_ms
,
count ( 1 )  as   count
group   by   __time__  -  __time__  %  60   order   by   time
desc   limit   60
```

- Compare the time consumed by the operations of two traces.

```
traceID : " trace1 "  or   traceID : " trace2 " |
select   operationN  ame ,
( max ( duration )- min ( duration ))/ 1000 / 1000   as
duration_d  iff_ms
group   by   operationN  ame
order   by   duration_d  iff_ms   desc
```

- Count the IP addresses of the traces whose latencies are more than 1.5 seconds.

```
process . serviceNam  e : " frontend "  and   operationN  ame : "
HTTP   GET  / dispatch "  and   duration  >  1500000000  |
select  " process . tags . ip "  as   IP ,
truncate ( avg ( duration )/ 1000 / 1000 )  as   avg_durati  on_ms
,
count ( 1 )  as   count
group   by  " process . tags . ip "
```

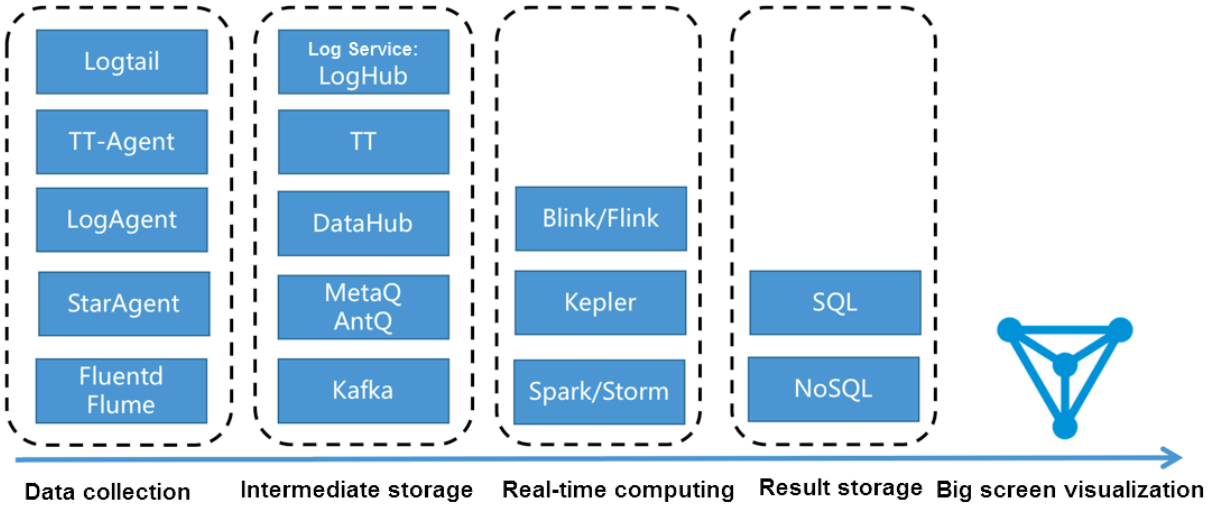
7.3.4 Interconnect with DataV big screen

People will think of the outstanding Tmall real-time big screen when talking about the Double 11 shopping campaign. The real-time big screen is impressive for its most typical stream computing architecture:

- Data collection: Collect data from each source in real time.
- Data collection: Collect data from each source in real time.
- Real-time computing: Subscribe to real-time data and compute data in windows by using the computing rules. This is the most important part in the process.
- Result storage: Store the computing results in SQL and NoSQL databases.
- Visualization: Call the results by using APIs for demonstration.

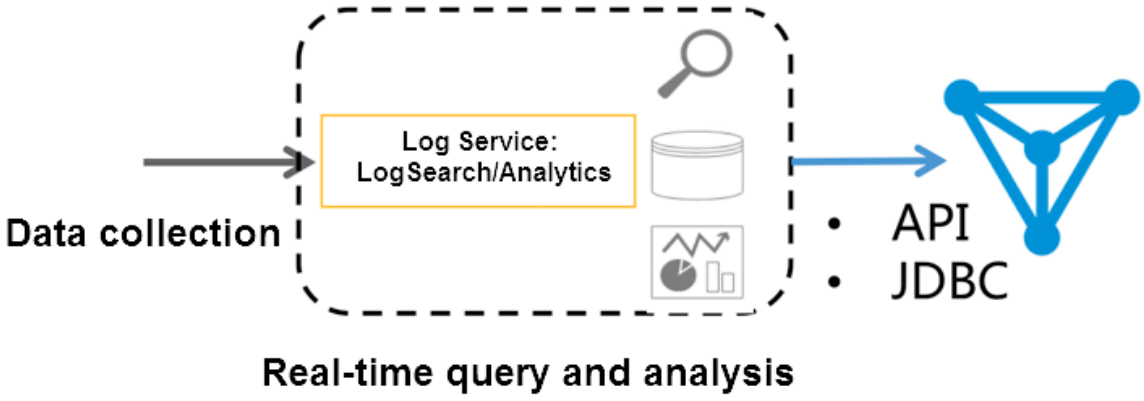
In Alibaba Group, many mature products can be used to complete such work. The following figure shows the products generally used.

Figure 7-32: Related products



Besides the preceding solution, you can also use the LogSearch/Analytics APIs of Log Service to directly interconnect with DataV to display data on a big screen.

Figure 7-33: Log Service + DataV



In September 2017, Log Service enhanced the real-time log analysis function (LogSearch/Analytics), which allows you to analyze logs in real time by using query and SQL92 syntax. Besides the built-in dashboard, Log Service supports the interconnection methods such as Grafana and Tableau (JDBC) to achieve result analysis visualization.

Features

Based on the data volume, timeliness, and business needs, computing is generally divided into two modes:

- Real-time computing (stream computing): Fixed computing + variable data.
- Offline computing (data warehouse + offline computing): Variable computing + fixed data.

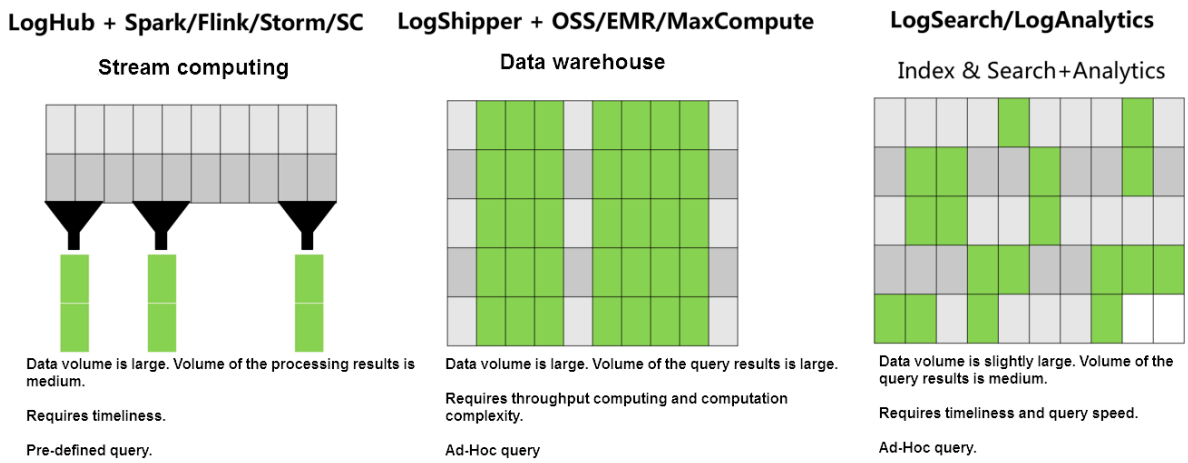
Log Service provides two interconnection methods to collect data in real time. In addition, in log analysis scenarios that has timeliness needs, LogHub data can be indexed in real time. Then, you can use LogSearch/Analytics to directly query and analyze data. This method has the following advantages:

- Fast: You can obtain the results immediately after query is passed in by using APIs, without waiting or pre-computing the results.
- Real-time: In 99.9% cases, the generated logs are displayed on the big screen within 1s.
- Dynamic: Whether statistic method modification or supplementary data, the display results are refreshed in real time, without waiting for recomputation.

However, no computing system is omnipotent. This method has the following limits:

- Data volume: Up to 10 billion GB data can be computed at a time. You must set the time limit if the data volume is exceeded.
- Computing flexibility: Currently, only the SQL92 syntax is supported for computing . Custom UDF is unsupported.

Figure 7-34: Log service advantage



Configuration process

Operation Demonstration:

To interconnect Log Service data with DataV big screen, follow these steps:

1. **Collect data. See *5-minute quick start* to access the data source to Log Service.**
2. **Set the index See *Index settings and visualization* or Use case for *website log analysis* in Best Practices.**
3. **Interconnect with the DataV plug-in to convert the real-time results queried by using the SQL statement to a view.**

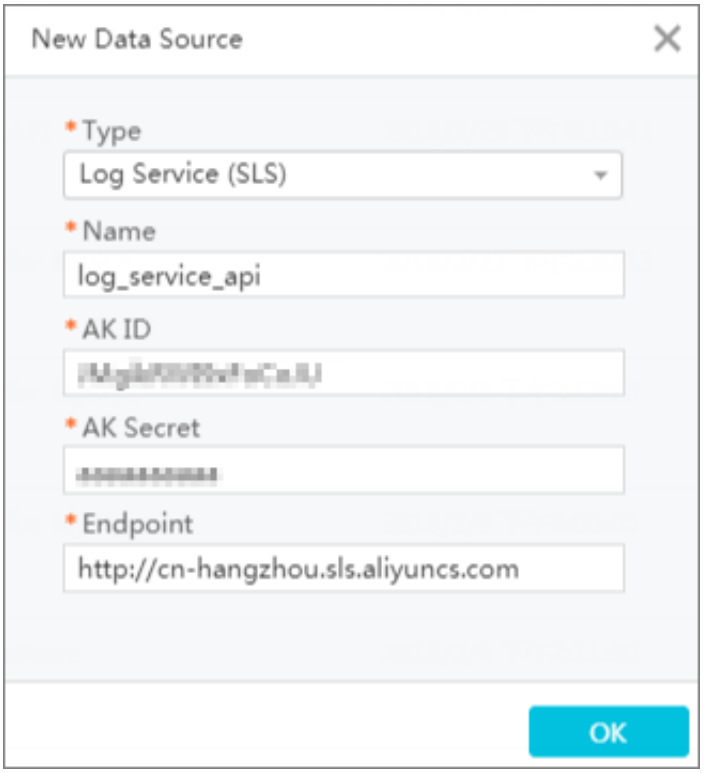
After completing steps 1 and 2, you can view the raw logs on the search page. This document mainly describes how to perform step 3.

Procedure

Step 1 Create a DataV data source

Click Data Sources in the left-side navigation pane. Click Add Source. The New Data Source dialog box appears. Enter the basic information of the data source. The following table describes the definition of each configuration item.

Figure 7-35: New data



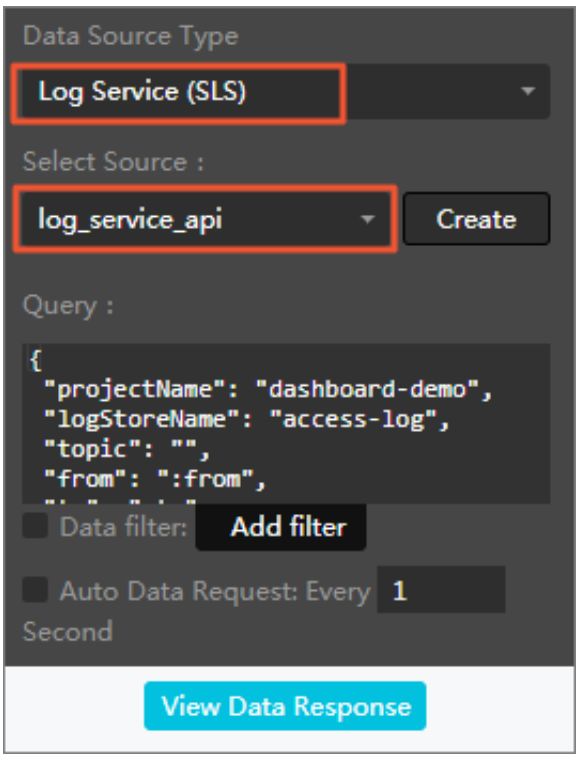
Configuration item	Description
Type	Select Log Service.
Name	Configure a name for the data source.
AK ID	The AccessKey ID of the main account, or the AccessKey ID of the sub-account that has the permission to read Log Service.
AK Secret	The AccessKey Secret of the main account, or the AccessKey Secret of the sub-account that has the permission to read Log Service.
Endpoint	The address of the region where the Log Service project resides. In the preceding figure, the address of region Hangzhou is entered.

Step 2 Create a line chart

1. Create a line chart.

   In the data configuration of the line chart, set the data source type to Log Service, select the data source log_service_api created in the previous step, and enter the parameters in the Query text box.

   Figure 7-36: Data source



   An example of the query parameters is as follows and the following table describes the parameters.

```
{
 " projectNam  e ": " dashboard - demo ",
 " logStoreNa  me ": " access - log ",
 " topic ": "",
 " from ": ": from ",
 " to ": ": to ",
 " query ": "*|  select   approx_dis  tinct ( remote_add  r )  as
   uv  , count ( 1 )  as   pv  ,  date_forma  t ( from_unixt  ime
 ( date_trunc (' hour ', __time__ ) ) ,'% Y /% m /% d  % H :% i :%
 s ')  as   time   group   by   time   order   by   time   limit
 1000 " ,
 " line ":  100 ,
 " offset ":  0
```

```
}
```

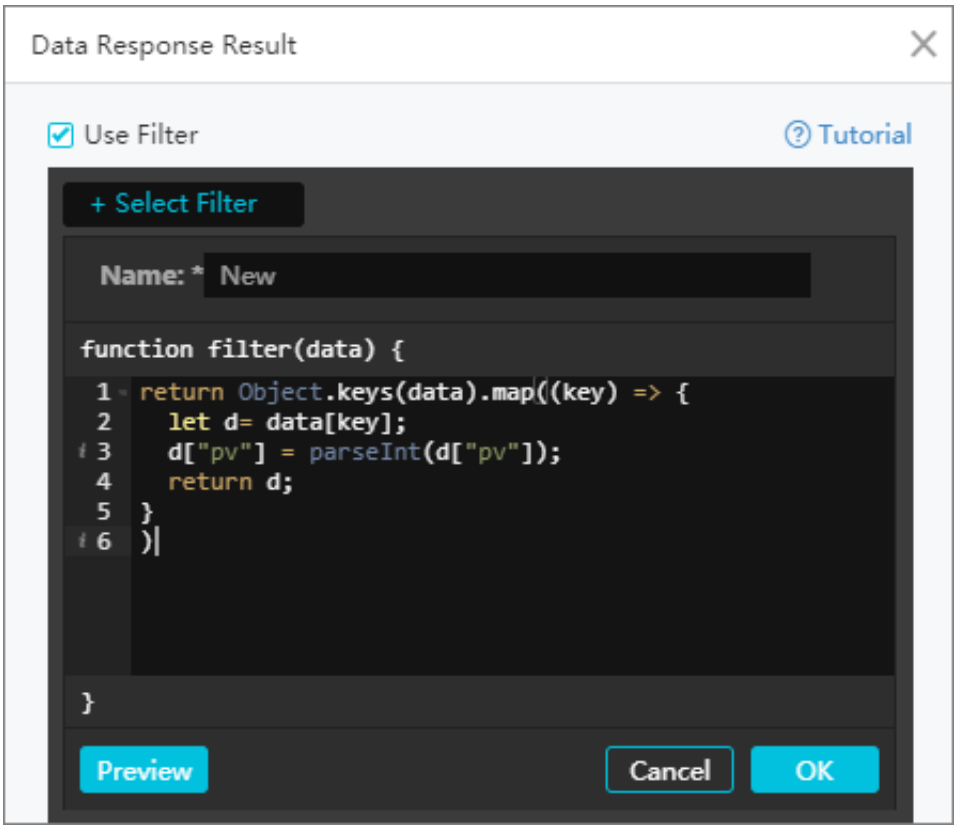
Configuration item	Description
projectName	The name of your project.
logstoreName	The name of your Logstore.
topic	Your log topic. If you have not set the topic, leave the parameter value empty.
from、to	from and to specify the start time and end time of the log respectively. Note: In the preceding example, the parameter values are respectively set to : `: from` and `: to`. During the test, you can enter the time in UNIX format, for example, 1509897600. After the release, convert the time to `: from` and `: to`, and set the specific time ranges of the values in the URL parameter. For example, the previewed URL is `http :// datav . aliyun . com / screen / 86312`. After `http :// datav . aliyun . com / screen / 86312 ? from = 1510796077 & to = 1510798877` is opened, the values are computed based on the specified time.

Configuration item	Description
query	Your query condition. In the preceding example, the query condition is the pv quantity per minute. For more information about the query syntax, see *Syntax description*. Note: The time in the query must be in the format like 2017/07/11 12:00:00. Therefore, use date_format(from_unixtime(date_trunc('hour',__ti %m/%d %H:%i:%s') to align the time on the hour, and then convert it to the target format. `date_forma t ( from_unixt ime ( date_trunc (' hour ', __time__ ) ) ,'% Y /% m /% d % H :% i :% s ')`
line	Enter the default value 100.

Configuration item	Description
offset	Enter the default value 0.

After the configurations, click View Data Response.

Figure 7-37: View Data Response



2. Create a filter.

   The Data Response Result dialog box appears after you click View Data Response. Select the Use Filter check box and click Select Filter > New Filter to create a filter.

   Enter the filter content in the following format:

```
 return   Object . keys ( data ). map (( key ) => {
 let   d =  data [ key ];
 d [" pv "] =  parseInt ( d [" pv "]);
 return   d ;
}
)
```

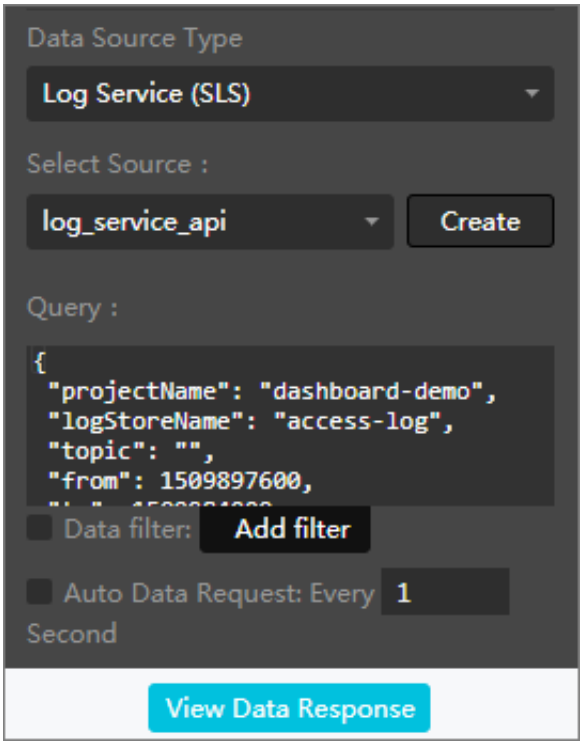
   In the filter, convert the result used by y-axis to the int type. In the preceding example, the y-axis indicates the pv. Therefore, the pv column must be converted.

   The results contain both the t and pv columns. You can set the x-axis to t and the y-axis to pv.

Step 3 Configure a pie chart

1. Create a carousel pie chart.

   Figure 7-38: Query text box

   Enter the following contents in the Query text box:

   ```
   {
    " projectNam  e ": " dashboard - demo ",
    " logStoreNa  me ": " access - log ",
    " topic ": "",
    " from ":  1509897600 ,
    " to ":  1509984000 ,
    " query ": "*|  select   count ( 1 )  as   pv  , method   group
    by   method " ,
    " line ":  100 ,
    " offset ":  0
   }
   ```

   During the query, the ratios of different methods can be computed.

2. Add a filter and enter the following contents in the filter:

   ```
    return   Object . keys ( data ). map (( key ) => {
    let   d =  data [ key ];
    d [" pv "] =  parseInt ( d [" pv "]);
    return   d ;
   }
   )
   ```

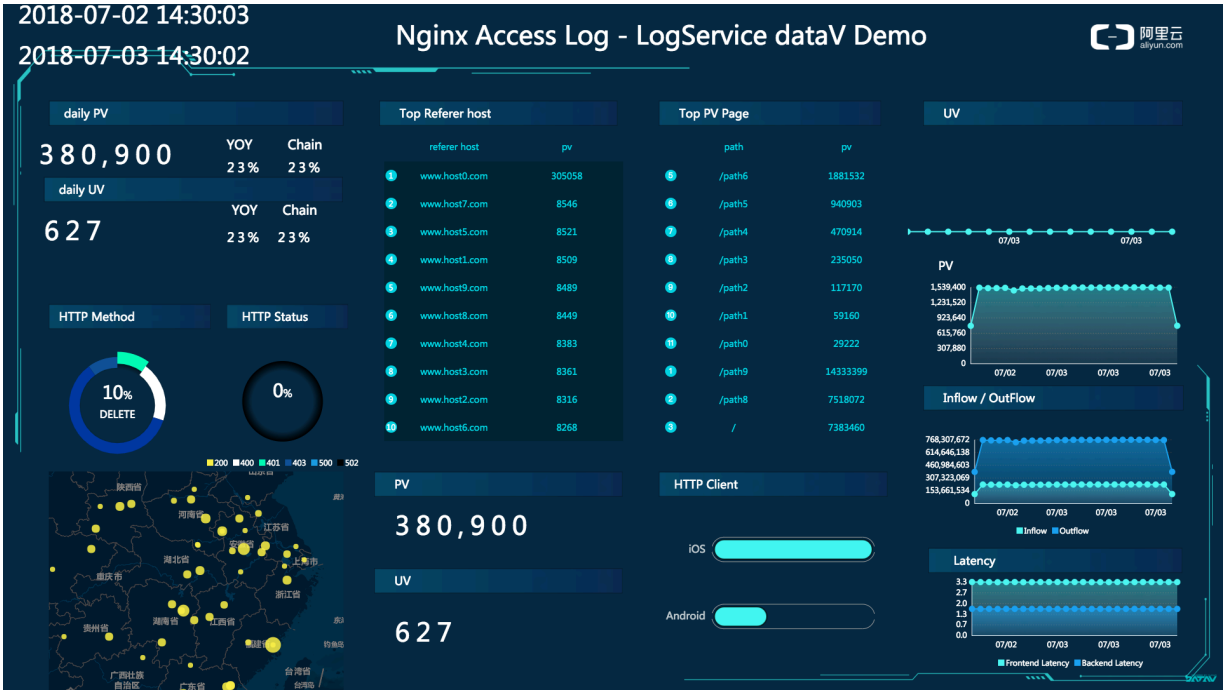
   Enter method in the type text box and pv in the value text box for the pie chart.

Step 4 Preview and release

Click Preview and Publish to create a big screen. Developers and business personnel can view their business access conditions in real time in the Double 11 shopping campaign.

Trial: *Demo*. You can set the values of the parameters from and to in the URL to any time.

Figure 7-39: Real-Time Screen



Use case: Continuously adjust the real-time big screen under the statistic criteria

For example, a temporary requirement is raised during the Computing Conference, which is to count the online (website) traffic across China. Total log data collection is configured and LogSearch/Analytics is enabled in Log Service. Therefore, you only need to enter your query condition.

1. For example, to count the UV, obtain the unique count of the forward field under Nginx in all access logs from October 11 to the present.

```
* |  select   approx_dis  tinct ( forward )  as   uv
```

2. After the system runs online for one day, the requirement is changed. Currently, only data under the domain yunqi needs to be counted. You can add a filter condition (host) for real-time query.

```
host : yunqi . aliyun . com  |  select   approx_dis  tinct (
forward )  as   uv
```

3. It is detected that the Nginx access logs contain multiple IP addresses. By default, only the first IP address is required. Therefore, process the query condition in the query.

```
host : yunqi . aliyun . com  |  select   approx_dis  tinct (
split_part ( forward ,',', 1 ))  as   uv
```

4. According to the requirement in the third day, the advertisement access in uc must be removed from access computing. In this case, you can add a filter condition not ... to obtain the latest result immediately.

```
host : yunqi . aliyun . com   not   url : uc - iflow  |  select
approx_dis  tinct ( split_part ( forward ,',', 1 ))  as   uv
```

7.3.5 Interconnection with Grafana

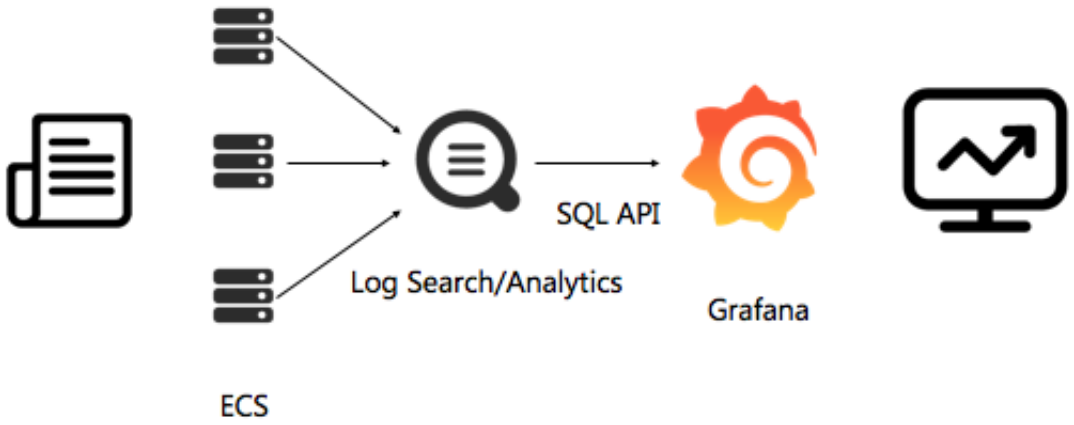
Alibaba Cloud Log Service is an all-in-one service for log-type data. Users can leave trivial jobs like data collection, storage computing interconnection, and data index and query to Log Service to focus on the analysis. In September 2017, Log Service upgraded the LogSearch/Analytics, allowing real-time log analysis using query + SQL92 syntax.

Besides the built-in Dashboard, interconnections with DataV, Grafana, Tableua, and Quick are also available to visualize the results analysis. This article demonstrates how to analyze and visualize Nginx logs using Log Service by giving an example of Grafana.

Process structure

Process from log collection to analysis is structured as follows.

Figure 7-40: Process structure



Procedure

1. Log data collection. For detailed procedures, see *5-minute quick start*.
2. Index settings and console query configuration. For detailed procedures, see *Overview* and *Collect and analyze Nginx access logs* or Website Log Analysis Case.
3. Install the Grafana plug-in to convert real-time query SQL into views.

After completing Step 1 and 2, you can view the raw log on the query page.

This document demonstrates Step 3.

Procedure

1. *Install Grafana*
2. *Install the Log Service plug-in*
3. *Configure the log data source*

4. *Add Dashboard*
    a. *Configure the template variables*
    b. *Configure PV and UV*
    c. *Configure inbound and outbound bandwidth*
    d. *Percentage of HTTP methods*
    e. *Percentage of HTTP status codes*
    f. *Page of top sources*
    g. *Pages of the maximum latency*
    h. *Top pages*
    i. *Top pages of non-200 requests*
    j. *Average frontend and backend latency*
    k. *Client statistics*
    l. *Save and release the dashboard*
5. *View the results*

1 Install Grafana

For detailed installation steps, see *Grafana official document*.

To Ubuntu, for example, the installation command is:

```
wget   https :// s3 - us - west - 2 . amazonaws . com / grafana -
releases / release / grafana_4 . 5 . 2_amd64 . deb
sudo   apt - get   install   - y   adduser   libfontcon  fig
sudo   dpkg  - i   grafana_4 . 5 . 2_amd64 . deb
```

To use the pie chart, you must install the pie chart plug-in. For detailed procedures, see *Grafana official document*.

The installation command is as follows:

```
grafana - cli   plugins   install   grafana - piechart - panel
```

2 install the Log service plug-in

Install the Log Service plug-in Confirm the directory location of Grafana plug-in. The location of Ubuntu plug-in is `/ var / lib / grafana / plugins /`. Restart the grafana-server after installing the plug-in.

Taking Ubuntu system as an example, run the following command to install the plug-in and restart the grafana-server.

```
cd  / var / lib / grafana / plugins /
git   clone   https :// github . com / aliyun / aliyun - log -
grafana - datasource - plugin
service   grafana - server   restart
```

3 Configure the log data source

For the deployment in a local machine, the default installation location is Port 3000. Open the Port 3000 in a browser.

1. Click on Grafana’s logo in the upper left corner and select Data Sources in the dialog box.
2. Click Add data source, and use Grafana and Alibaba Cloud Log Service for log visual analysis.
3. Complete the configuration items for the new data source.

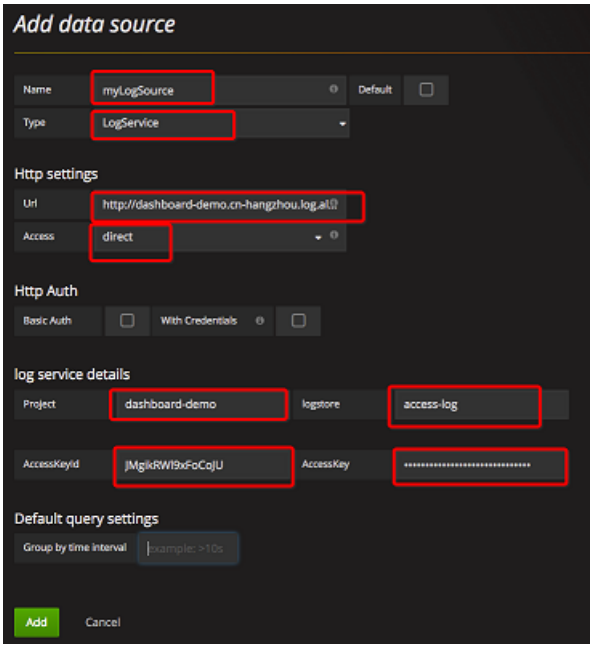
   Each part is configured as follows:

Configuration item	Configuration content
datasource	The name can be customized and the type is LogService.
Http Setting	URL input example: `http :// dashboard - demo . cn - hangzhou . log . aliyuncs . com` The `dashboard - demo` (project name) and `cn - hangzhou . log . aliyuncs . com` (endpoint of the project region) must be replaced with your project and region addresses when configuring your data source. You can select Direct or Proxy for Access.
Http Auth	Use the default configuration.

Configuration item	Configuration content
log service details	For detailed configuration of Log Service, enter Project, Logstore, and AccessKey that has the read permission respectively. AccessKey can belong to the primary account or the subaccount.

Configuration example

Figure 7-41: Configuration example



After the configuration is complete, click Add to add DataSource. Next, add Dashboard.

4 Add Dashboard

Click to open the menu in the upper left corner, select Dashboards and click New. Add a new Dashboard to the menu in the upper left corner.

4. 1 Configure the template variables

You can configure the template variables in Grafana to show different views in the same view by choosing different variable values. This document describes the configuration of each time interval and the access of different domain names.

1. Click the Settings icon at the top of the page, and then click Templating.

2. On the current page, the configured template variables are displayed. Click New to create a new template. First, configure a time interval.

   The name of the variable is the one you used in the configuration, which is called $myinterval here. You must write `$ myinterval` in the query condition. Please refer to the following table for configuration.

Configuration items	Configuration content
Name	The variable name, which you can name myinterval.
Type	Please select `Interval`
Lable.	Please enter `time interval`
Internal Options	Please enter `1m , 10m , 30m , 1h , 6h , 12h , 1d , 7d , 14d , 30d` in Value entry.

3. Configure a domain name template.

   Generally, multiple domain names can be mounted on one vps. You must view the accesses of different domain names. For the template value, enter `*, www . host . com , www . host0 . com , www . host1 . com` to view all the domain names, or view the accesses of `www . host . com` , `www . host0 . com` or `www . host1 . com` respectively.

   The domain name template configuration as below.

Configuration items	Configuration content
Name	Variable name, you can name it hostname.
Type	Please select `Custom`
Lable	Please enter a `domain name`
Custom Options	Please enter `*, www . host . com , www . host0 . com , www . host1 . com` forValue entry.

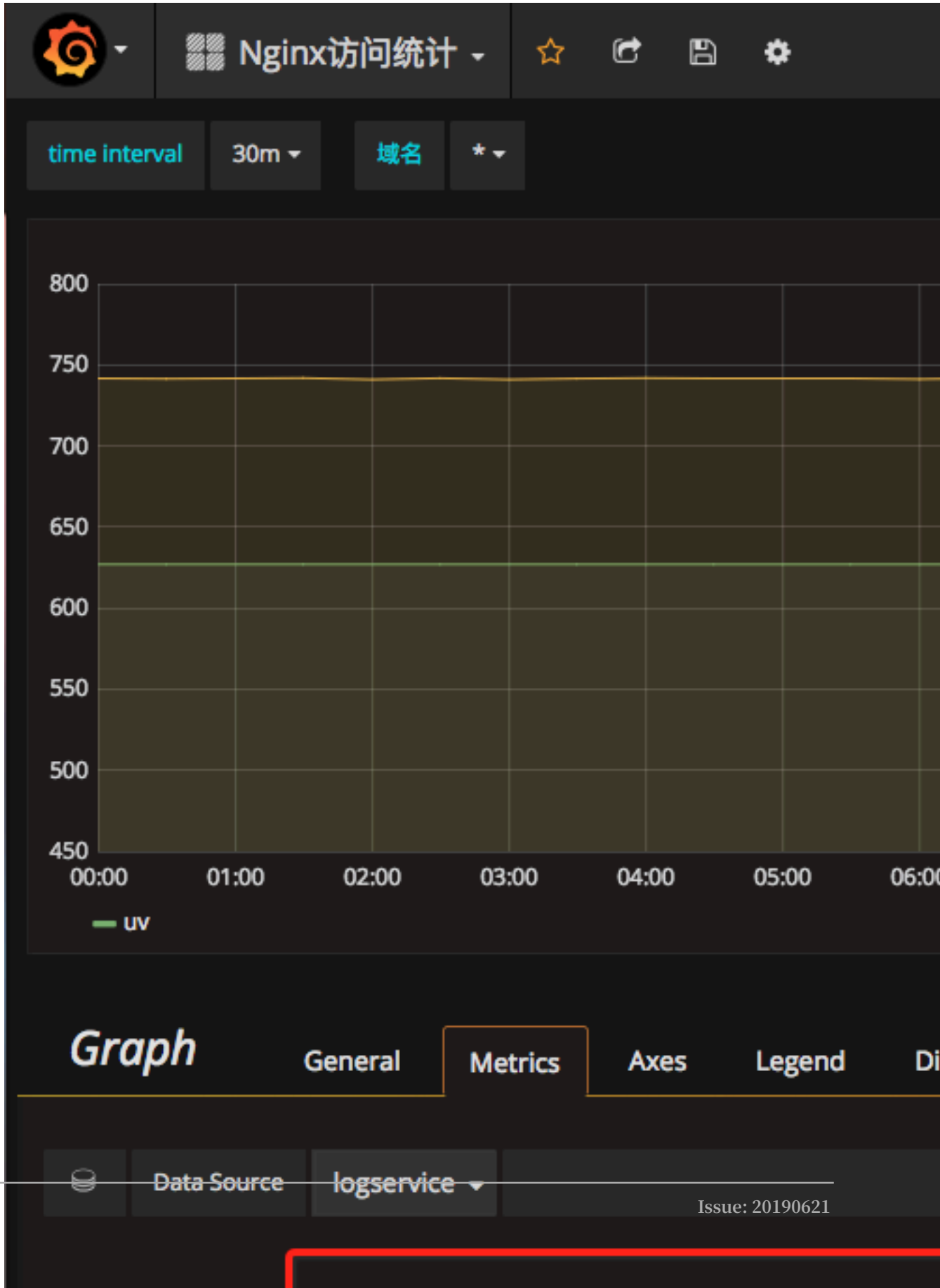
   After the configuration is complete, the template variables configured appears on the top of the Dashboard page. You can select any value from the drop-down box. For example, the following values can be selected for time interval.

4.2 Configure PV and UV

1. **Click ADD ROW on the left to create a new Row. If a Row already exists, you can select Add Panel in the left dialog box.**
2. **Grafana supports multiple types of views. For PV and UV data, create a Graphview.**
3. **Click Pannel Title and click Edit in the dialog box.**

4. In Metrics configuration, select `logservice` for datasource and enter Query, Y axis, and X axis:

Figure 7-42: Configure PV and UV

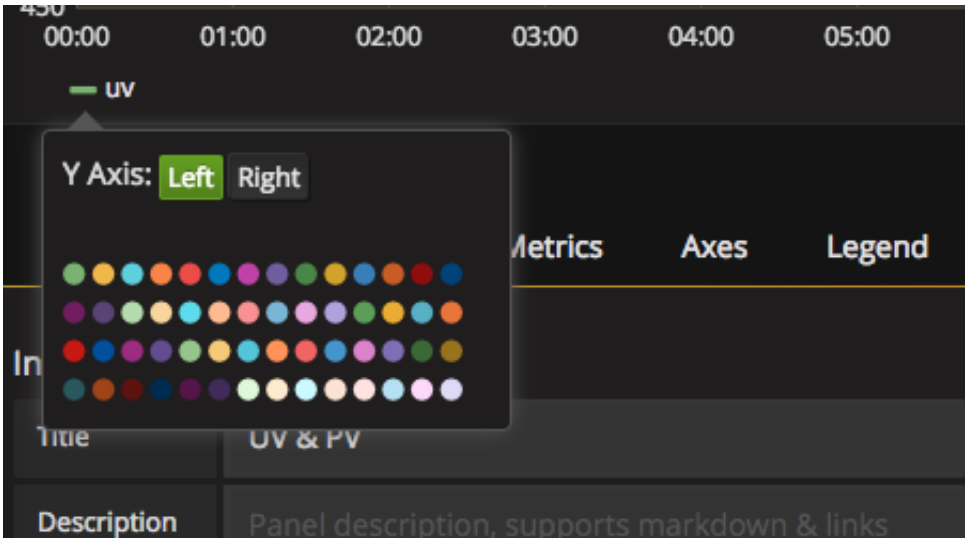


5. In the dataSource drop-down box, select the configured: `logservice`.

Configuration items	Configuration content
Query	`$ hostname | select approx_distinct(remote_addr) as uv, count(1) as pv, __time__ - __time__ % $$myinterval as time group by time order by time limit 1,000` The $hostname in the preceding Query is replaced with the domain name selected by the user for actual display. The $$myinterval is replaced with a time interval. Note that the number of $ before myinterval is two and for hostname is one.
X-Column	time
Y-Column	uv,pv

UV and PV values are displayed with two Y axes due to their large difference. Click the colored line on the left of UV below the icon to decide whether the UV is displayed on the left or the right Y axis:

Figure 7-43: Y-axis display



The default view for the title is Panel Title. To modify, open the General tab and enter a new title in the Title configuration item, such as `PV & UV`.

4.3 Configure inbound and outbound bandwidth

You can add inbound and outbound bandwidth traffic in the same way with *4.2 Configure PV and UV*.

The main configuration items are as follows:

Configuration items	Configuration content
Query	`$ hostname | select sum ( body_byte_ sent ) as net_out , sum ( request_le ngth ) as net_in , __time__ - __time__ % $$ myinterval as time group by __time__ - __time__ % $$ myinterval limit 10000`
X-Column	Time
Y-Column	net_in,net_out

4.4 Percentage of HTTP methods

You can configure the percentage of HTTP methods in the same way with *4.2 Configure PV and UV*.

Create a new Row, select Pie Chart , and enter Query, X axis, and Y axis in the configuration.

The main configuration items are as follows:

Configuration items	Configuration content
Query	`$ hostname | select count ( 1 ) as pv , method group by method`
X-Column	pie
Y-Column	method,pv

4.5 Percentage of HTTP status codes

You can configure the percentage of HTTP status codes in the same way with *4.2 Configure PV and UV*.

Create a new Row, and select Pie Chart for the view.

The main configuration items are as follows:

Configuration items	Configuration content
Query	`$ hostname | select count ( 1 ) as pv , status group by status`
X-Column	pie
Y-Column	status,pv

4.6 Page of top sources

You can configure the Page of top sources in the same way with *4.2 Configure PV and UV*.

Create a new Row, and select Pie Chart for the view:

The main configuration items are as follows:

Configuration items	Configuration content
Query	`$ hostname | select count ( 1 ) as pv , referer group by referer order by pv desc`
X-Column	pie
Y-Column	referer,pv

4.7 Pages of the maximum latency

You can configure the pages of the maximum latency in the same way with *4.2 Configure PV and UV*.

To show URL and its latency in a table, you must specify the view as Table at the time of creation.

The main configuration items are as follows:

Configuration items	Configuration content
Query	`$ hostname | select url as top_latenc y_url , request_ti me order by request_ti me desc limit 10`
X-Column	X-Column is left blank
Y-Column	top_latency_url,request_time

4.8 Top pages

You can add top pages in the same way with *4.2 Configure PV and UV*.

Create a new Table view. Data Source select logservice, query content, X-axis, and y-axis please refer to the following table.

Configuration items	Configuration content
Query	`$ hostname | select count ( 1 ) as pv , split_part ( url ,'?', 1 ) as path group by split_part ( url ,'?', 1 ) order by pv desc limit 20`
X-Column	X-Column is left blank
Y-Column	path,pv

4.9 Top pages of non-200 requests

You can add top pages of non-200 requests in the same way with *4.2 Configure PV and UV*.

Create a new Table view. Data Source select logservice, query content, X-axis, and y-axis please refer to the following table.

Configuration items	Configuration content
Query	`$ hostname not status : 200 | select count ( 1 ) as pv , url group by url order by pv desc`
X-Column	X-Column is left blank
Y-Column	url,pv

4.10 Average frontend and backend latency

You can add average frontend in the same way with *4.2 Configure PV and UV*.

Create a new Graph view: Data Source select logservice, query content, X-axis, and y-axis please refer to the following table.

Configuration items	Configuration content
Query	`$ hostname | select avg ( request_time ) as response_time , avg ( upstream_response_time ) as upstream_response_time , __time__ - __time__ % $$ myinterval as time group by __time__ - __time__ % $$ myinterval limit 10000`
X-Column	time
Y-Column	upstream_response_time,response_time

4.11 Client statistics

You can add client statistics in the same way with *4.2 Configure PV and UV*.

Create a new Pie Chart. Data Source select logservice, query content, X-axis, and y-axis please refer to the following table.

Configuration items	Configuration content
Query	`$ hostname | select count (1) as pv , case when regexp_lik e (http_user_ agent , ' okhttp ') then ' okhttp ' when regexp_lik e (http_user_ agent , ' iPhone ') then ' iPhone ' when regexp_lik e (http_user_ agent , ' Android ') then ' Android ' else ' unKnown ' end as http_user_ agent group by http_user_ agent order by pv desc limit 10`
X-Column	pie
Y-Column	http_user_agent,pv

4.12 Save and release the dashboard

Click the Save button at the top of the page to release the Dashboard.

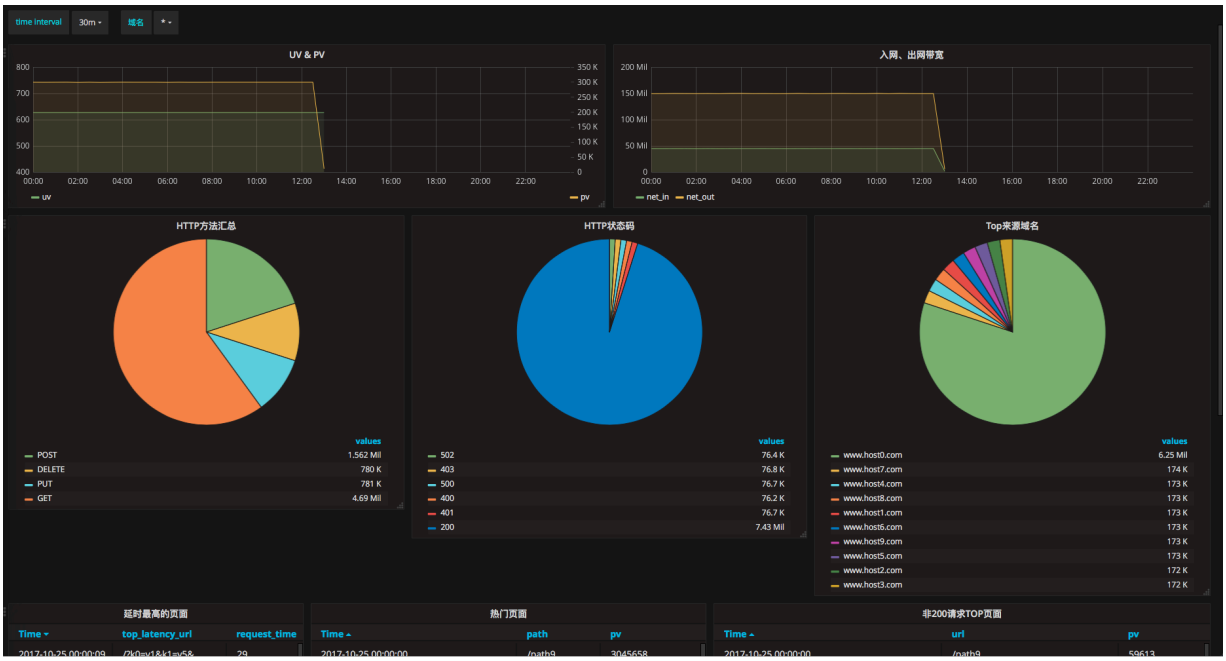
5 View the results

Open the home page of Dashboard to view the results. Demo address: *Demo*.

At the top of the page, you can choose the time range or time granularity of statistics, or you can choose a different domain name.

Now, the configuration of Dashboard for Nginx access statistics is completed, enabling you to mine valuable information from your views.

Figure 7-44: Check the result.



8 Alarm

8.1 Alarm function overview

This topic describes the alarm mechanism, alarm configuration limits, and the statements used by an alarm in typical scenarios. With the alarm function provided by Log Service, you can create an alarm and associate it with the charts in a dashboard to monitor logged services in real time.

Overview

An alarm is configured based on the data in specific charts in a dashboard. On the Search or Dashboard page of the Log Service console, you can configure an alarm. Specifically, you can set the condition for triggering an alarm and the alarm notifications. After you configure an alarm, Log Service checks the query results of the charts in a dashboard at specified intervals. If a query result meets the condition specified in your alarm rule, Log Service then sends an alarm notification. For more information, see *Configure an alarm*.

> **Note:**
> The alarm function has been upgraded in the Log Service. After the upgrade, you can retain an alarm of an earlier version because Log Service retains your alarm configuration information from before this upgraded version. However, we recommend that you upgrade all alarms of an earlier version to the latest version. For more information, see *Upgrade an alarm configuration to the latest version*.

Limits

Configuration	Description
Charts associated with an alarm	Each alarm must be associated with a chart and can be associated with up to three charts.

Configuration	Description
Condition	An expression (displayed as Trigger Condition in the console) must be 1 to 128 characters in length. · Only the first 100 log entries in the query output of a statement are analyzed to determine whether any log entries meet the the condition that you have set to trigger the alarm. · A condition can be used for up to 10,000 calculation.
Log entry character	The system can use up to 1024 characters of a log entry (output by a statement) for calculation.
Search period	Each search and analysis statement can at most search log data from a period of 24 hours at most.

Statements used by an alarm

Alarms are based on the data in charts in a dashboard. Each chart shows the search results of a query statement or a search and analysis statement.

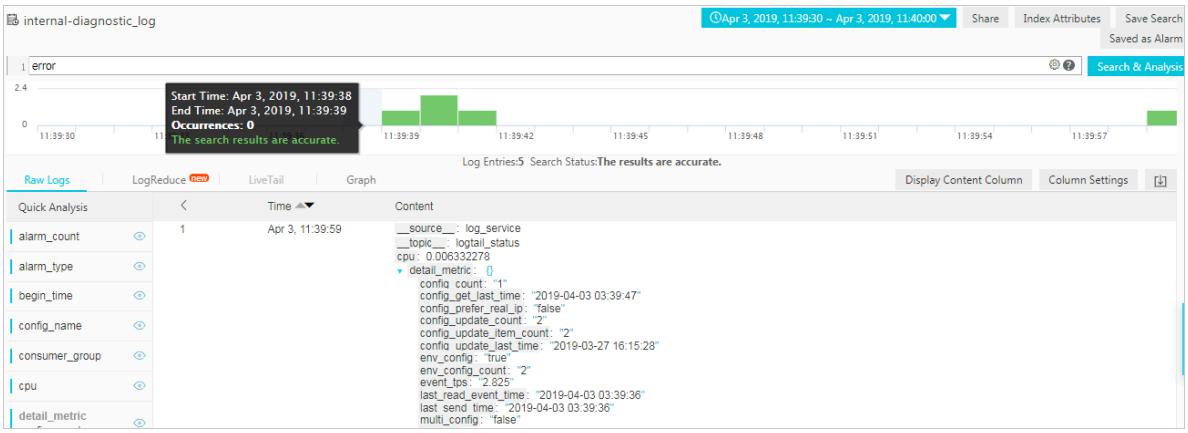
- If you use a query statement, the system outputs the log entries that meet the conditions of the query statement.
- If you use a search and analysis statement, the system collects the statistics of the log entries that meet the conditions of the statement and then outputs these statistics.
- Configure an alarm for the output of a query statement

  In this example, a query statement of error is used to query the log entries that contain the word error within the last fifteen minutes, and the system outputs 144 log entries. Each log entry consists of key-value pairs. For this example, you can set an alarm for the value of a key.

   Note:
  If the system outputs more than 100 log entries for a query statement, an alarm only analyzes the first 100 log entries. This means that the alarm can be triggered

only by log entries that meet the condition for triggering the alarm and also are among the first 100 log entries.

Figure 8-1: Query statement



- Configure an alarm for the output of a search and analysis statement

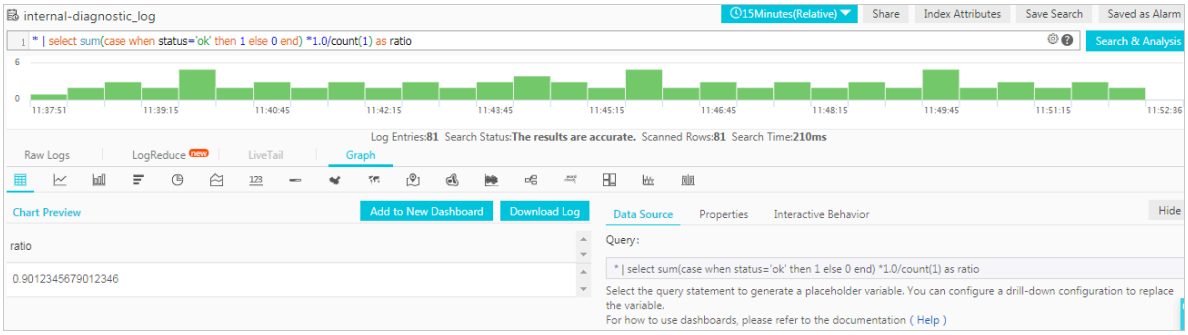
  In this example, the following search and analysis statement is used to obtain the ratio of the log entries with a status code of the OK format in all log entries:

```
* | select sum ( case when status =' ok ' then 1 else 0 end ) * 1 . 0 / count ( 1 ) as ratio
```

Note:

For more information, see *Query syntax*.

Figure 8-2: Search and analysis statement



For this example, you can configure an alarm by setting the condition to trigger the alarm as the `ratio < 0 . 9`. This means that the alarm is triggered when the ratio of the log entries with status codes of the OK format in all log entries drops below 90%.

8.2 Configure an alarm

8.2.1 Configure an alarm

This topic describes how to configure an alarm. After you configure an alarm, Log Service checks log data at specified intervals, and sends alarm notifications when the condition for an alarm to be triggered is met.

Prerequisites

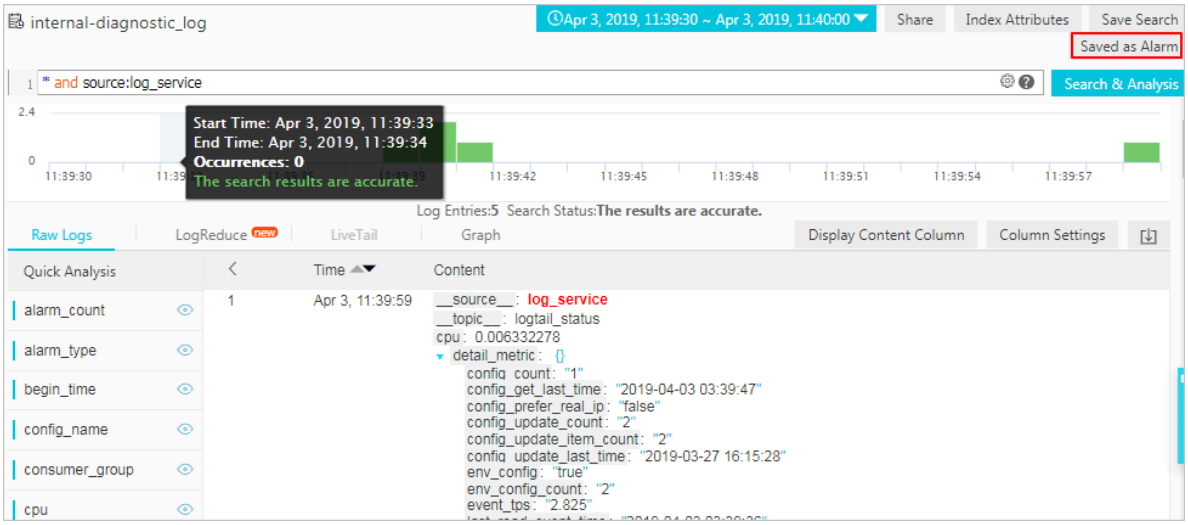
- Log data is collected.
- Log indexes are set. For more information, see *Enable and set indexes*.

Context

Alarms are configured based on charts. When you view a chart, you can create an alarm based on the chart by clicking Saved as Alarm. You can also create an alarm based on a chart in a dashboard by choosing More Option > Create Alert in the upper-right corner of the chart.

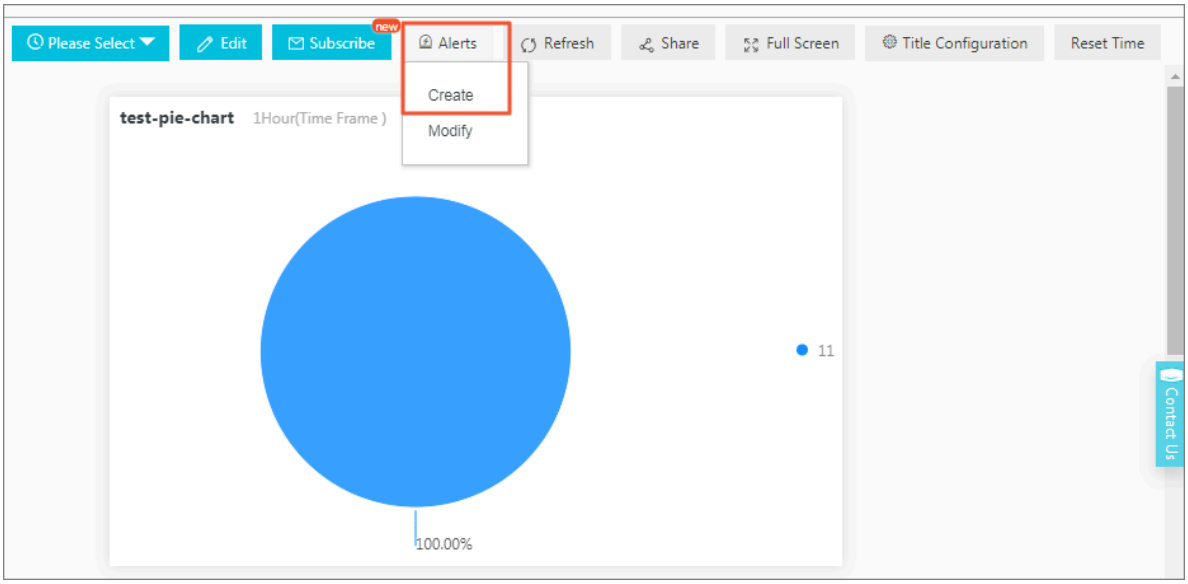
- Create a chart and configure an alarm for the chart

  On the Search page, enter a search and analysis statement in the search box, and then click Saved as Alarm. In this case, you can create an alarm, create a chart that indicates the search and analysis statement, associate the alarm with the chart, and set the dashboard where the chart is attached.



- Configure an alarm for the chart in a dashboard

  You can configure an alarm for one or more than one chart in a dashboard at a time . If you associate an alarm with multiple charts, you can add these charts to the condition for which an alarm is triggered.



The following shows the steps to configure an alarm for the chart in a dashboard.

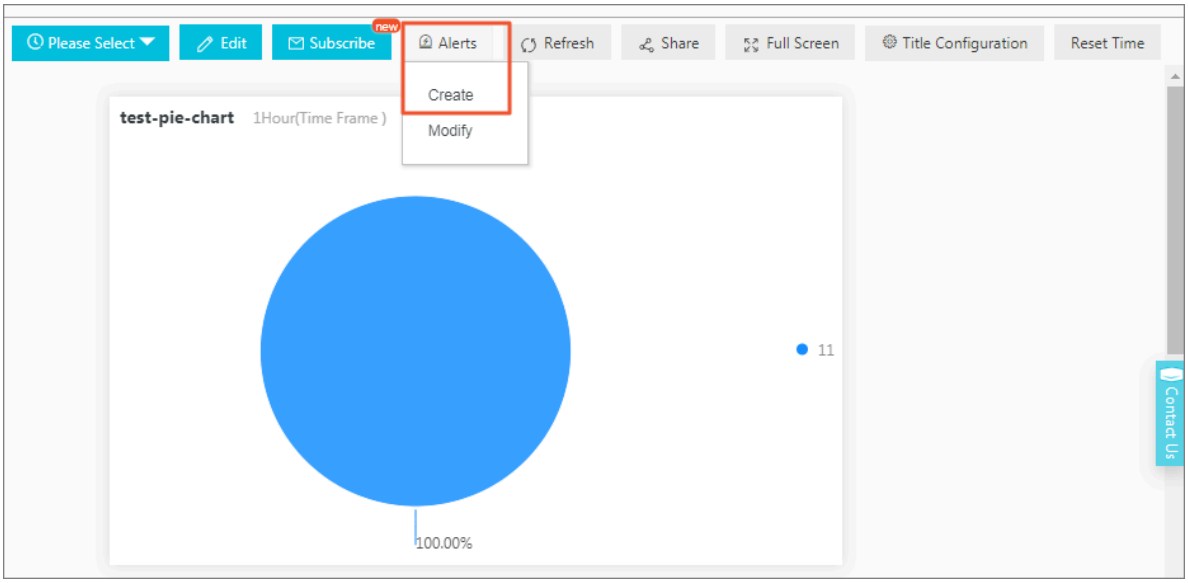
> Note:
> In a dashboard, if an update has been made to the search and analysis statement indicated by a chart that is associated with an alarm, you need to manually update the alarm. More specifically, you need to use the updated search and analysis statement to replace the search and analysis statement that was associated with the alarm. For more information, see *Modify an alarm configuration*.

For more information, see *Alarm configuration example*.

Procedure

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane, click Dashboard.
3. Click the target dashboard name.

4. In the upper-right corner of the page, choose Alert > Create.



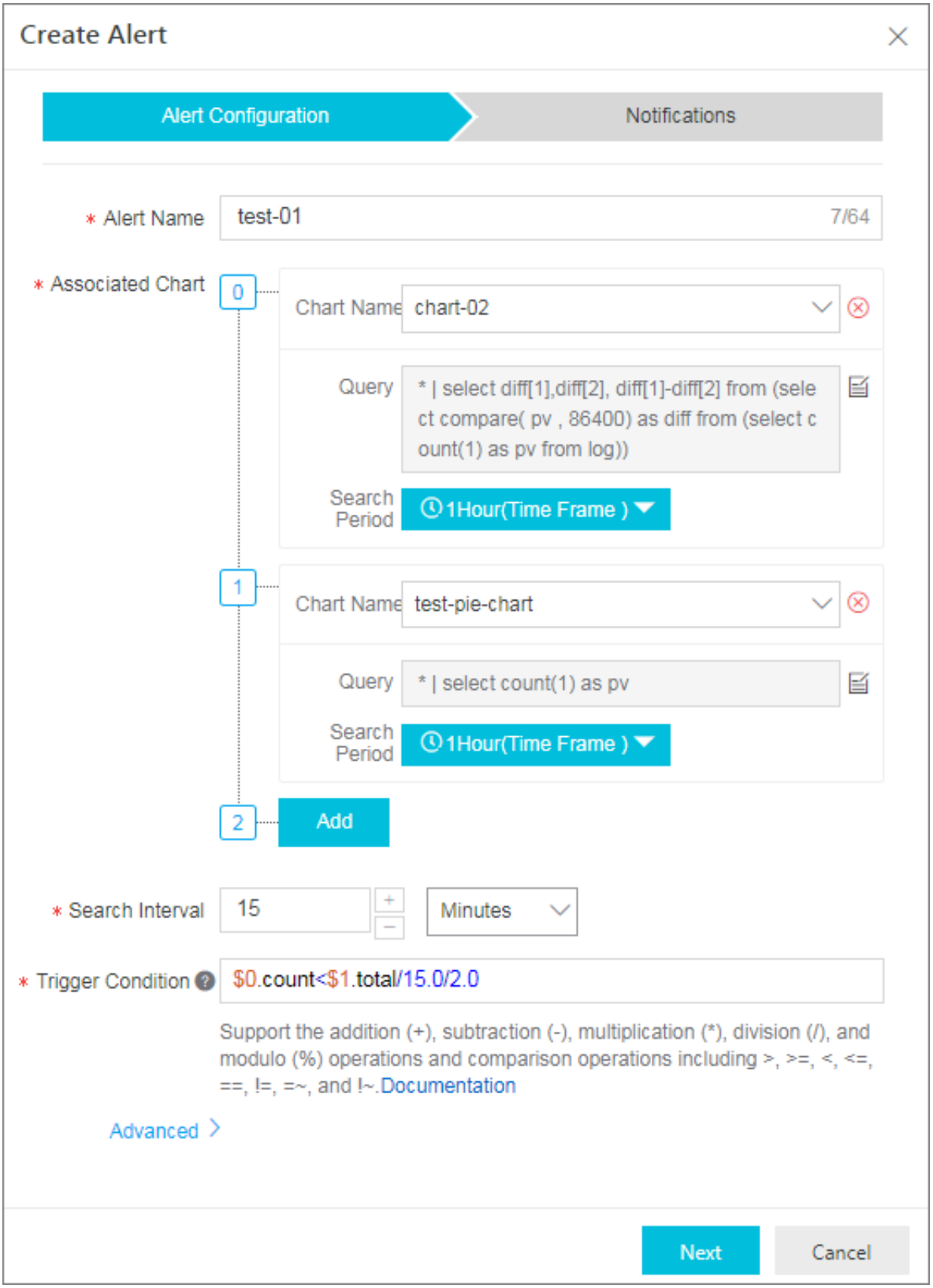
5. Set alarm parameters, and then click Next.

Configuration	Description
Alert Name	Set the name of an alarm. It must be 4 to 64 characters in length.

Configuration	Description
Associated Chart	Set the chart with which the alarm is associated with. Click Add, select a chart from the Chart Name drop-down list, and set a Search Period. Note: Search Period refers to the time range in which data is measured. For the time range, you can choose either Relative or Time Frame. If you set the parameter as 15Minutes(Relative) at 14:30:06, the system measures data with the period between 14:15:06- 14:30:06 relative to your time. However, if you set the parameter as 15Minutes(Time Frame) at 14:30:06, the system measures data with the exact period between 14:15:00- 14:30:00. To associate the alarm with multiple charts, add charts multiple times. Note: Each chart has a sequence number, which appears before the chart name. You can use the sequence number of a chart to associate the alarm triggering condition with the chart.
Search Interval	Set the intervals (in seconds, minutes, or hours) at which the system detects the output of the charts associated with the alarm. The minimum interval is 60 seconds, and the maximum interval is 24 hours. Note: Each time the system only detects the first one hundred log entries within the specified Search Period when the alarm is executed.

Configuration	Description
Trigger Condition	Enter an expression to set the condition for the alarm to be triggered. When the specified condition for the alarm to be triggered is met, the system sends an alarm notification according to your settings for Search Interval and Notification Interval. For example, you can set this parameter as `pv % 100 > 0 && uv > 0`. Note: You can use `$ sequence number` to indicate a chart in the condition expression. For example, if you set `$ 0`, this means that the chart that has the sequence number of 0. For more information, see *How to view a chart sequence number?*
Advanced	
Notification Trigger Threshold	Set the threshold for the number of times for which the specified condition has to be triggered for alarm notifications to be sent. When the threshold you specified is reached, the system checks whether the specified Notification Interval is reached to determine whether to send an alarm notification. The default value of this parameter is 1. That is, each time the specified Trigger Condition is met, the system checks whether the specified Notification Interval is reached. After an alarm notification is sent, the number of times counted by the system for when the condition was met is reset by the system. Network failures or other exceptions do not affect the overall count reported by the system.

Configuration	Description
Notification Interval	Set the intervals at which alarm notifications are sent. If the output of associated charts meets the specified condition for the alarm to be triggered, and the specified Notification Trigger Threshold and Notification Interval values are reached, then the system sends an alarm notification. For example, if you set this parameter as five minutes, you may receive up to one alarm notification within 5 minutes. The default value of this parameter is 0.  Note: Notification Trigger Threshold and Notification Interval help you control the number of alarm notifications that you receive for each event.


Create Alert
Alert Configuration
Notifications
* Alert Name
test-01
7/64
* Associated Chart
0
Chart Name
chart-02
Query
* | select diff[1],diff[2], diff[1]-diff[2] from (sele ct compare(pv , 86400) as diff from (select c ount(1) as pv from log))
Search Period
1Hour(Time Frame)
1
Chart Name
test-pie-chart
Query
* | select count(1) as pv
Search Period
1Hour(Time Frame)
2
Add
* Search Interval
15
Minutes
* Trigger Condition
$0.count<$1.total/15.0/2.0
Support the addition (+), subtraction (-), multiplication (*), division (/), and modulo (%) operations and comparison operations including >, >=, <, <=, ==, !=, =~, and !~.Documentation
Advanced
Next
Cancel

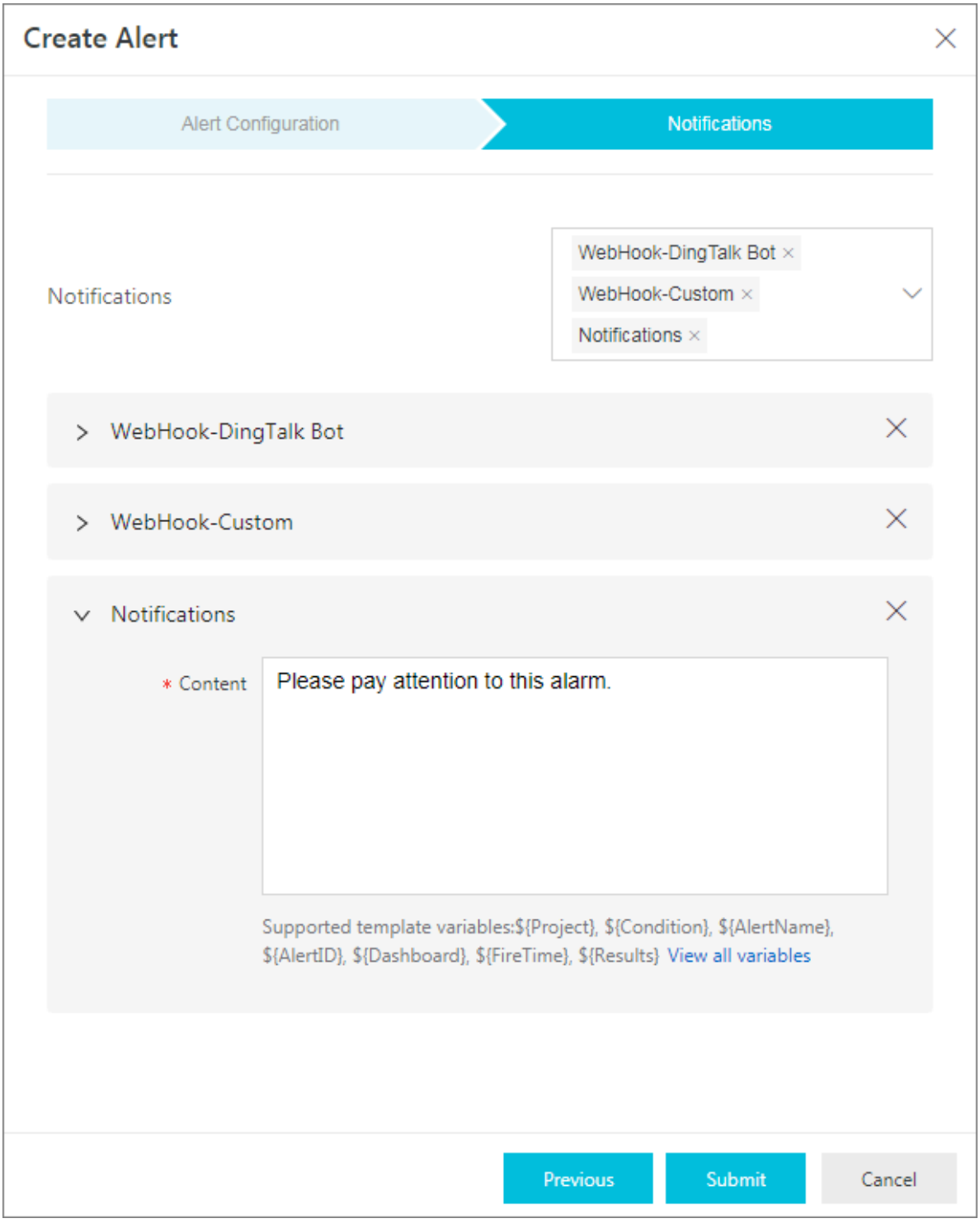
6. Set notification types.

   You can set one or more notification types, including DingTalk chatbot, WebHook, and Notification Center.

   For more information, see *Set alarm notifications*.

Alarm notification type	Description
DingTalk chatbot	Alarm notifications are sent as DingTalk messages to a specified DingTalk group. To use this type of alarm notification, you must set Request URL and Content. Content refers to the content of a DingTalk message. It must be 1 to 500 characters in length, and can contain template variables. For more information about how to configure a DingTalk chatbot and how to obtain a Request URL, see *Set alarm notifications*. Note: Each DingTalk chatbot sends up to 20 alarm notifications per minute.
WebHook	Alarm notifications are sent to a specified WebHook URL through a specified method. To use this type of alarm notification, you must set Request URL, Request Method, and Request Content. Available values of Request Method are GET, PUT, POST, DELETE, and OPTIONS. Content refers to the alarm notification content. It must be 1 to 500 characters in length, and can contain template variables.

Alarm notification type	Description
Message Center	Alarm notifications are sent to specified recipients through a method specified in Alibaba Cloud Message Center. To use this type of alarm notification, you must set Content. Content refers to the alarm notification content. It must be 1 to 500 characters in length, and can contain template variables. In addition, you need to set notification recipients and notification methods.



7. Click Submit.

What's next

Manage alarm settings or *View alarm records*.

8.2.2 Set alarm notifications

This topic describes how to set the DingTalk chatbot message, WebHook, and Message Center as the notification method for an alarm.

DingTalk

You can configure Log Service to send alarm notifications as DingTalk chatbot messages to the specified DingTalk group. Furthermore, you can specify group members to be reminded by an at (@).

 Note:
Each chatbot can send up to 20 alarm messages per minute.

Procedure

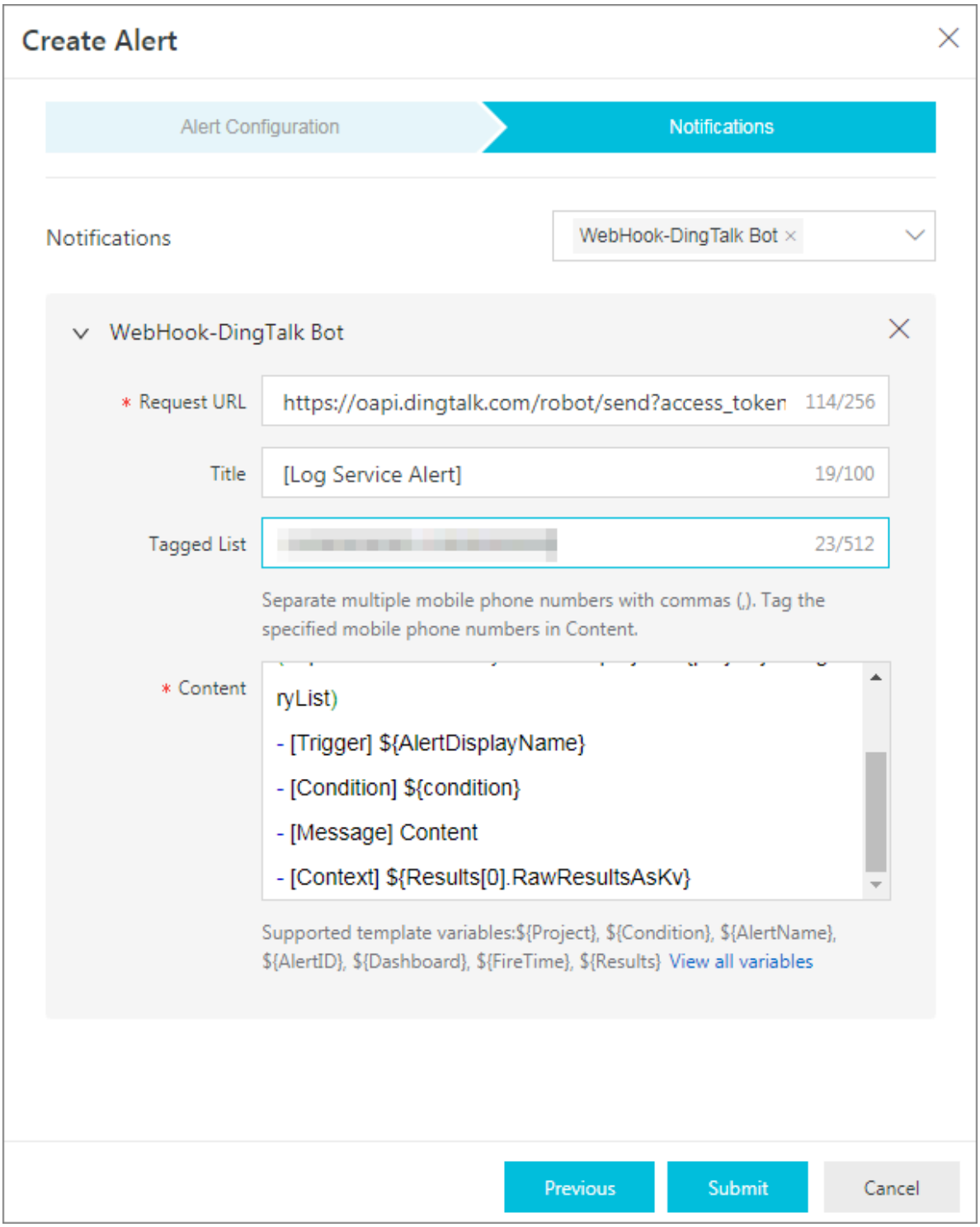
1. Open DingTalk and select the target DingTalk group.
2. In the upper-right corner, click Group Settings and click ChatBot.
3. Select Custom (Custom message services via webhook), and click Add.
4. Enter a ChatBot Name and click Finished.
5. Click Copy to copy the WebHook URL.
6. *Create an alarm* in the Log Service console, and then select WebHook-DingTalk Bot from the notification drop-down list.
7. In the Request URL box, paste the URL copied in *Step 5*. In the Tagged List box, enter the cell phone numbers of the group members to which you want to notification sent, and use commas (,) to separate different cell phone numbers.

8. **Edit the content of the notification in the Content area.**

   Log Service provides default content for notifications. You can customize this content according to your needs.

To call attention of one or more group members by using ats (@), enter `@ cell phone number`.

Figure 8-3: Content of Notification



WebHook

You can set WebHook as the notification method for alarm notifications to be sent to a specified URL.

 Note:
The timeout limit of this type of notification is five seconds. If no response is received within five seconds after a request is sent, Log Service determines that the request as a failed request.

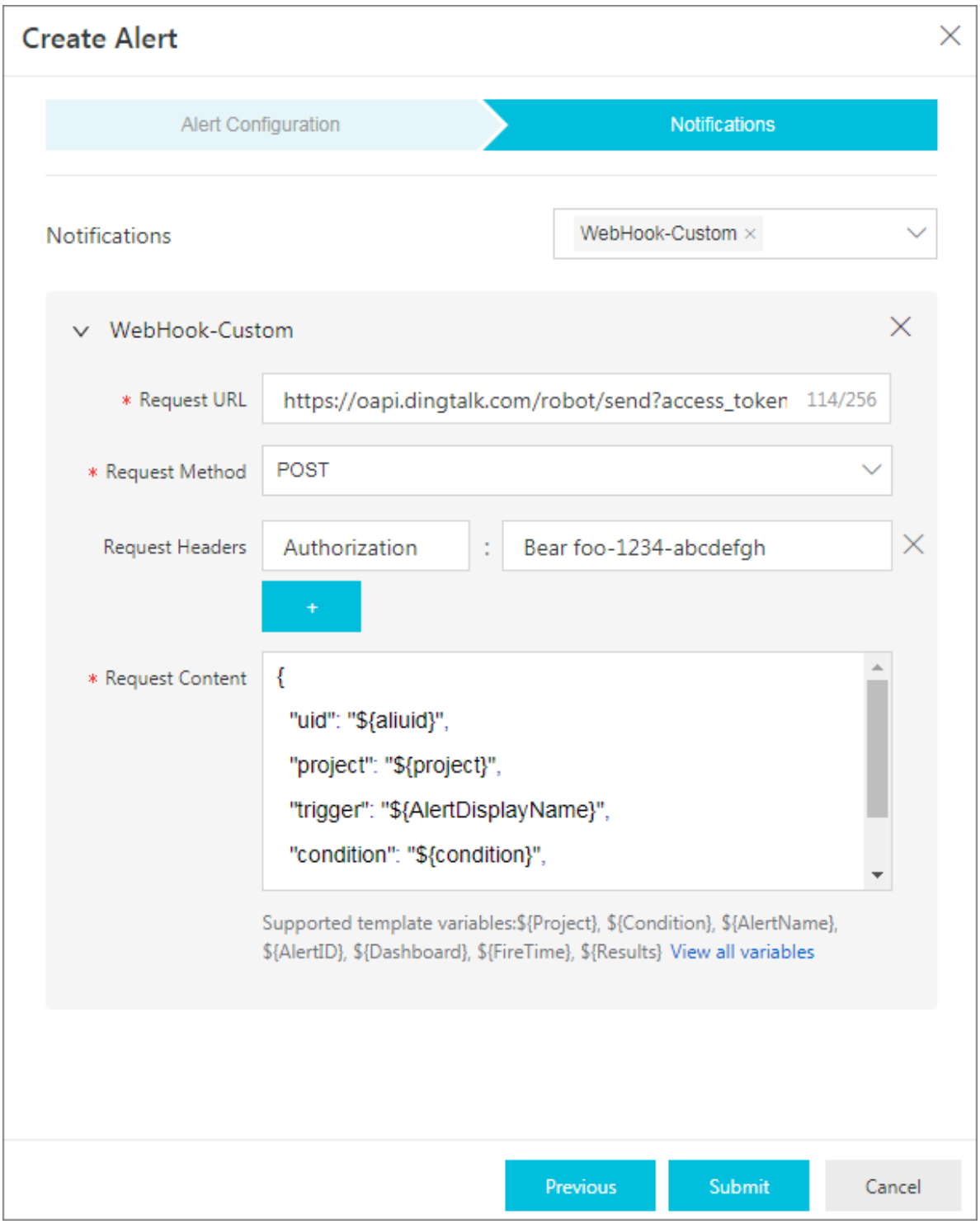
Procedure

1. *Create an alarm* in the Log Service console. Select WebHook-Custom from the notification drop-down list.
2. In the Request URL box, enter a customized WebHook URL. Select a Request Method.
3. (Optional) Click Add Request Headers to add request headers.

   By default, the `Content - Type : applicatio n / json ; charset = utf - 8` request header is included in the notification.

4. Enter Request Content.

Figure 8-4: Notification content



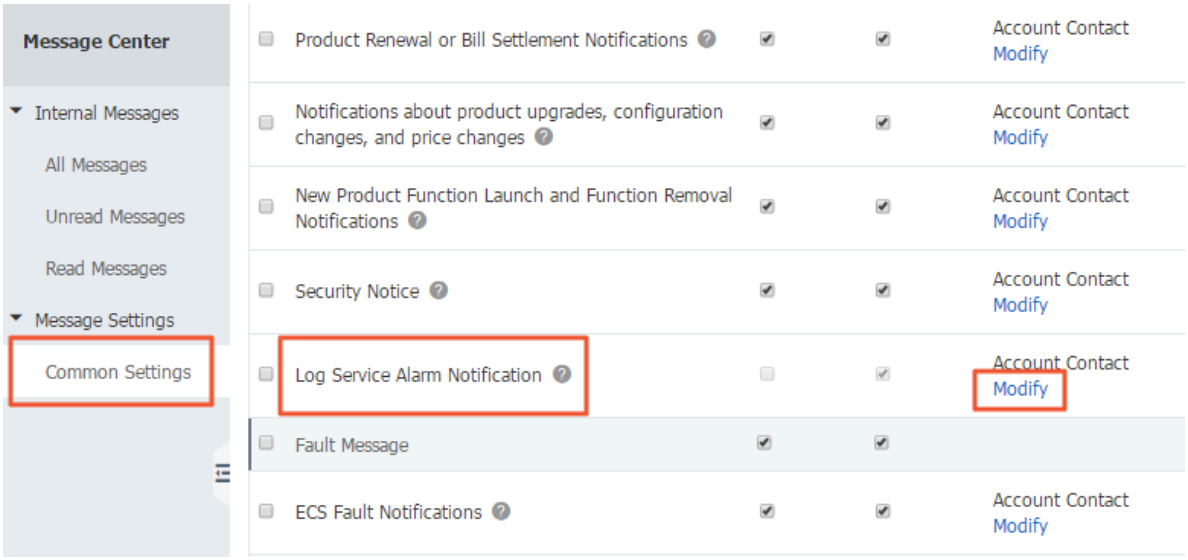
5. Click Submit.

Message Center (recommended)

In the Message Center console, you can set receivers of Log Service alarm notificati ons. Then, when an alarm is triggered, Message Center sends an alarm notification by using the specified method.

Procedure

1. *Configure an alarm*Select Notifications from the notification drop-down list.
2. In the *Message Center* console, choose Message Settings > Common Settings.



3. In the Account Contact column of Notification Type > Log Service Alarm Notification, click Modify.
4. In the Modify Contact dialog box, select the required message receivers.

china site china site

> **Note:**
>
> - china site
> - You must set at least one message receiver.
> - By default, notification messages are sent through email only and no other method can be set.
> - Up to fifty alarm notification messages can be sent to a specified email inbox per day.

Notification content

You must enter notification content for each notification. The template variables used when an alarm is triggered can be referenced in the format of ${fieldName}. Log Service replaces the template variables of the notification content with the real values when sending an alarm. For example, ${Project} will be replaced with the name of the Project to which the alarm rules belong.

 Note:
You must reference valid variables. If a referenced variable does not exist or you reference an invalid variable, the variable will be processed as an empty string. If the value of the variable that you referenced is of the object type, the value is converted to a JSON string for display.

The following table describes all available variables and their corresponding reference methods.

Variable	Description	Example	Reference example
Aliuid	Aliuid to which a Project belongs	1234567890	The alarm rules of user ${Aliuid} is triggered.
Project	Project to which alarm rules belong	my-project	The alarm of Project ${Project} is triggered.
AlertID	Unique ID of the executed alarm.	0fdd88063a611aa11493 8f9371daeeb6-1671a52eb23	The ID of an executed alarm is ${AlertID}.
AlertName	Alert rule name, which must be unique in a Project	alarm-1542111415-153472	Alert rule ${ AlertName} is triggered.
AlertDisplayName	Displayed name of an alarm rule	My alarm rule	Alert ${AlertDisplayName} is triggered.

Variable	Description	Example	Reference example
Condition	Condition expression that triggers an alarm. Each variable in the condition expression is replaced with the value that triggers an alarm, and is enclosed in square brackets.	[5] > 1	An alarm condition expression is in the format of ${Condition}.
RawCondition	Raw condition expression in which variables of the condition are not replaced with other values	count > 1	A raw condition expression is in the format of ${RawCondition}.
Dashboard	Name of the dashboard associated with an alarm	mydashboard	The dashboard associated with an alarm is ${Dashboard}.
DashboardUrl	Address of the dashboard associated with an alarm	https://sls.console.aliyun.com/next/project/myproject/dashboard/mydashboard	The address of the dashboard associated with an alarm is ${DashboardUrl}.
FireTime	Time when an alarm is triggered	2018-01-02 15:04:05	The triggering time of the alarm is ${FireTime}.

Variable	Description	Example	Reference example
FullResultUrl	URL used to query records of triggered alarms	https://sls.console.aliyun.com/next/project/my-project/logsearch/internal-alarm-history?endTime=1544083998&queryString=AlertID%3A9155ea1ec10167985519fccede4d5fc7-1678293caad&queryTimeType=99&startTime=1544083968	Click to view details: ${FullResultUrl}.
Results	Parameters and results of a query, and array type.	[{ " EndTime ": 1542507580 , " FireResult ": { " __time__ ": " 1542453580 ", " count ": " 0 " }, " LogStore ": " test - logstore ", " Query ": "*	SELECT COUNT (*) as count ", " RawResultCount ": 1 , " RawResults ": [{ " __time__ ": " 1542453580 ", " count ": " 0 " }], " StartTime ": 1542453580 }]	The first query starts at ${Results[0].StartTime}, and ends at ${Results[0].EndTime}. The value of the count parameter is ${Results[0].FireResult.count}. Note: In this example, 0 indicates the sequence number of a chart (the visualization form of a search and analysis statement). For more information, see *How to view the sequence number of a chart?*

8.2.3 Grant a RAM user the permissions to configure alarms

This topic describes how to grant a RAM user the permission to use and manage alarms provided by Log Service.

Context

- If you want to grant a RAM user the corresponding permissions to perform all required actions in Log Service, set the `AliyunLogF ullAccess` parameter. For more information, see *Grant a RAM user account the permission to access Log Service*.
- If you only want to grant a RAM user the permission to create and modify an alarm , you need to create a permission policy and grant it to the user.

Procedure

1. Log on to the *RAM console*.
2. In the left-side navigation pane, choose Permissions > Policies.
3. Click Create Policy.
4. Enter a Policy Name and Note.
5. Select the Script radio box.
6. Copy the following, and then paste it to the Policy Document.

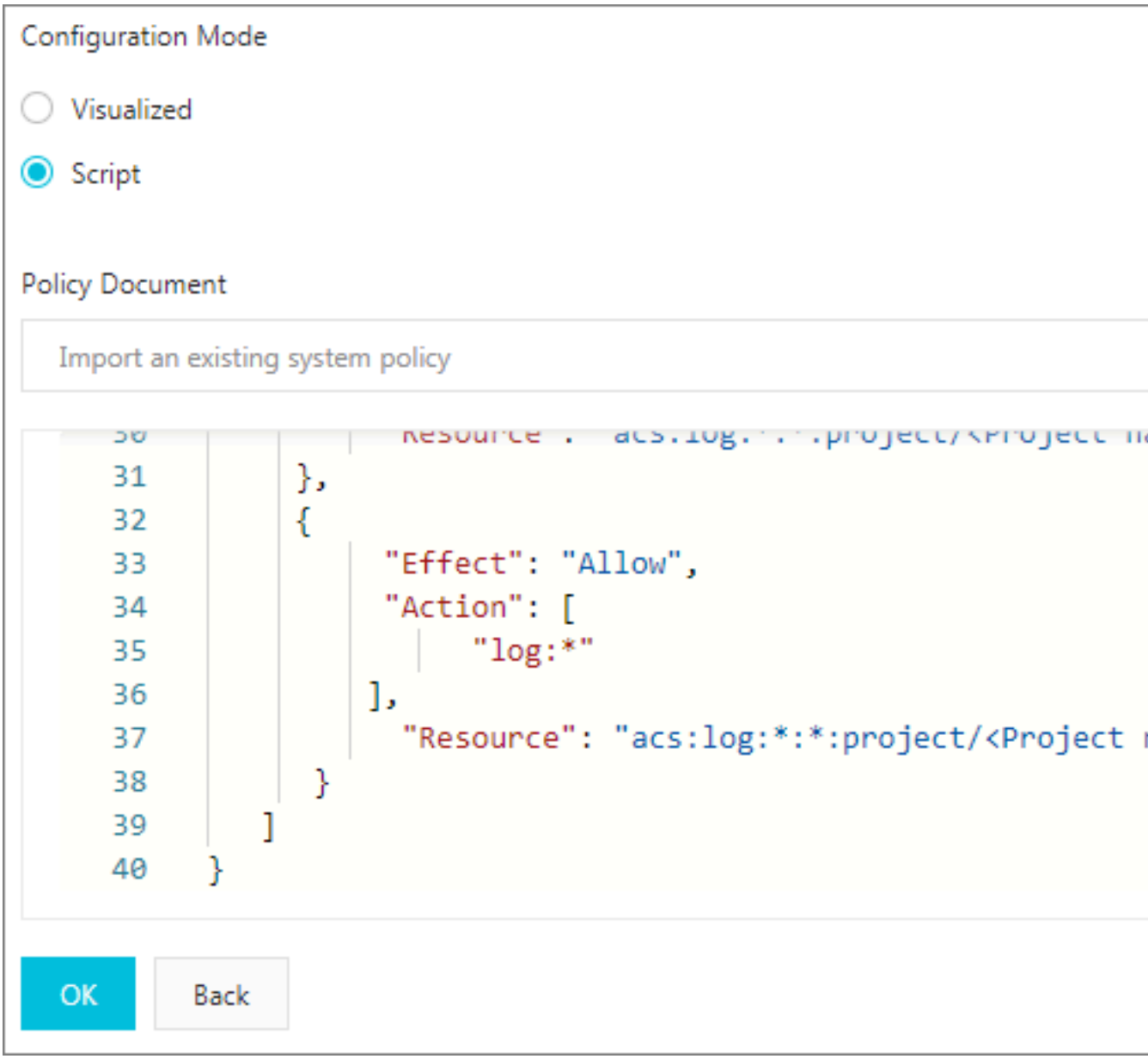


Note:

You need to replace `< Project name >` in the following example with your target project name.

```
{
    " Version ": " 1 ",
    " Statement ": [
     {
         " Effect ": " Allow ",
         " Action ": [
         " log : GetLogStor  e ",
         " log : CreateLogS  tore ",
         " log : GetIndex ",
         " log : CreateInde  x ",
         " log : UpdateInde  x "
       ],
         " Resource ": " acs : log :*:*: project /< Project   name
 >/ logstore / internal - alert - history "
     },
     {
         " Effect ": " Allow ",
         " Action ": [
             " log : CreateProj  ect ",
             " log : GetProject "
        ],
         " Resource ": " acs : log :*:*: project /< Project   name
 >"
     },
     {
          " Effect ": " Allow ",
          " Action ": [
              " log : CreateDash  board ",
```

```
                " log : CreateChar  t ",
                " log : UpdateDash  board "
            ],
             " Resource ": " acs : log :*:*: project /< Project   name
 >/ dashboard /*"
        },
        {
             " Effect ": " Allow ",
             " Action ": [
                  " log :*"
            ],
              " Resource ": " acs : log :*:*: project /< Project   name
 >/ job /*"
        }
    ]
}
```



7. Click OK.
8. In the left-side navigation pane, choose Identities > Users.

9. Click Add Permissions on the right of the target user account.
10. Select the policy created in the preceding step, and then click OK.

8.3 Modify and view an alarm

8.3.1 Modify an alarm configuration

This topic describes how to modify an alarm configuration through using the example of updating the statement associated with the alarm.

Limits

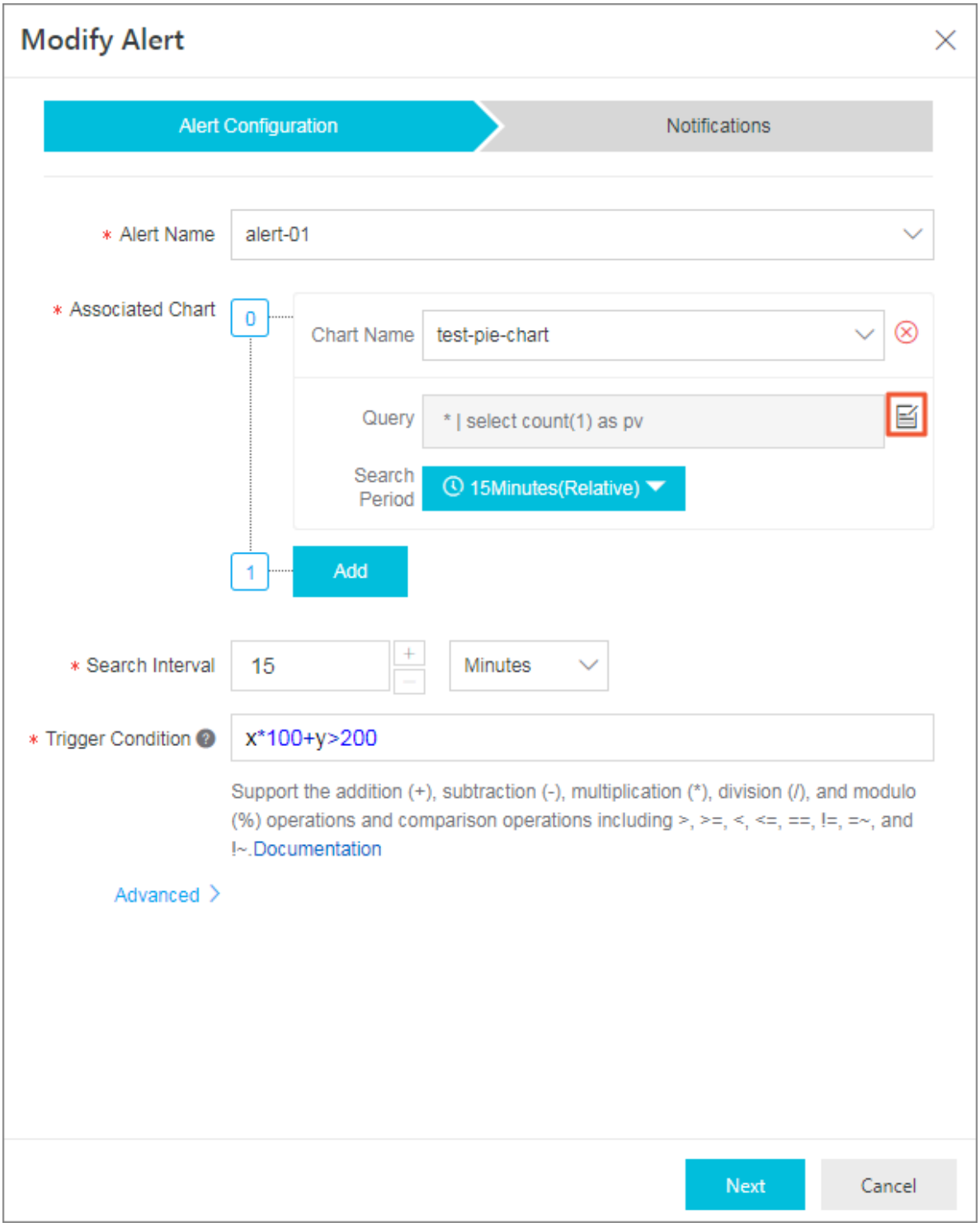
- An alarm can be associated with two types of statements: search statements and search-and-analysis statements. After you associate a statement of either type with an alarm, you can only update the statement as a new version of the same type, rather than one of the other type.

  For example, after you associate the `request_me thod : GET` search statement with an alarm, you can change it to `error` (search statement), but you cannot change it to `error | select count ( 1 ) as c` (search-and-analysis statement).
- For more information about how to modify an alarm configuration of an earlier version, see *Upgrade an alarm configuration to the latest version*.
- If you want to modify an alarm configuration of the latest version, you can click Modify on the Alarm page, or choose Alarm > Modify on the dashboard page where the charts associated with the alarm is added.

Procedure

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane of the Logstores page, click Dashboard.
3. Click the name of the target dashboard.
4. In the progress bar, choose Alert > Modify.

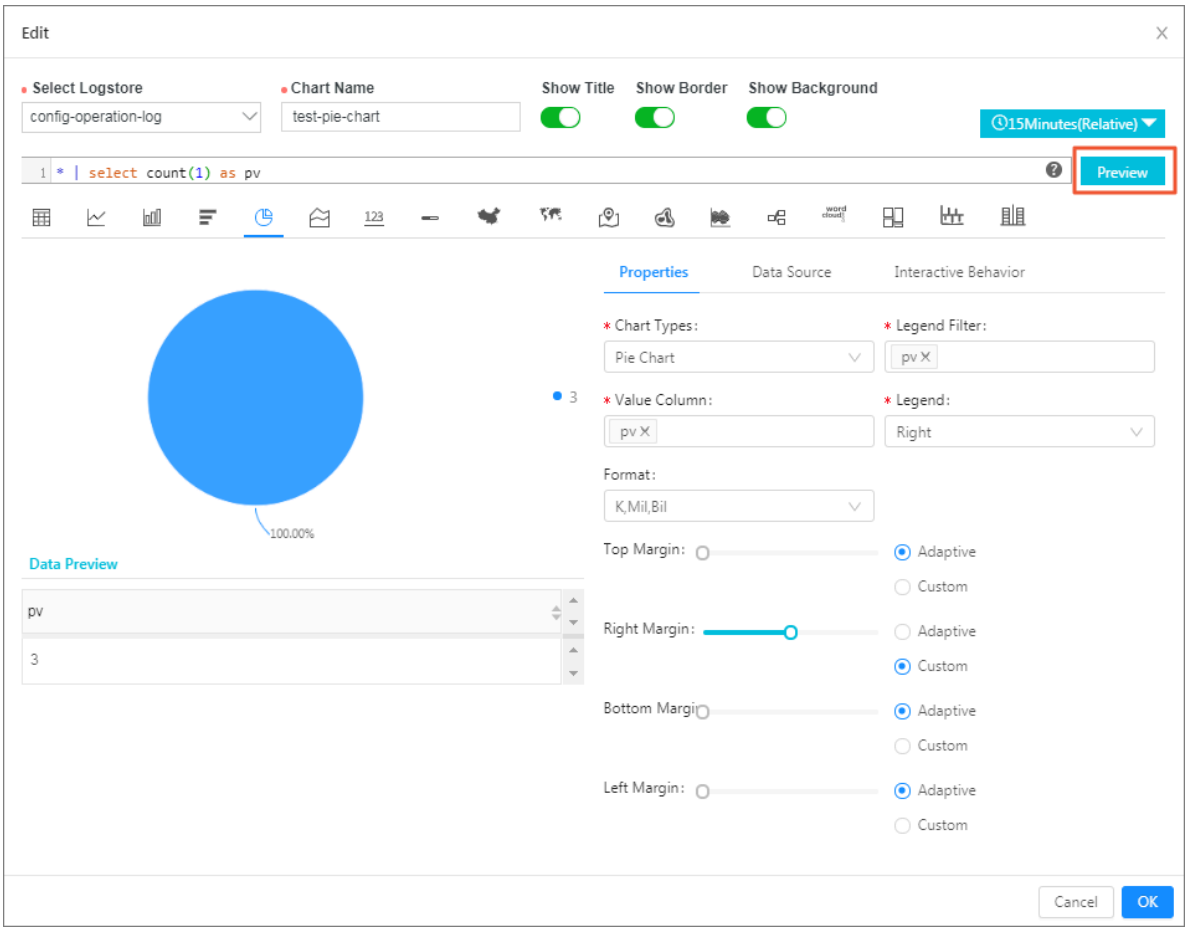
5. On the right of the target statement, click the edit icon .



6. On the displayed page, edit the statement, and then click OK.

Note:

Before clicking OK, you can click Preview to preview the output of the new statement on the page.



7. Modify other parameters for the alarm as needed, such as Search Interval and Trigger Condition, and then click Next. For more information, see *Configure an alarm*
8. Modify notifications for the alarm as needed. For more information, see *Set alarm notifications*.
9. Click Submit.

   The most recent time at which an alarm configuration was updated is shown in the Last Updated At column of the Alarm page.

8.3.2 View and use alarm logs

This topic describes how to view, search, and analyze alarm logs recorded in a Logstore that is created automatically, and also describes how to view the details of operation and notifications of all alarms in a dashboard that is created automatically.

Background

- Alarm logs stored in a Logstore

  When you create an alarm for the first time in a project, Log Service automatically creates a Logstore named internal-alert-history that records the data of all alarms in this project. Each time that an alarm in the project is executed, a log entry is generated to record the event no matter if the alarm is triggered. The log entry is stored in the internal-alert-history Logstore. For more information, see *Alarm log field*.

   Note:
  This Logstore is free of charge. It cannot be deleted or modified. Each log entry is retained in this Logstore for seven days.

- Details of alarm events displayed in a dashboard

  When you create an alarm for the first time in a project, Log Service automatically creates a dashboard named Alert History Statistics in the project to record and display all alarm events. The details of all alarm events in the project include the following information:

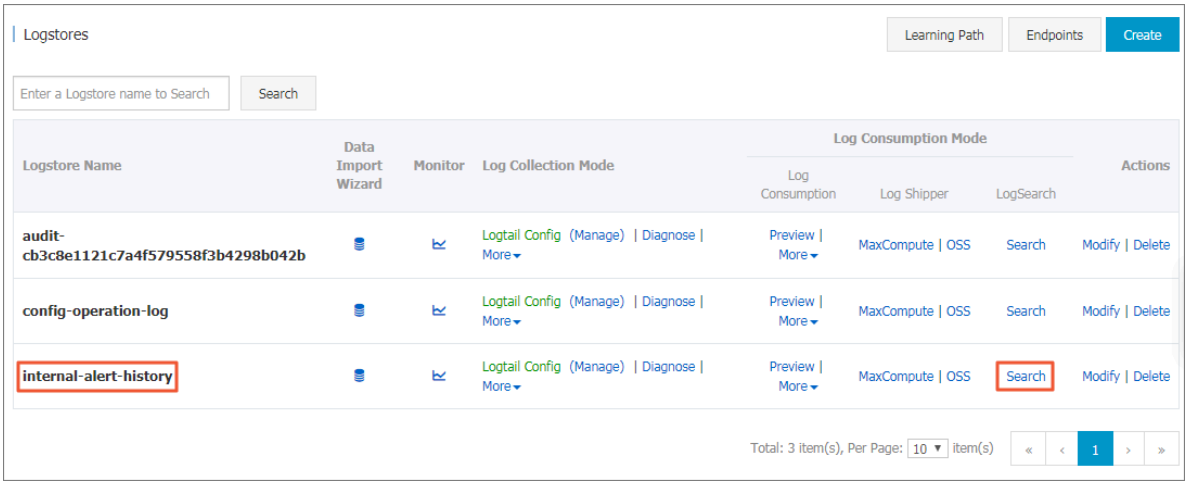
  - The number of times in which alarm notifications are sent.
  - The ratio of successful alarms to the total number of alarms.
  - The ratio of alarms whose notifications are sent to the total number of successful alarms.
  - The 10 alarms that are executed for the greatest number of times.
  - The status of whether an alarm is executed or triggered.
  - The status of whether the notifications of an alarm are sent.
  - The cause for which an alarm failed to be triggered.
  - Each error message and its description and solution.

   Note:
  This dashboard cannot be deleted or modified. You can use it free of charge.

View alarm logs in the Logstore

On the search page of the internal-alert-history Logstore, you can preview, search, and analyze alarm logs recorded in this Logstore. For more information, see *Alarm log field*.

1. Log on to the *Log Service console*, and then click the target project name.
2. Find the Logstore internal-alert-history and click Search in the LogSearch column.

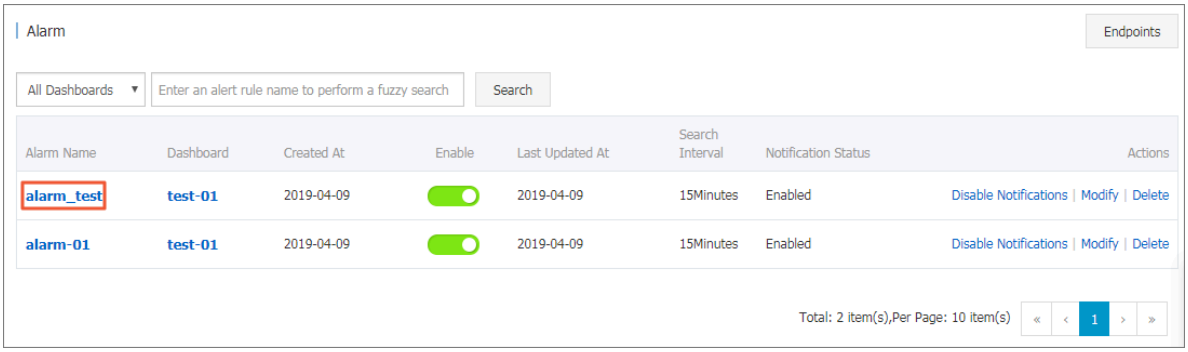


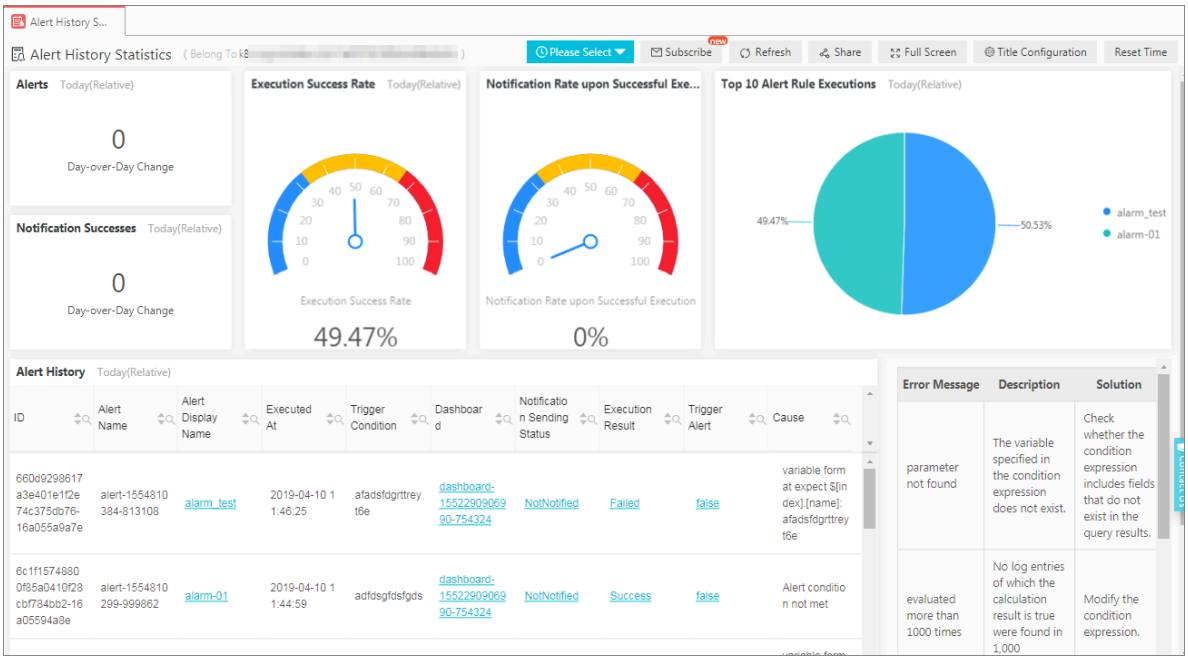
3. Search for and analyze alarm logs as needed.

View alarm records in a dashboard

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane of the Logstores page, click Alarm.

3. Click any alarm name to open the Alert History Statistics dashboard.





8.3.3 Manage an alarm

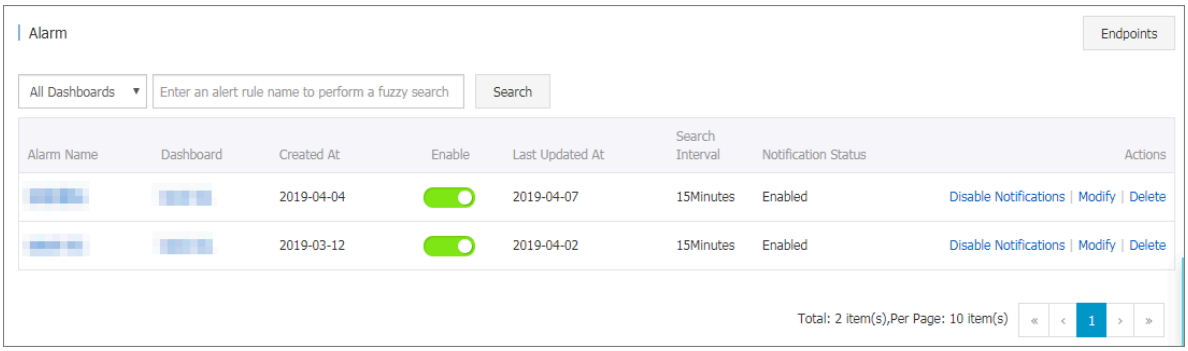
This topic describes how to manage an alarm on the Alarm page. Specifically, this topic describes how to view overview information of an alarm, how to disable or enable an alarm, disable and recover alarm notifications, and how to delete an alarm.

View overview information of an alarm

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane of the Logstores page, click Alarm.

   The Logstores page displays information relating to the alarms you created such as the name of the corresponding dashboard where an alarm is attached, the date at

which each alarm was created and updated, and whether the notification of each alarm is enabled.

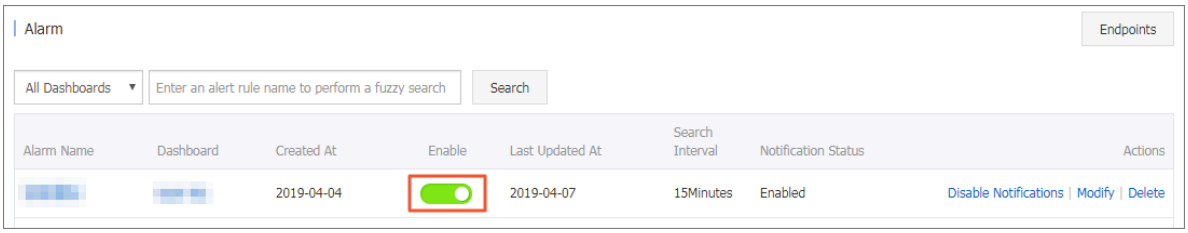


Disable or enable an alarm

After you create an alarm, you can disable or enable the alarm at any time. If you disabled an alarm, then the system does not perform required analyses at specified intervals or send alarm notifications.

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane of the Logstores page, click Alarm.
3. On the right of the target alarm, turn on the Enable switch.

   The switch indicates the status of an alarm.

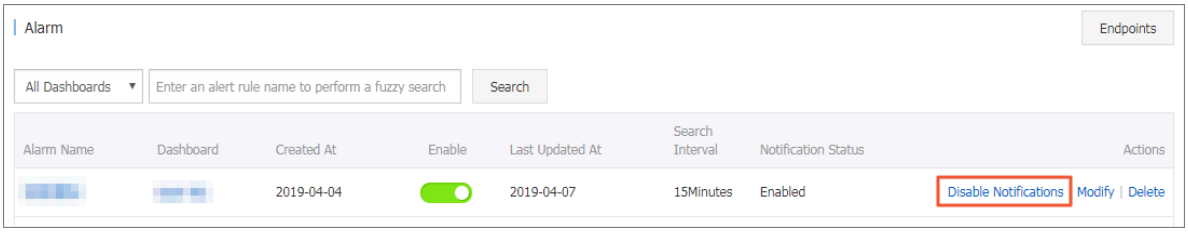


Disable and recover alarm notifications

After you disable alarm notifications for an enabled alarm, the system still performs the required analyses at specified intervals. However, the system does not send any notifications when the condition for triggering an alarm is met during the period in which you have disabled alarm notifications.

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane of the Logstores page, click Alarm.

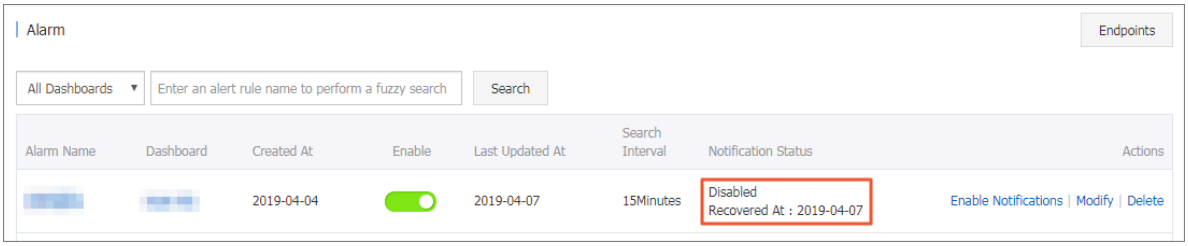
3. On the right of the target alarm, click Disable Notifications.



4. Set the time length in which the alarm notifications remains disabled, and then click Confirm.

   After you disable alarm notifications for an alarm, you can view the data at which alarm notifications will be recovered in the Notification Status column.

> **Note:**
> The disabled state of alarm notifications can last up to 30 days.

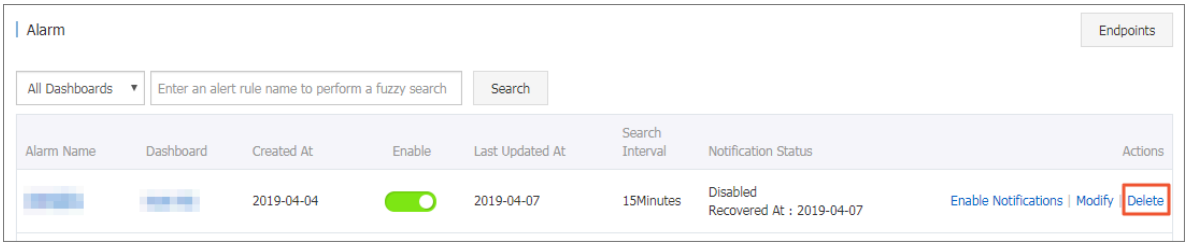


5. Optional. To enable alarm notifications for the alarm before the time that alarm notifications are recovered for the alarm, click Enable Notifications.

Delete an alarm

> **Note:**
> A deleted alarm cannot be recovered.

1. Log on to the *Log Service console*, and then click the target project name.
2. In the left-side navigation pane of the Logstores page, click Alarm.
3. On the right of the target alarm, click Delete.



4. In the displayed dialog box, click Confirm.

What to do next

You can also manage an alarm by viewing its records and modify its configurations. For more information, see *View alarm records* and *Modify an alarm configuration*.

8.3.4 Upgrade an alarm configuration to the latest version

This topic describes how to upgrade an alarm configuration of an earlier version to the latest version.

Context

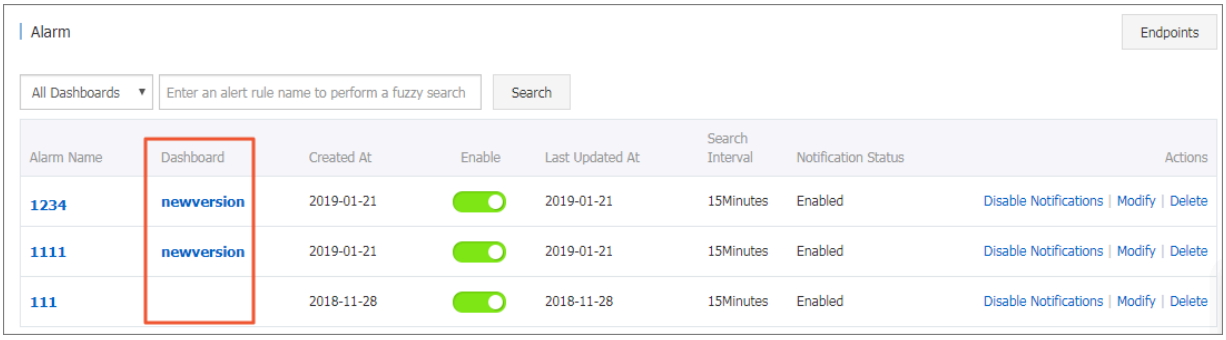
Alarms have been upgraded in the Log Service. Log Service has upgraded the alarm function. If you want to modify an alarm configuration of an earlier version, you need to add modifications to the alarm manually and upgrade it to the latest version. You can retain an alarm of an earlier version because Log Service retains your alarm configuration from before this upgrade. However, we recommend that you upgrade your alarm of an earlier version to the latest version.

The differences between an alarm configuration of an earlier version and the latest version are as follows:

- Alarm configurations of an earlier version

  An alarm configuration created with an earlier version is not attached to any dashboard. That is, on the Alarm page, no information is shown in the Dashboard column of an alarm of an earlier version.
- Alarm configurations of the latest version

  An alarm configuration created with the latest version is attached to a dashboard. That is, on the Alarm page, the Dashboard column of an alarm of the latest version shows the name of the dashboard to which the alarm is attached.

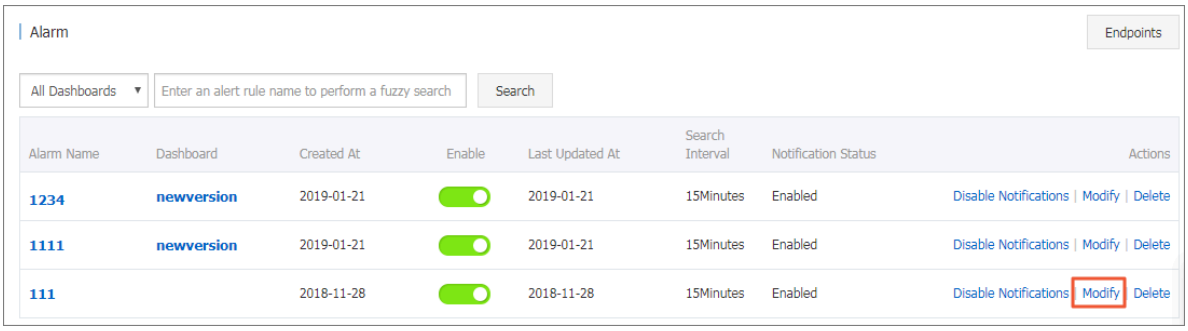


Procedure

1. Log on to the *Log Service console*, and then click the target project name.

2. In the left-side navigation pane, click Alarm.
3. In the Actions column of the alarm of an earlier version, click Modify.

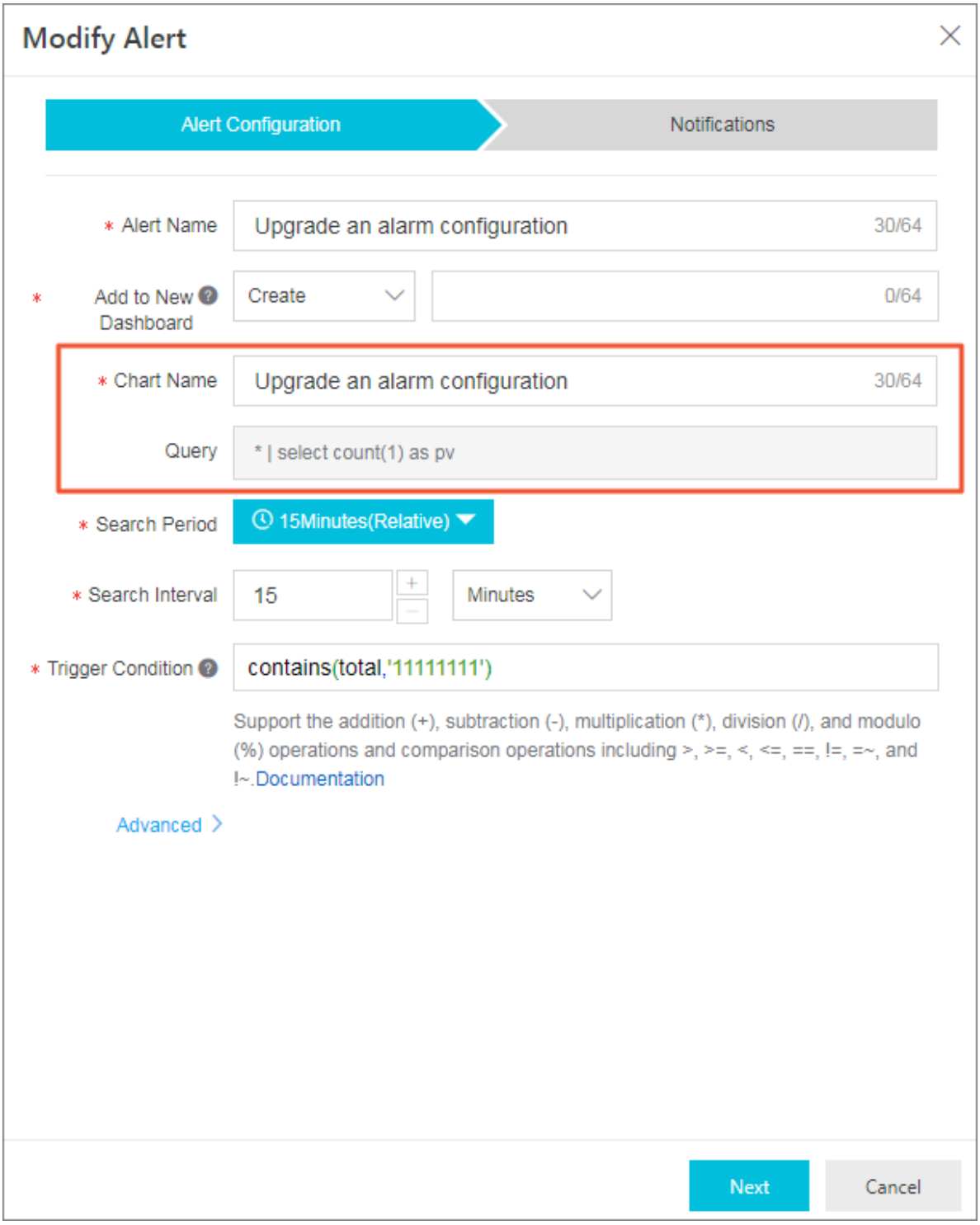
> **Note:**
> The Dashboard column of the alarm of an earlier version shows no information.



4. Set the alarm parameters, and then click Next.

   You only need to set the Chart Name, and the dashboard to which you attach the alarm. Log Service reserves the original configuration for you, such as the original

Alarm Name, Query Statement, and Trigger Condition. For more information, see *Set an alarm*.



5. Set the notification method.

   By default, Log Service retains the notification method and content of the original alarm configuration. You can add one or multiple notification methods.

6. Click Submit.

   After you upgrade an alarm configuration of the earlier version to the latest version, you can view the chart that was automatically created by the system in the dashboard where the alarm is attached. Moreover, you can view the records and configuration of the alarm on the Alert History Statistics page.

8.4 Relevant syntax and fields for reference

8.4.1 Set an alarm condition expression

To use the alarm function, you can set an expression of alarm conditions. Based on the true or false result of the expression, the system determines whether the alarm conditions are met.

When the system determines whether an alarm condition expression is true or false, the results of your configured queries are used as fixed values and log fields are used as variables. If the conditions of your expression are true, an alarm is triggered.

Limits

- You must enclose negative numbers in parentheses (), for example, x+(-100)<100.
- The numeric value type is a 64-bit floating-point number. If you perform comparison operations, errors may occur. For example, using the equal to operator (==) may cause errors.
- A variable can contain only letters and numbers, and must start with a letter.
- An expression can be up to 128 characters in length.
- If you need to combine multiple result sets to evaluate your expression, up to 1000 combinations of result sets can be calculated. If all the combinations of result sets are false, your expression is then considered false.
- Up to three queries can be configured for an alarm.
- An alarm is triggered only when the value of an expression is the Boolean value true. For example, the expression of 100+100 does not trigger an alarm because the expression calculation result of is 200.
- `true` , `false` ,`$`, and `.` are reserved and cannot be used as variables.

Basic syntax

Alarm condition expressions support the following types of syntax.

Syntax type	Description	Example
Basic operators	Available basic operators are: addition (+), subtraction (-), multiplication (*), division (/), and modulus (%).	x*100+y>200 x%10>5
Comparison operators	Available comparison operators are: greater than (>), greater than or equal to (> =), less than (<), less than or equal to (<=), equal to (==), not equal to (! =), regular expression match (= ~), regular expression not-match (! ~) . Note: · Slashes must be escaped. · Regular expressions support syntax that meets the requirements of *RES2 Guide*.	x >= 0 x < 100 x <= 100 x == 100 x == "foo" Regular expression match: x =~ "\\w+"
Logical operators	Available logical operators are: and (&&) and or (||).	x >=0&&y <=100 x > 0 || y > 0
Not operator	Not operator (!).	!(a < 1 && a > 100)
Numeric constants	Numeric constants are handled as 64-bit floating-point numbers.	x > 100
String constants	The form of a string constant is a string enclosed in single quotation marks. For example, 'string '.	foo == 'string'
Boolean constants	Available Boolean constants are true and false.	(x > 100) == true
Parentheses ()	Parentheses () raise calculation precedence.	x*(y+100)>100
Contains function	The contains function is used to determine whether a sub-string is included. For example, if the contains(field, 'xxxx') expression returns the true result, you can determine that the xxxx sub-string is included in the field string.	contains(foo, 'hello ')

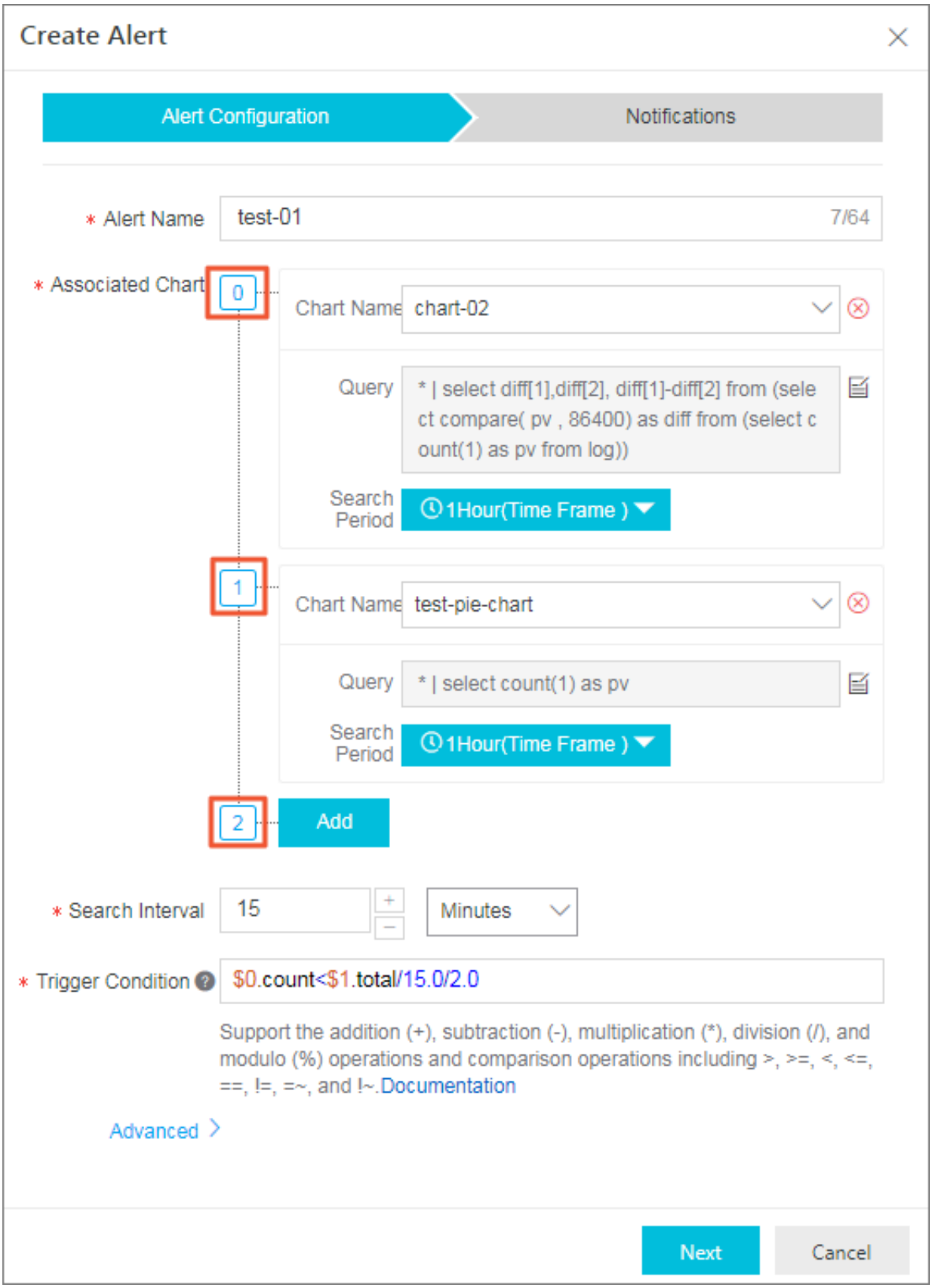
Combine multiple result sets to evaluate an expression

- Syntax

  You can associate multiple charts with an alarm. The system will then obtain multiple query results to evaluate the condition expression that you set. In this case, you must prefix the variables of your condition expression. Then the system can determine from which query result to obtain the corresponding values of the variables when evaluating your expression. The format of the variables is $N.fieldname, where N indicates the order number of a query. You can configure up to three queries. The value range of N is 0 to 2. For example, $0.foo indicates the foo filed of the first query. If you configure only one query, no prefix is required.
- Sequence numbers of charts associated with an alarm

  The Associated Chart area on the Create Alert page provides a sequence number (0, 1, or 2) for each of the charts to be associated with the alarm. These numbers

are based on the order in which these three charts are associated with the alarm chronologically.



- Evaluate an expression

  If multiple query results are returned, the system determines which query result to use to evaluate your expression according to the variables set in your expression. For example, if you configure three queries, the number of results returned by the queries are x, y, and z, and your expression is $0.foo > 100 && $1.bar < 100. In this case, only the first two result sets are needed to evaluate the expression. The system will evaluate your expression for x*y times until the true value is returned, or continue calculating until the limit of calculation attempts is reached and the false value is returned. The maximum limit of calculation attempts is 1000.

Operation methods



Note:

- Numbers used in operations are 64-bit floating-point numbers.
- Each string constant must be enclosed in single quotation marks or double quotation marks, for example, 'string', and "string".
- Boolean values include true and false.

Operator	Operation method		
	Operation on two variables	Operation on a non-string constant and a variable	Operation on a string constant and a variable
Arithmetic operations (+-*/%)	The left and right values are converted to numbers to be used in an operation.		Not supported.

Operator	Operation method		
	Operation on two variables	Operation on a non-string constant and a variable	Operation on a string constant and a variable
Comparison operations: Greater than (>), greater than or equal to (>=), less than (<), less than or equal to (<=), equal to (==), and not equal to (! =)	The order of operation precedence is as follows: 1. The left and right values are converted to numbers and then used in an operation in the numeric value order. If the left and right values fail to be converted, then they are used in an operation of a lower precedence. 2. The left and right values are used as string-type values in an operation of the lexicographic order.	The left and right values are converted to numbers to be used in an operation of the numeric order.	The left and right values are used as string-type values in an operation of the lexicographic order.
Whether a regular expression is matched: regular match (= ~), regular not match (! ~)	The left and right values are used as string-type values in an operation.	Not supported.	The left and right values are used as string-type values in an operation.
Logical operations: and (&&) and or (||)	These two operators cannot be directly used on the query result fields. The left and right values must both be sub-expressions, and the operation results are of both the bool type.		
Not operator (!)	This operator cannot be directly used on the query result fields. The inverted value must be a sub-expression and the operation result is of the bool type.		

Operator	Operation method		
	Operation on two variables	Operation on a non-string constant and a variable	Operation on a string constant and a variable
String lookup function (contains)	The left and right values are converted to the string-type values to participate in an operation.	Not supported.	The left and right values are used as string-type values in an operation.
Parentheses ()	Determine the order of operation precedence.		

8.4.2 Alarm log fields

This topic describes alarm log fields. Log Service automatically creates a Logstore to record the events related to alarms by using the form of logs.

Fields of alarm execution logs

Field name	Description	Example
AlertDisplayName	The customized alarm name that is displayed in the console.	alarm-01
AlertID	The ID of a specific execution of an alarm.	0fdd88063a611aa11493 8f9371daeeb6-1671a52eb23
AlertName	The unique alarm name generated by the system in a project.	alert-1542111415-153472
Condition	The condition for triggering an alarm.	$0.count > 1
Dashboard	The dashboard where an alarm is configured.	my-dashboard
FireCount	The number of times for which an alarm has been triggered after the last time when alarm notifications are sent.	1

Field name	Description	Example
Fired	Indicates whether an alarm is triggered. Valid values: true | false.	true
LastNotifiedAt	The time when the most recent alarm notifications are sent. It is displayed as a Unix timestamp.	1542164541
NotifyStatus	The alarm notification status. · `Success`: indicates that alarm notifications were sent successfully. · `Failed`: indicates that alarm notifications failed to be sent. · `NotNotifie d`: indicates that no notifications were sent. · `PartialSuc cess`: indicates that only part of alarm notifications were sent successfully.	`Success`

Field name	Description	Example
Reason	Causes for which the system failed to send alarm notificati ons or the system did not send alarm notifications as required.	result type is not bool
Results	The alarm log searching result, which includes the parameters used and the array type. For more information, see *Result fields*.	[{ " EndTime ": 1542334900 , " FireResult ": null , " LogStore ": " test - logstore ", " Query ": "* | select count (1) as count ", " RawResultC ount ": 1 , " RawResults ": [{ " __time__ ": " 1542334840 ", " count ": " 0 " }], " StartTime ": 1542334840 }]
Status	The alarm execution result. Valid values: Success | Failed.	Success

Result fields

Field name	Description	Example
Query	The query statement, which can be a search statement or a search-and-analysis statement.	* | select count(1) as count
LogStore	The target Logstore.	my-logstore
StartTime	The starting time of a search.	2019-01-02 15:04:05
StartTimeTs	The starting time of a search (displayed as a Unix timestamp).	1542334840
EndTime	The end time of a search.	2019-01-02 15:19:05
EndTimeTs	The end time of a search (displayed as a Unix timestamp).  Note: The time range of a search is between the StartTime and the EndTime.	1542334900

RawResults	The raw results of a search, including the array type.	[
		{
		" __time__ ": " 1542334840 ",
		" count ": " 0 "
		}
		]

RawResults AsKv	The formatted key-value raw logs that trigger an alarm.  Note: This field can only be used as a template variable, and will not be stored in a Logstore.	[foo:0]

Field name	Description	Example
RawResultCount	The number of the raw results.	1

FireResult	The log of a triggered alarm. If no alarm is triggered, this field displays null.	<pre>{
		" __time__ ": " 1542334840 ",
		" count ": " 0 "
		}</pre>

FireResultAsKv	The formatted key-value raw logs that trigger an alarm.  Note: This field can only be used as a template variable, and will not be stored in a Logstore.	[foo:0]

9 Real-time subscription and consumption

9.1 Overview

Logs collected to the LogHub of the Log Service can be consumed in the following three methods:

Approach	Scenario	Real time	Storage period
Real-time consumption (loghub)	Stream computing and real-time computing	Real-time	Customize
Query and analysis (LogSearch/ Analytics)	Online query and analysis	Real-time (less than one second in 99.99 % cases)	Customize
Shipping and storage (LogShipper)	Full log storage for offline analysis	5–30 minutes	Depends on the storage system

Real-time consumption

Logs are consumed after being written. Both log consumption and log query require the capability of reading logs. Logs in a shard are consumed as follows.

1. Obtain a cursor based on a set of criteria such as time, Begin, and End.
2. The system reads logs based on the cursor and step and returns the next cursor.
3. Moves the cursor continuously to consume logs.

Besides the basic APIs, Log Service provides many methods to consume logs, such as SDKs, Storm spout, Spark Streaming client, Flink connector, consumer library, and Web console.

- Use *Spark Streaming Client* to consume logs.
- Use *Storm Spout* to consume logs.
- Use *Flink Connector*, including Flink consumer and Flink producer to consume logs.
- Use *LogHub Consumer Library* to consume logs. The consumer library is an advanced mode for LogHub consumers, which provides a lightweight computing framework and solves the issue of automatic shard allocation and order preservation when multiple consumers consume a Logstore at the same time.

- Use *SDKs* to consume logs. Log Service provides SDKs in multiple languages (Java and Python) that support the log consumption APIs. For more information about SDKs, see *Log Service SDK*.
- Use cloud products to consume logs:
  - *Use CloudMonitor to consume logs* : Monitoring scenario.
  - Use E-MapReduce to consume logs: See *Storm*, *Spark Streaming*.
  - Use *Command-line tools CLI* to consume logs.

Query and analysis

Overview of real-time query and analysis:

- Query logs in the Log Service console: See *Query logs*.
- Query logs by using Log Service SDKs/APIs: Log Service provides RESTful APIs that are implemented based on HTTP protocol. The Log Service APIs also provide a full-featured log query API. For more information, see *Log Service APIs*.

Shipping and storage

- *Ship logs to OSS*: Store logs for a long term or use E-MapReduce to analyze logs.Analysis log.
- *Use function calculations for custom delivery*.

Others

Secure Log Service: Log Service interconnects with cloud security products and uses ISV to consume logs of cloud products.

9.2 Preview log data

Log preview is a common form of log consumption. The Log Service console provides a preview page to directly preview some logs in the Logstore in the console.

Procedure

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. On the Logstore List page, click Preview at the right of the Logstore.

4. 4.On the log query page, select the shard of the Logstore and the log time range. Then, click Preview.

   Data of the first 10 data packets in the specified time range is displayed.

   Figure 9-1: Preview log data



9.3 Consumption by consumer groups

9.3.1 Use a consumer group to consume logs

The consumer library is an advanced mode of log consumption in Log Service, and provides the consumer group concept to abstract and manage the consumption end. Compared with using SDKs directly to read data, you can only focus on the business logic by using the consumer library, without caring about the implementation details of Log Service, or the load balancing or failover between consumers.

Spark Streaming, *Storm*, and Flink connector use consumer library as the base implementation.

Basic concepts

You must understand two concepts before using the consumer library: consumer group and consumer.

- Consumer group

  A consumer group is composed of multiple consumers. Consumers in the same consumer group consume the data in the same Logstore and the data consumed by each consumer is different.

- Consumer

  Consumers, as a unit that composes the consumer group, must consume data. The names of consumers in the same consumer group must be unique.

In Log Service, a Logstore can have multiple shards. The consumer library is used to allocate a shard to the consumers in a consumer group. The allocation rules are as follows:

- Each shard can only be allocated to one consumer.
- One consumer can have multiple shards at the same time.

After a new consumer is added to a consumer group, the affiliations of the shards for this consumer group is adjusted to achieve the load balancing of consumption. However, the preceding allocation rules are not changed. The allocation process is transparent to users.

The consumer library can also save the checkpoint, which allows consumers to consume data starting from the breakpoint after the program fault is resolved and makes sure that the data is consumed only once.

Usage

Add maven dependency

```
< dependency >
  < groupId > com . google . protobuf </ groupId >
  < artifactId > protobuf - java </ artifactId >
  < version > 2 . 5 . 0 </ version >
</ dependency >
< dependency >
  < groupId > com . aliyun . openservic  es </ groupId >
  < artifactId > loghub - client - lib </ artifactId >
  < version > 0 . 6 . 16 </ version >
</ dependency >
```

main .java file

```
public   class   Main  {
   //  Enter   the   domain   name   of   Log   Service   according
to   your   actual   situation .
  private   static   String   sEndpoint  = " cn - hangzhou . log .
aliyuncs . com ";
   //  Enter   the   project   name   of   Log   Service   according
  to   your   actual   situation .
  private   static   String   sProject  = " ali - cn - hangzhou - sls
- admin ";
   //  Enter   the   Logstore   name   of   Log   Service   according
  to   your   actual   situation .
  private   static   String   sLogstore  = " sls_operat  ion_log ";
   //  Enter   the   consumer   group   name   according   to   your
  actual   situation .
  private   static   String   sConsumerG  roup  = " consumerGr  oupX
";
   //  Enter   the   AccessKey   of   data   consumptio  n
according   to   your   actual   situation .
  private   static   String   sAccessKey  Id  = "";
  private   static   String   sAccessKey  = "";
  public   static   void   main ( String  [] args )  throws
LogHubClie  ntWorkerEx  ception ,  Interrupte  dException
 {
            //  The   second   parameter   is   the   consumer
name .  The   consumer   names   in   the   same   consumer   group
must   be   unique .  However ,  the   consumer   group   names   can
  be   duplicate .  Different   consumer   names   start   multiple
```

```
 processes   on   multiple   machines   to   consume   a   Logstore
 in   a   load   balancing   way .  In   this   case ,  the   consumer
   names   can   be   classified   by   machine   IP   address .   The
   ninth   parameter   maxFetchLo  gGroupSize   is   the   number   of
   Logstores   each   time   obtained   from   Log   Service .  Use
   the   default   value .  If   you   must   adjust   the   value ,
 make   sure   the   value   range   is  ( 0 , 1000 ].
       LogHubConf  ig   config  =  new   LogHubConf  ig ( sConsumerG
 roup , " consumer_1 ",  sEndpoint ,  sProject ,  sLogstore ,
 sAccessKey  Id ,  sAccessKey ,  LogHubConf  ig . ConsumePos  ition .
 BEGIN_CURS  OR );
       ClientWork  er   worker  =  new   ClientWork  er ( new
 SampleLogH  ubProcesso  rFactory (),  config );
         Thread   thread  =  new   Thread ( worker );
        // The   ClientWork  er   automatica  lly   runs   after   the
   thread   is   running   and   extends   the   Runnable   API .
        thread . start ();
        Thread . sleep ( 60  *  60  *  1000 );
             // Call   the   Shutdown   function   of   worker   to
 exit   the   consumptio  n   instance .  The   associated   thread
 is   automatica  lly   stopped .
        worker . shutdown ();
       // Multiple   asynchrono  us   tasks   are   generated   when
 the   ClientWork  er   is   running .  We   recommend   that   you
 wait   30   seconds   until   the   running   tasks   exit   after
 the   shutdown .
        Thread . sleep ( 30  *  1000 );
  }
}
```

SampleLogHubProcessor.java files

```
 public   class   SampleLogH  ubProcesso  r   implements   ILogHubPro
 cessor
{
   private   int   mShardId ;
  //  Records   the   last   persistent   checkpoint   time .
   private   long   mLastCheck  Time  =  0 ;
   public   void   initialize ( int   shardId )
  {
       mShardId  =  shardId ;
  }
  //  The   main   logic   of   data   consumptio  n .  Catch   all
   the   exceptions   but   the   caught   exceptions   cannot   be
 thrown .
   public   String   process ( List < LogGroupDa  ta >  logGroups ,
          ILogHubChe  ckPointTra  cker   checkPoint  Tracker )
  {
         //  Write   checkpoint   to   Log   Service   every   30
   seconds .  If   worker   crashes   within   30   seconds ,  the
 newly   started   worker   consumes   data   starting   from   the
 last   checkpoint .  Slight   duplicate   data   may   exist .
       for ( LogGroupDa  ta   logGroup :  logGroups ){
          FastLogGro  up   flg  =  logGroup . GetFastLog  Group ();
          System . out . println ( String . format ("\ tcategory \ t
 :\ t % s \ n \ tsource \ t :\ t % s \ n \ ttopic \ t :\ t % s \ n \
 tmachineUU  ID \ t :\ t % s ",
                  flg . getCategor  y (),  flg . getSource (),  flg .
 getTopic (),  flg . getMachine  UUID ()));
          System . out . println (" Tags ");
          for  ( int   tagIdx  =  0 ;  tagIdx  <  flg . getLogTags
 Count (); ++ tagIdx ) {
              FastLogTag   logtag  =  flg . getLogTags ( tagIdx );
```

```
                System . out . println ( String . format ("\ t % s \ t
 :\ t % s ",  logtag . getKey (),  logtag . getValue ()));
            }
             for  ( int   lIdx  =  0 ;  lIdx  <  flg . getLogsCou  nt
 (); ++ lIdx ) {
                FastLog   log  =  flg . getLogs ( lIdx );
                System . out . println ("--------\ nLog : " +  lIdx  +
 ",  time : " +  log . getTime () + ",  GetContent  Count : " +  log .
 getContent  sCount ());
                for  ( int   cIdx  =  0 ;  cIdx  <  log . getContent
 sCount (); ++ cIdx ) {
                    FastLogCon  tent   content  =  log . getContent  s
 ( cIdx );
                    System . out . println ( content . getKey () + "\ t
 :\ t " +  content . getValue ());
                }
            }
        }
         long   curTime  =  System . currentTim  eMillis ();
        //  Write   checkpoint   to   Log   Service   every   30
 seconds .  If   worker   crashes   within   30   seconds ,
        //  the   newly   started   worker   consumes   data   starting
   from   the   last   checkpoint .  Slight   duplicate   data   may
 exist .
         if  ( curTime  -  mLastCheck  Time  >  30  *  1000 )
        {
             try
            {
                            // The   parameter   true   indicates   to
 update   the   checkpoint   to   Log   Service   immediatel  y .  The
   parameter   false   indicates   to   cache   the   checkpoint   to
   your   local   machine   and   refresh   the   cached   checkpoint
   to   Log   Service   every   60   seconds   by   default .
                checkPoint  Tracker . saveCheckP  oint ( true );
            }
             catch  ( LogHubChec  kPointExce  ption   e )
            {
                e . printStack  Trace ();
            }
             mLastCheck  Time  =  curTime ;
        }
         return   null ;
    }
    //  The   worker   calls   this   function   upon   exit .  You
 can   perform   cleanup   here .
     public   void   shutdown ( ILogHubChe  ckPointTra  cker
 checkPoint  Tracker )
    {
        // Saves   the   consumptio  n   breakpoint   to   the   Log
 Service .
         try  {
             checkPoint  Tracker . saveCheckP  oint ( true );
        }  catch  ( LogHubChec  kPointExce  ption   e ) {
             e . printStack  Trace ();
        }
    }
}
 class   SampleLogH  ubProcesso  rFactory   implements   ILogHubPro
 cessorFact  ory
{
    public   ILogHubPro  cessor   generatorP  rocessor ()
    {
        //  Generates   a   consumptio  n   instance .
         return   new   SampleLogH  ubProcesso  r ();
```

```
    }
}
```

Run the preceding codes to print all the data in a Logstore. To allow multiple consumers to consume one Logstore, follow the program annotations to modify the program, use the same consumer group name and different consumer names, and start other consumption processes.

Limits and exception diagnosis

Each Logstore can create at most 10 consumer groups. The error `ConsumerGr oupQuotaEx ceed` is reported when the number exceeds the limit.

We recommend that you configure Log4j for the consumer program, which is used to throw the errors occurred in the consumer group and locate the exceptions. Put the log4j.properties file to the resources directory and run the program, the following exception occurs:

```
[ WARN ] 2018 - 03 - 14 12 : 01 : 52 , 747 method : com . aliyun
. openservic es . loghub . client . LogHubCons umer . sampleLogE
rror ( LogHubCons umer . java : 159 )
com . aliyun . openservic es . log . exception . LogExcepti on :
Invalid loggroup count , ( 0 , 1000 ]
```

See the following log4j.properties configuration for reference:

```
log4j . rootLogger = info , stdout
log4j . appender . stdout = org . apache . log4j . ConsoleApp
ender
log4j . appender . stdout . Target = System . out
log4j . appender . stdout . layout = org . apache . log4j .
PatternLay out
log4j . appender . stdout . layout . Conversion Pattern = [%- 5p ]
% d { yyyy - MM - dd HH : mm : ss , SSS } method :% l % n % m % n
```

Status and alarm

1. *View the consumer group status on the console*
2. *View the consumer group delay with CloudMonitor and configure the alarm*

Advanced Configuration

For ordinary users, the data can be consumed using the program above, advanced configurations will be discussed in the following.

- Want to consume data that starts at a certain time

  The loghubconfig in the code above has two constructors:

```
//  The   consumerst  arttimeins  econds   parameter   represents
 the   number   of   seconds   after   1970 ,  meaning   that    the
   data   after   this   is   consumed .
 public   LogHubConf  ig ( String   consumerGr  oupName ,
                       String   consumerNa  me ,
                       String   loghubEndP  oint ,
                       String   project ,  String   logStore ,
                       String   accessId ,  String   accessKey ,
                       int   consumerSt  artTimeInS  econds );
//  Position   is   an   enumeratio  n   variable ,  loghubconf
 ig .  glaseposit  ion .  begin_curs  or   indicates   that
 consumptio  n   starts   with   the   oldest   data ,  loghubconf
 ig .  glaseposit  ion .  end_cursor   indicates   that
 consumptio  n   starts   with   the   latest   data .
 public   LogHubConf  ig ( String   consumerGr  oupName ,
                       String   consumerNa  me ,
                       String   loghubEndP  oint ,
                       String   project ,  String   logStore ,
                       String   accessId ,  String   accessKey ,
                       ConsumePos  ition   position );
```

  You can use different construction methods according to consumer needs, but note that if the server is saved with checkpoint, then the starting consumption position is based on the checkpoint saved by the server.

- Use RAM user to access Log Service

  You need to set the ram permissions associated with the consumer group, and set the method to reference the documentation of the ram, the permissions that need to be set are as follows:

Action	Resource
log:GetCursorOrData	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}
log:CreateConsumerGroup	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/*
log:ListConsumerGroup	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/*

Action	Resource
log:ConsumerGroupUpdateCheckPoint	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}
log:ConsumerGroupHeartBeat	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}
log:UpdateConsumerGroup	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}
log:GetConsumerGroupCheckPoint	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}

- Reset the consumption point

  In some scenarios (fill data, repeat the calculation), we need to set a ConsumerGroup point to a certain point in time, so that the current consumer groups can start to consume from the new point. There are two ways:

  1. Delete consumer group
     - Delete consumer group on the console, and restart consumer group program.
     - consumer group program start to consume from default starting point (configured by program)
  2. Reset the current consumer group to a certain point-in-time using SDK
     - The program and Java code example are as follows
     - Restart the consumer program by using the SDK to modify the site.

```
Client   client  =  new   Client ( host ,  accessId ,  accessKey );
long   time_stamp  =  Timestamp . valueOf (" 2017 - 11 - 15   00 :
00 : 00 "). getTime () /  1000 ;
ListShardR  esponse   shard_res  =  client . ListShard ( new
ListShardR  equest ( project ,  logStore ));
ArrayList < Shard >  all_shards  =  shard_res . GetShards ();
```

```
 for  ( Shard   shard :  all_shards )
{
   shardId  =  shard . GetShardId ();
   long   cursor_tim  e  =  time_stamp ;
   String   cursor  =  client . GetCursor ( project ,  logStore ,
 shardId ,  cursor_tim  e ).  GetCursor ();
   client . UpdateChec  kPoint ( project ,  logStore ,  consumerGr
 oup ,  shardId ,  cursor );
}
```

9.3.2 View consumer group status

The consumer group is an advanced mode of real-time data consumption, which provides multiple consumption instances for the automatic load balancing of Logstore consumption. Both Spark Streaming and Storm use consumer group as the basic mode.

View consumption progress in the console

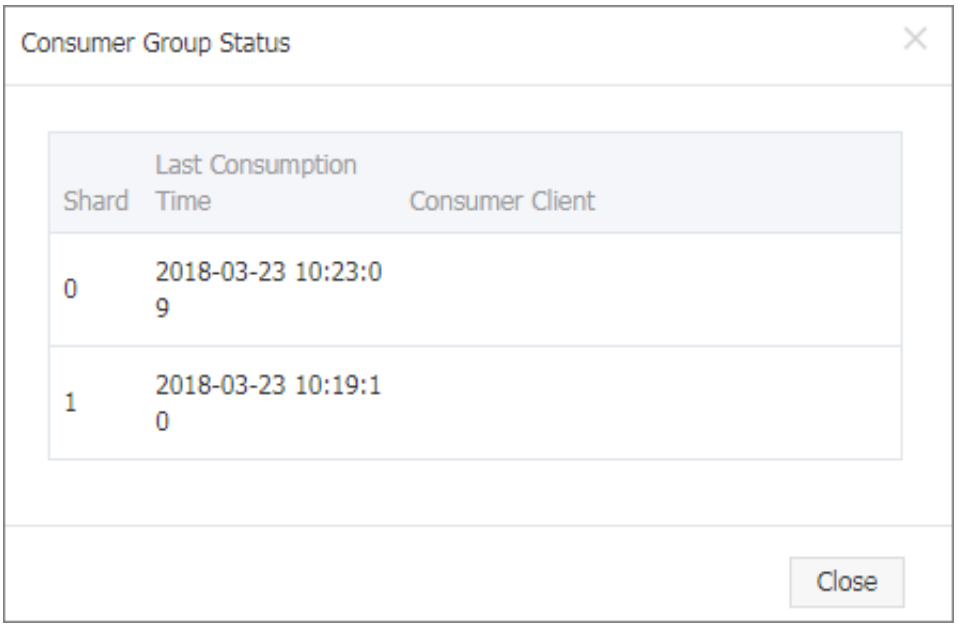
1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. Click LogHub - Consume > Consumerin the left-side navigation pane.
4. On the Consumer Groups page, select a Logstore to view whether or not the consumer group function is enabled or not.

Figure 9-2: Consumer



5. Click Status at the right of the consumer group to view the data consumption progress for each shard.

Figure 9-3: Consumption status



As shown in the preceding figure, the Logstore has six shards and corresponds to three consumers. The latest data consumption time for each consumer is shown under the second column. You can use the data consumption time to determine if the current data processing can keep up with data generation. If data processing severely lags behind (that is, data consumption is slower than data generation), we recommend that you increase the number of consumers.

Use APIs/SDKs to view consumption progress

The following commands use Java SDK as an example, which shows how to use APIs to obtain the consumption status:

```
package   test ;
import   java . util . ArrayList ;
import   com . aliyun . openservic  es . log . Client ;
import   com . aliyun . openservic  es . log . common . Consts .
CursorMode ;
import   com . aliyun . openservic  es . log . common . ConsumerGr
oup ;
import   com . aliyun . openservic  es . log . common . ConsumerGr
oupShardCh  eckPoint ;
import   com . aliyun . openservic  es . log . exception .
LogExcepti  on ;
public   class   ConsumerGr  oupTest  {
    static   String   endpoint  = "";
    static   String   project  = "";
    static   String   logstore  = "";
    static   String   accesskeyI  d  = "";
```

```
    static   String   accesskey  = "";
    public   static   void   main ( String []  args )  throws
LogExcepti  on  {
        Client   client  =  new   Client ( endpoint ,  accesskeyI  d
,  accesskey );
              // Retrieve   all   consumer   groups   in   this
Logstore .  If   no   consumer   group   exists ,  the   consumerGr
oups   length   is   0 .
        ArrayList < ConsumerGr  oup >  consumerGr  oups ;
        try {
            consumerGr  oups  =  client . ListConsum  erGroup (
project ,  logstore ).  GetConsume  rGroups ();

        catch ( LogExcepti  on   e ){
            if ( e . GetErrorCo  de () == " LogStoreNo  tExist ")
                System . out . println (" this   logstore   does
not   have   any   consumer   group ");
            else {
               // internal   server   error   branch

            return ;

        for ( ConsumerGr  oup   c :  consumerGr  oups ){
                        // Print   consumer   group   properties ,
including   names ,  heartbeat   timeout ,  and   whether   or   not
  the   consumptio  n   is   in   order .
                       System . out . println (" Name :" +  c .
getConsume  rGroupName ());
                        System . out . println (" Heartbeat   timeout
:" +  c . getTimeout ());
                        System . out . println (" Consumptio  n   in
  order " +  c . isInOrder ());
            for ( ConsumerGr  oupShardCh  eckPoint   cp :  client .
GetCheckPo  int ( project ,  logstore ,  c . getConsume  rGroupName
()).  GetCheckPo  ints ()){
                System . out . println (" shard : " +  cp . getShard
());
              //  Please   format ,  this   time   returns   the
  exact   time   to   millisecon  ds ,  the   length   of   the
integer
                                // Format   the   returned   time   to
  be   precise   to   millisecon  ds   in   the   long   integer .
                                System . out . println (" Last   data
  consumptio  n   time :" +  cp . getUpdateT  ime ());
                String   consumerPr  g  = "";
                if ( cp . getCheckPo  int (). isEmpty ())
                                      consumerPr  g  = "
Consumptio  n   not   started ";
                else {
                                     // UNIX   timestamp .
Measured   in   seconds .  Format   the   value   upon   output .
                    try {
                        int   prg  =  client . GetPrevCur  sorTime (
project ,  logstore ,  cp . getShard (),  cp . getCheckPo  int ()).
GetCursorT  ime ();
                        consumerPr  g  = "" +  prg ;

                    catch ( LogExcepti  on   e ){
                        if ( e . GetErrorCo  de () == " InvalidCur
sor ")
                                                        consumerPr
g  = " Invalid .  The   previous   consumptio  n   time   has
exceeded   the   data   lifecycle   in   the   Logstore .";
                        else {
```

```
                            // internal   server   error
                             throw   e ;



                                System . out . println (" Consumptio
n   progress :" +  consumerPr  g );
                 String   endCursor  =  client . GetCursor ( project
,  logstore ,  cp . getShard (),  CursorMode . END ).  GetCursor ();
                 int   endPrg  =  0 ;
                 try {
                     endPrg  =  client . GetPrevCur  sorTime ( project
,  logstore ,  cp . getShard (),  endCursor ).  GetCursorT  ime ();

                 catch ( LogExcepti  on   e ){
                    // do   nothing

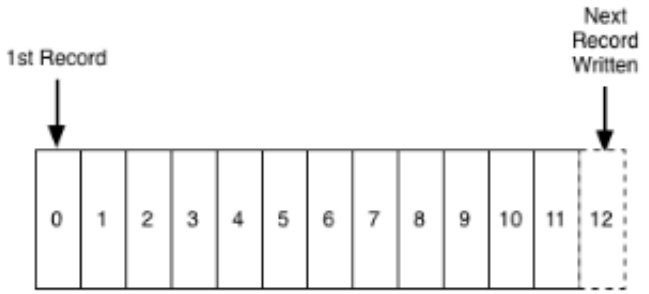
                                // UNIX   timestamp .  Measured   in
seconds .  Format   the   value   upon   output .
                                System . out . println (" The
arrival   time   of   the   last   piece   of   data :" +  endPrg );
```

9.3.3 Consumer group - Monitoring alarm

A consumer group is a group of consumers. Each consumer consumes some of the shards in a Logstore.

The data model of shards can be understood as a queue. The newly written data is added to the tail of the queue and each piece of data in the queue corresponds to a write time. The following shows the data model of shards.

Figure 9-4: Shard Data Model



Basic concepts in collaborative consumption latency alarm:

- Consumption process: The process that a consumer reads data from the head of the queue in sequence.
- Consumption progress: The corresponding write time of the data read by a consumer currently.

- Consumption lagging duration: The difference between the current consumption progress and the latest data write time in the queue, which is measured in seconds.

The consumption lagging duration of a ConsumerGroup takes the maximum value among the consumption lagging durations of all contained shards. When it exceeds the preset threshold (that is, data consumption lags far behind data production), an alarm is triggered.

Procedure

Procedure

1. Log on to the Log Service console. On the Project List page, click the project name.
2. On the Logstore List page, click the Monitor icon at the right of the Logstore.

Figure 9-5: Click the Delay Time of Consumption chart name.



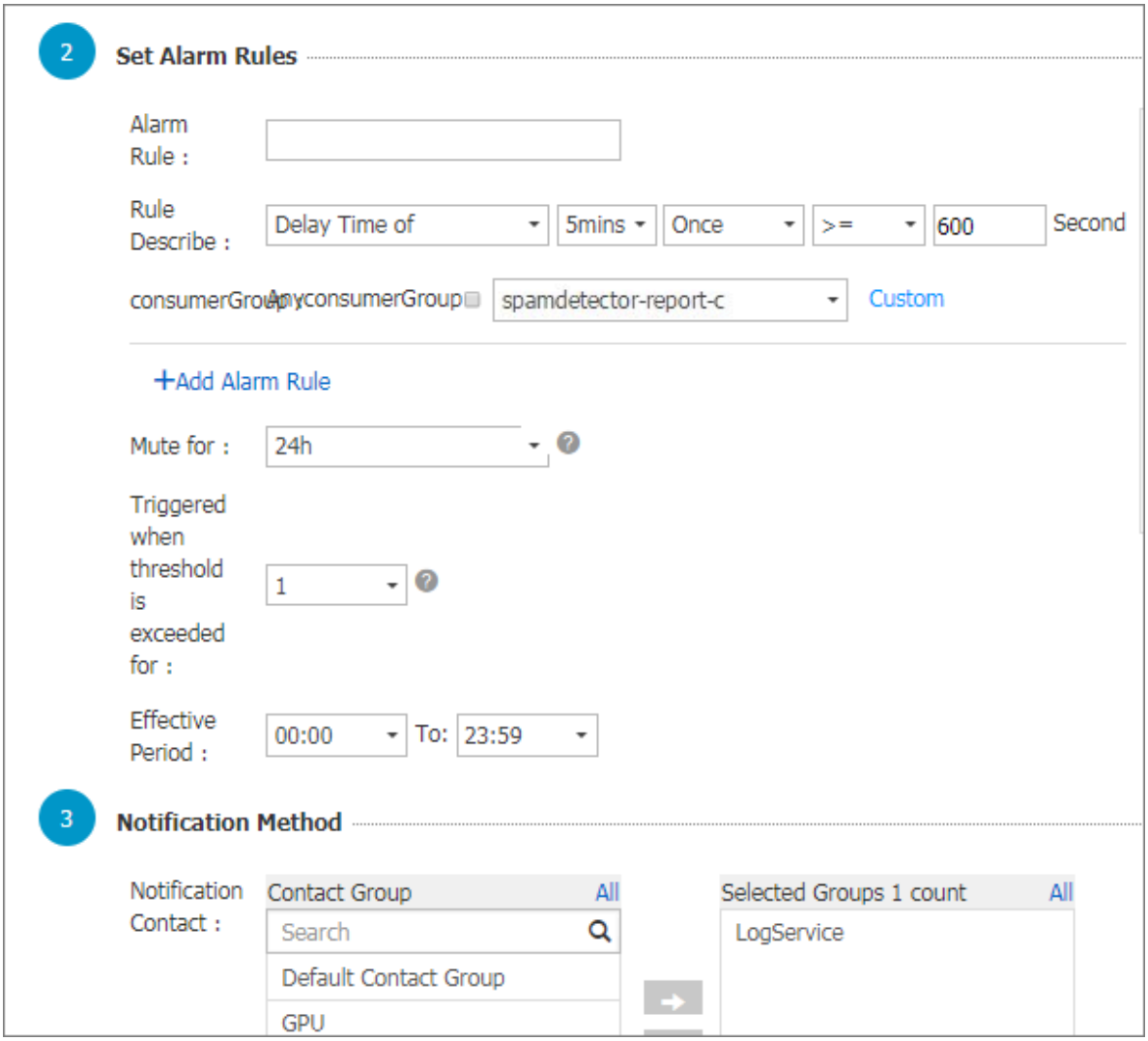
3. The figure shows the length, in seconds, of consumption, for all Java groups under logstore. which is measured in seconds. Click Create Alarm Rule in the upper-right corner to enter the Create Alarm Rule page.

Figure 9-6: Create an alarm rule for consumer group spamdetector-report-c.



4. The alarm is triggered if the latency within five minutes is greater than or equal to 600 seconds. Configure the Effective Period and Notification Contact, and then save the rule.

   Figure 9-7: Set an alarm rule



   Then, an alarm rule is created. If you have any questions about the configurations of alarm rules, open a ticket.

9.4 Use Fuction Compute to cosume LogHub Logs

9.4.1 Development guide

The data consumer terminal of Log Service *custom ETL function* is running on the Alibaba Cloud Function Compute service. You can use *function templates provided by Log Service* or user-defined functions according to different ETL purposes.

This document explains how to implement a user-defined Log Service ETL function.

Function event

The function event is a collection of input parameters used to run a function, and is in the format of a serialized JSON Object string.

Field descriptions

- jobName field

  The name of the Log Service ETL job. A Log Service trigger on the Function Compute service corresponds to a Log Service ETL job.

- taskId field

  For an ETL job, taskId is the identifier of a deterministic function call.

- cursorTime field

  The unix_timestamp when Log Service receives the last log of the data contained in this function call.

- source field

  This field is generated by Log Service. Log Service regularly triggers function This field is generated by Log Service. Log Service regularly triggers function execution based on the task interval defined in the ETL job. The source field is an important part of the function event. This field defines the data to be consumed by this function call.

  This data source range is composed of the following fields (for more information about the related field definitions, see *Log Service glossary*).

Field	Description
endpoint	The Service endpoint of the region where the Log Service project resides. *Service endpoint*
projectName	Project name
logstoreName	Logstore name
Shardid	Identifies a definite shard in the Logstore
beginCursor	The shard location from which to start consuming data

Field	Description
endCursor	The shard location where data consumption ends

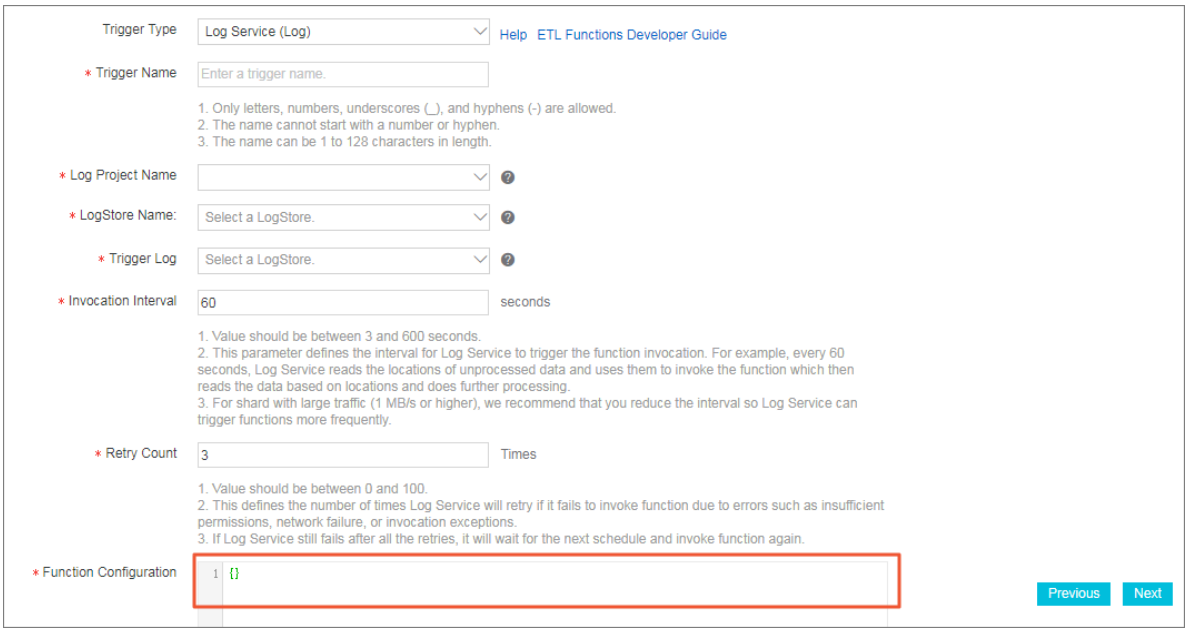
Note:
The [beginCursor, endCursor) of a shard is a left-closed and right-opened interval.

- parameter field

  This JSON Object field is set when you create the ETL job (Log Service trigger of Function Compute). This field is parsed during user-defined function operations to obtain runtime parameters required by the function.

  Set this field in the Function Configuration field when you create a Log Service trigger in the Function Compute console.

  Figure 9-8: Function configuration



Example of function event

```
{
    " source ": {
        " endpoint ": " http :// cn - shanghai - intranet . log .
 aliyuncs . com ",
        " projectNam  e ": " fc - 1584293594  28 ****",
        " logstoreNa  me ": " demo ",
        " shardId ":  0 ,
        " beginCurso  r ": " MTUwNTM5MD  I3NTY1ODcw  NzU2Ng ==",
        " endCursor ": " MTUwNTM5MD  I3NTY1ODcw  NzU2OA =="
    },
    " parameter ": {
```

```
        ...
    },
    " jobName ": " fedad35f51  a2a97b466d  a57fd71f31  5f539d2234 ",
    " taskId ": " 9bc06c96 - e364 - 4f41 - 85eb - b6e579214a  e4 ",
    " cursorTime ":  1511429883
}
```

When debugging a function, you can obtain the cursor by using the GetCursor API and manually assemble a function event for testing according to the preceding format .

Function development

You can implement functions by using many languages such as Java, Python, and Node.js. Log Service provides the corresponding runtime *SDKs in various languages* to facilitate function integration.

In this section, use Java 8 runtime as an example to show how to develop a Log Service ETL function. As this involves details of Java 8 function programming, read the *Java programming guide for Function Compute* first.

Java function Template

Currently, Log Service provides *user-defined ETL function templates* based on the Java 8 execution environment. You can use these templates to implement the custom requirements.

The templates have already implemented the following functions:

- Parse the source, taskId, and jobName fields in the function event.
- Use the *Log Service Java SDK* to pull data based on the data source defined in source and call the processData API to process each batch of data.

In the template, you must also implement the following functions:

- Use `UserDefine dFunctionP arameter . java` to parse the parameter field in the function event.
- Use the processData API of `UserDefine dFunction . java` to customize the data business logic in the function.
- Replace `UserDefine dFunction` with a name that properly describes your function.

processData method implementation

In processData, you must consume, process, and deliver the data batch according to your specific needs.

See*LogstoreReplication*, which reads data from one Logstore and writes it to another Log Service Logstore.

Notes

 Note:

1. If data is successfully processed by using processData, true is returned. If an exception occurs when data is processed and the exception persists after the retry , false is returned. However, in this case, the function continues to run and Log Service judges it as a successful ETL task, ignoring the incorrectly processed data
2. When a fatal error occurs or the business logic determines that function execution must be terminated prematurely, use the Throw Exception method to exit function execution. Log Service can detect a function operation exception and call function execution again based on the ETL job rules.

Instructions

- When shard traffic is high, configure sufficient memory for the function to prevent an abnormal termination because of function OOM.
- If time-consuming operations are performed in a function or shard traffic is high, set a short function trigger interval and long function operation timeout threshold.
- Grant sufficient permissions to function services. For example, to write Object Storage Service (OSS) data in the function, you must grant the OSS write permission to the function service.

ETL logs

- ETL scheduling logs

  Scheduling logs only record the start time and end time of the ETL task, whether or not the ETL task is successful, and the successfully returned information of the ETL task. If an ETL task encounters an error, it generates an ETL error log and sends an alert email or text message to the system administrator. When creating a

trigger, set the trigger log Logstore and activate the index query function for this Logstore.

Function execution statistics can be written out and returned by functions, such as the Java 8 function outputStream. The default template provided by Log Service writes a serialized JSON Object string. The string is recorded in the ETL task scheduling logs, which facilitates your statistics and query.

- ETL process logs

  This log records the key points and errors for each step in the ETL execution process, including step start and end times, initialization operation completion, and module error information. The ETL process log keeps you up to date on the ETL operation situation at all times. If an error occurs, you can immediately locate the cause in the process log.

  You can use context.getLogger() to record the process logs to the specific project and Logstore of Log Service. We recommend that you enable the index and query functions for this Logstore.

9.4.2 Configure Function Compute log consumption

Relying on the Function Compute service, Log Service provides a fully-hosted processing service for streaming data.

After configuring an ETL job, Log Service regularly retrieves updated data and triggers function execution, that is, incrementally consumes Log Service Logstore data to complete custom processing tasks in functions. Functions used to process data can be templates provided by Log Service or user-defined functions.

Applicable scenario

Data cleaning and processing

Log Service allows you to quickly collect, process, query, and analyze logs.

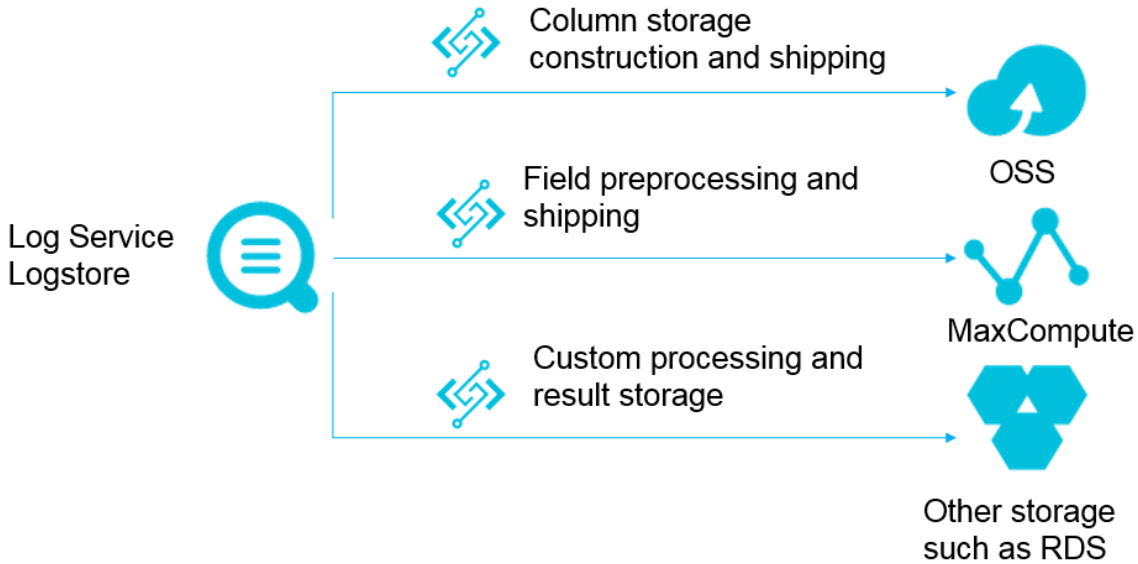
Figure 9-9: Data cleaning and processing



Data shipping

Log Service supports shipping data to the destination and constructs the data pipeline between cloud-based big data products.

Figure 9-10: Data shipping



Working principles

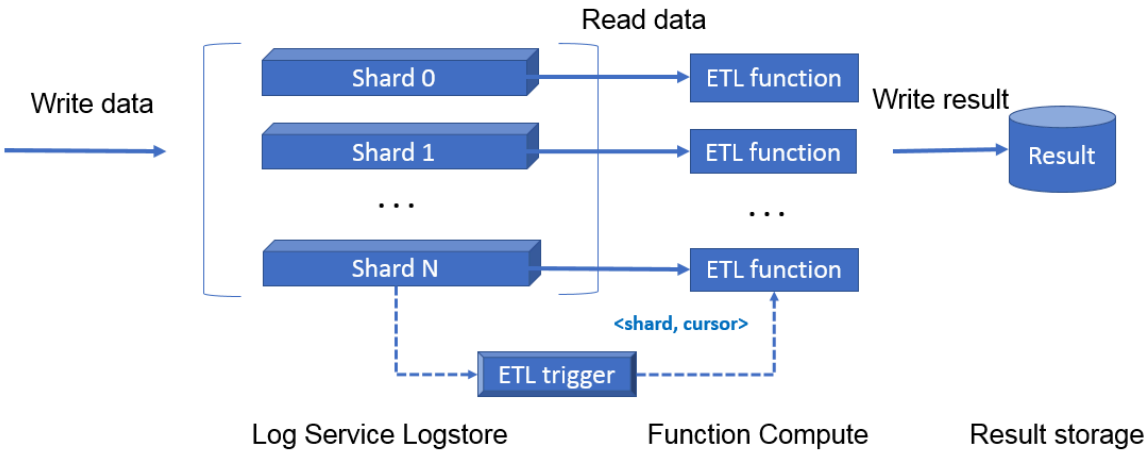
Trigger

A Log Service ETL job corresponds to a Function Compute trigger. After you create an ETL job, Log Service starts a timer based on the job configuration. The timer polls Logstore shard information. When a new log is written, the generated information

which is composed of three elements < shard_id, begin_cursor, end_cursor > serves as a function event and triggers function execution.

Log Service ETL job is triggered based on time. For example, if the ETL job trigger interval is 60 seconds and data is consistently written to shard 0 of the Logstore, the function execution is triggered every minute for shard 0. If no new data is written to shard 0, the function execution is not triggered. The input for function execution is the cursor interval for the last 60 seconds. In the function, shard 0 data is read based on the cursor and then processed.

Figure 9-11: Trigger



ETL functions

You can use the function templates or user-defined functions. Before you get started, we recommend that you learn about the Basic concepts of Function Compute services .

- Function templates maintained by Log Service

  Function templates are maintained on GitHub. Click *aliyun-log-fc-functions* to access the GitHub.

- User-defined functions

  Implement your own functions. The function configuration formats are related to the specific function implementations. For more information, see *Development guide for ETL function*.

User Guide

Step 1 Authorize Log Service and prepare resources

1. On the *quick authorization* page, click Confirm Authorization Policy to grant function trigger permission to Log Service.
2. Create a Log Service project and a Logstore for function process logs.

   If you have not created a project or a Logstore before, create one by following *Preparation* process.

   > Note:
   > Log Service project and Function Compute service must be in the same region.

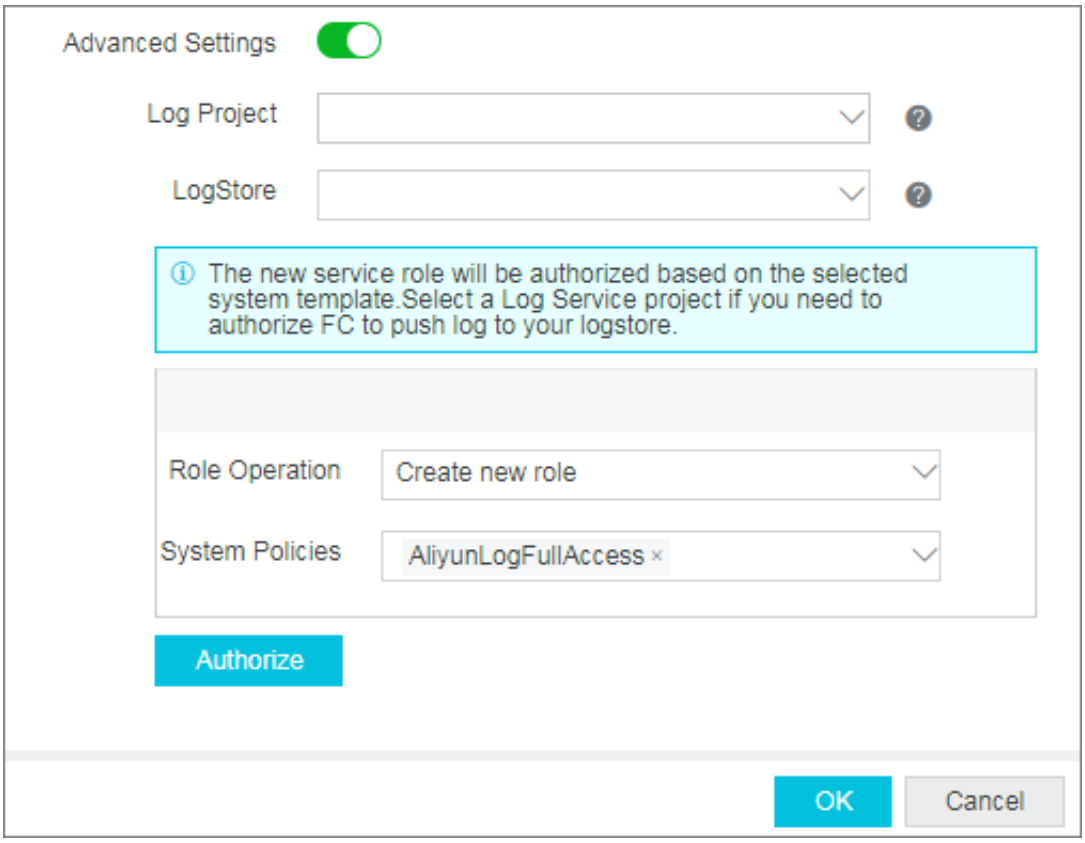
Step 2 Create a service

1. In the Function Compute console, click Create Service.
2. Enter the Service Name and Description. Turn on the Advanced Settings switch.

Configuration item	Meaning
Service name	The name of the Function Compute service to be created. Naming rules: · The name can contain uppercase letters, lowercase letters, numbers, hyphens (-), and underscores (_). · The name must begin with an uppercase letter, lowercase letter, or underscore (_). · The name is case sensitive and must contain 1–128 characters.
Feature description	The description of the new service.
Log project	The name of the Log Service project . The Logstore must be in the same region as the new Function Compute service.
Log repository	The name of the Log Service Logstore . The Logstore must be in the same region as the new Function Compute service.

Configuration item	Meaning
Role Operation	Create a service role and create the corresponding permissions based on the selected system template. Authorize Function Compute to push logs to the specified Logstore. You can create a new role or select an existing role. To use an existing role, you must select a role that already exists.
System Policies	Select a system authorization policy. Select the system authorization policies. Log Service supports two system authorization policies: `AliyunLogFullAccess` and `AliyunLogReadOnlyAccess`.

Figure 9-12: Create a service



After selecting the system authorization policy, click Authorize. The Role Templates page appears. Confirm your role information and permission information, including the Policy Name, Policy Description, and Policy Details. If you are creating a new role, you must confirm the Role Name and Role Description.

In the Policy Details, you can refine the authorization policy to customize an authorization policy suitable for this role.

After the successful authorization, click OK to go to the Overview page of the service.

Step 3 Create a function and a trigger

1. On the Overview page of the service, click Create Function.

   Select a function template.

   You can select a business template similar to your business model and modify it to create a function, or select a blank function template to customize the function.

   - Log Service template: Log Service template: Log Service provides the business templates `logstore_r eplication` and `oss - shipper - csv`. You can create a function and a trigger based on these templates.
   - Blank template: You can use the blank function template to create a blank function. Then, on the guide page, configure the trigger, function parameters, and write the relevant code to create a function.

2. Configure the trigger and then click Next.

   If you select a template provided by Log Service, you can configure the trigger directly. If you select the blank template, you must first select the trigger type and then configure the trigger.

   Complete the required items to configure the trigger, such as the trigger name, the project name, and the Logstore name. A Log Service type trigger of Function Compute corresponds to an ETL job of Log Service.

Configuration item	Meaning	Value
Trigger Name	The name of the new trigger.	The trigger name must be 1–256 bytes long and can contain English letters, numbers, underscores (_), and hyphens (-). It cannot start with a number or hyphen (-).

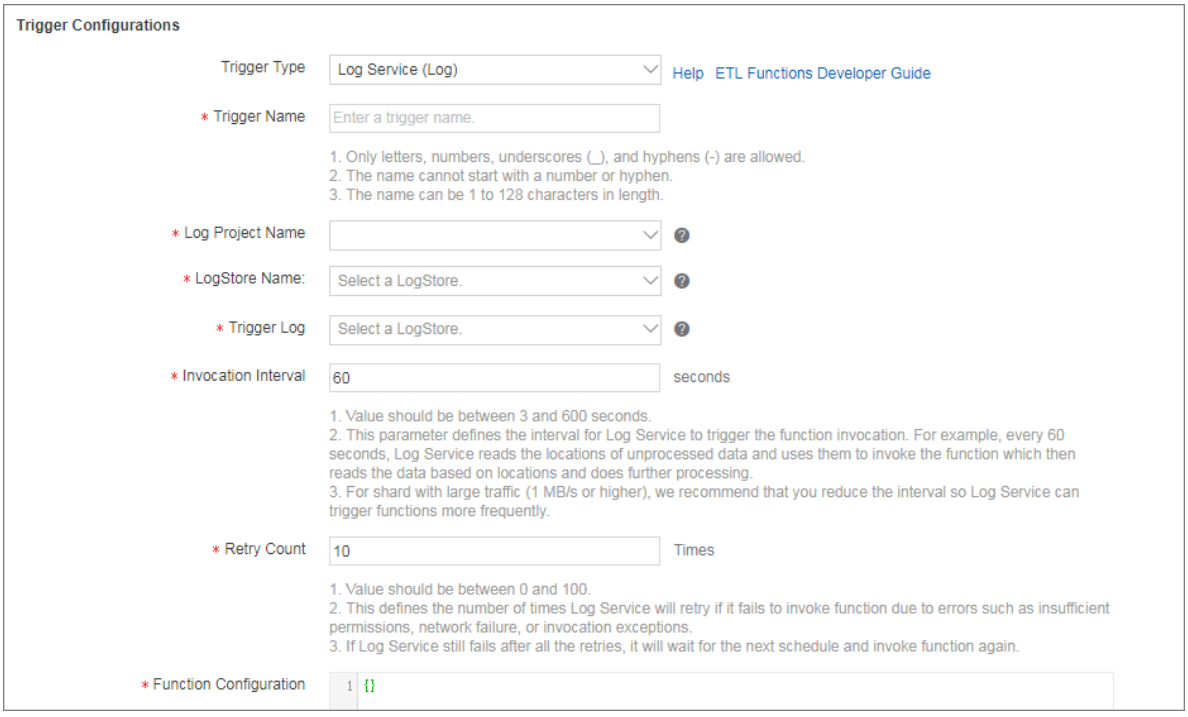
Configuration item	Meaning	Value
Project name	The name of the Log Service project.	It must be the name of an existing project. This project must be in the same region as your service.
Logstore name	The name of the Log Service project. This trigger regularly transmits the subscribed data of this Logstore to Function Compute for custom processing. You cannot change this parameter after the ETL job is created.	Select an existing Logstore and the Logstore must belong to the project selected in Log Project Name.
Trigger log	Log Service regularly triggers the function execution of Function Compute. Exceptions during the trigger process and function execution statistics are recorded in this Logstore. You can create an index for the Logstore for future viewing.	It must be the name of an existing Logstore and the Logstore must belong to the project selected in Log Project Name.

Configuration item	Meaning	Value
Invocation Interval	The interval at which Log Service triggers function execution. For example, when set to 60 seconds, Log Service reads the data location in the last 60 seconds for each Logstore shard, using this as a function event to call function execution. In the function, the user logic reads the shard data and performs computation. If the Logstore shards have a high traffic volume (over 1 Mbit/s), we recommend you set a shorter trigger interval to ensure the data volume processed by each function operation is of a reasonable size.	The value range is 3–600 seconds.

Configuration item	Meaning	Value
Retries	If an error occurs when Log Service triggers function execution according to the set trigger interval (such as insufficient permissions, network failure, or function execution return exception), this parameter sets the maximum number of times the function can be re-triggered. If the function is re-triggered the maximum number of times and the operation is still unsuccessful, the trigger interval must elapse before Log Service attempts to trigger the function execution again. The impact of reties on the business varies according to the specific function code implementation logic.	The value range is 0–100 times.

Configuration item	Meaning	Value
Function configuration	Log Service uses this configuration content as a part of the function event to pass into the function. The way in which this function is used is determined by the custom logic of the function. Different types of functions have different requirements for function configurations. For the vast majority of provided function templates, you must read the instructions when entering your parameters. When no parameters are passed in by default, enter: `{}`.	The configuration content must be a string in JSON Object format.

Figure 9-13: Trigger configuration



Note:

You already have the permissions to read/write Logstore data and allow Log Service to call your function.

3. Complete the basic configurations

   such as Function Name and Function Handler.Then, click Next.

4. Complete the function permissions.

   Confirm the template authorization and trigger role authorization. Then, click Next.

5. Review your Function Information and Trigger Information. Then, click Create.

View trigger logs

Log on to the Log Service console and create an index for the trigger log Logstore configured in the job. This allows you to view task execution statistics.

View function operation logs

Log on to the Log Service console to view detailed information in the function execution process. For more information, see *Logging*.

FAQs

I created a trigger, but it does not trigger function execution

1. Make sure you have used *quick authorization* to authorize Log Service to trigger function execution.
2. Make sure the data in the job's Logstore is incrementally modified, as function execution is triggered when shard data changes.
3. Log on to the Log Service console and check if any exceptions exist in the trigger logs and function operation logs.

9.5 Use Flink to consume LogHub logs

The Flink log connector is a tool provided by Alibaba Cloud Log Service and used to connect to Flink. It consists of two parts: consumer and producer.

The consumer reads data from Log Service. It supports the exactly-once syntax and shard-based load balancing.

The producer writes data into Log Service. When using the connector, you must add the Maven dependency to the project:

```
< dependency >
```

```
            < groupId > org . apache . flink </ groupId >
            < artifactId > flink - streaming - java_2 . 11 </
 artifactId >
            < version > 1 . 3 . 2 </ version >
</ dependency >
< dependency >
            < groupId > com . aliyun . openservic  es </ groupId >
            < artifactId > flink - log - connector </ artifactId >
            < version > 0 . 1 . 7 </ version >
</ dependency >
< dependency >
            < groupId > com . google . protobuf </ groupId >
            < artifactId > protobuf - java </ artifactId >
            < version > 2 . 5 . 0 </ version >
</ dependency >
 < dependency >
            < groupId > com . aliyun . openservic  es </ groupId >
            < artifactId > aliyun - log </ artifactId >
            < version > 0 . 6 . 19 </ version >
 </ dependency >
< dependency >
            < groupId > com . aliyun . openservic  es </ groupId >
            < artifactId > log - loghub - producer </ artifactId >
            < version > 0 . 1 . 8 </ version >
</ dependency >
```

Prerequisites

1. Access key is enabled and project and logstore have been created. For detailed instructions, see *Preparation*.
2. To use a sub-account to access Log Service, make sure that you have properly set the Resource Access Management (RAM) policies of Logstore. For more information, see *Grant RAM sub-accounts permissions to access Log Service*.

Log consumer

In the connector, the Flink log consumer provides the capability of subscribing to a specific LogStore in Log Service to achieve the exactly-once syntax. During use, you do not need to concern about the change of the number of shards in the LogStore.

Each sub-task in Flink consumes some shards in the LogStore. If shards in the LogStore are split or merged, shards consumed by the sub-task change accordingly.

Associated API

The Flink log consumer uses the following Alibaba Cloud Log Service APIs:

- Getcursorordata

  This API is used to pull data from a shard. If this API is frequently called, data may exceed the shard quota of Log Service. You can use ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS and

ConfigConstants.LOG_MAX_NUMBER_PER_FETCH to control the time interval of API calls and the number of logs pulled by each call. For more information about the shard quota, see *Shard*.

```
configProp  s . put ( ConfigCons  tants . LOG_FETCH_  DATA_INTER
VAL_MILLIS ,  " 100 ");
configProp  s . put ( ConfigCons  tants . LOG_MAX_NU  MBER_PER_F
ETCH ,  " 100 ");
```

- ListShards

  This API is used to obtain the list of all shards and shard status in a Logstore. If your shards are always split and merged, you can adjust the period of calling API to find shard changes in time.

```
//  Call   ListShards   every   30s
 configProp  s . put ( ConfigCons  tants . LOG_SHARDS  _DISCOVERY
 _INTERVAL_  MILLIS ,  " 30000 ")
```

- CreateConsumerGroup

  This API is called only when consumption progress monitoring is enabled. It is used to create a consumer group to synchronize the checkpoint.

- ConsumerGroupUpdateCheckPoint

  This API is used to synchronize snapshots of Flink to a ConsumerGroup of Log Service.

User Permission

The following table lists the RAM authorization policies required for sub-users to use the Flink log consumer.

Action	Resources
log:GetCursorOrData	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}
log:ListShards	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}
log:CreateConsumerGroup	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/*

Action	Resources
log:ConsumerGroupUpdateCheckPoint	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}

Configuration steps

1. Configure the startup parameter.

```
 Properties   configProp  s  =  new   Properties ();
//  Set   the   domain   to   access   Log   Service
 configProp  s . put ( ConfigCons  tants . LOG_ENDPOI  NT ,  " cn -
 hangzhou . log . aliyuncs . com ");
//  Set   the   AccessKey
 configProp  s . put ( ConfigCons  tants . LOG_ACCESS  SKEYID ,  "");
 configProp  s . put ( ConfigCons  tants . LOG_ACCESS  KEY ,  "");
//  Set   the   Log   Service   project
 configProp  s . put ( ConfigCons  tants . LOG_PROJEC  T ,  " ali - cn
 - hangzhou - sls - admin ");
//  Set   the   Log   Service   LogStore
 configProp  s . put ( ConfigCons  tants . LOG_LOGSTO  RE ,  "
 sls_consum  ergroup_lo  g ");
//  Set   the   start   position   to   consume   Log   Service
 configProp  s . put ( ConfigCons  tants . LOG_CONSUM  ER_BEGIN_P
 OSITION ,  Consts . LOG_END_CU  RSOR );
//  Set   the   message   deserializ  ation   method   for   Log
 Service
 RawLogGrou  pListDeser  ializer   deserializ  er  =  new
 RawLogGrou  pListDeser  ializer ();
 final   StreamExec  utionEnvir  onment   env  =  StreamExec
 utionEnvir  onment . getExecuti  onEnvironm  ent ();
 DataStream < RawLogGrou  pList >  logTestStr  eam  =  env . addSource
 (
        new   FlinkLogCo  nsumer < RawLogGrou  pList >( deserializ
 er ,  configProp  s ));
```

The preceding is a simple consumption example. As java.util.Properties is used as the configuration tool, configurations of all consumers can be located in ConfigConstants.



Note:

The number of sub-tasks in the Flink stream is independent from that of shards in the Log Service LogStore. If the number of shards is greater than that of sub-tasks, each sub-task consumes multiple shards exactly once. If the number of shards is smaller than that of sub-tasks, some sub-tasks are idle until new shards are generated.

2 Set consumption start position

You can set the start position for consuming a shard on the Flink log consumer. By setting ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, you can set whether to consume a shard from its header or tail or at a specific time. The values are as follows : The specific values are as follows:

- Consts.LOG_BEGIN_CURSOR: Indicates that the shard is consumed from its header , that is, from the earliest data of the shard.
- Consts.LOG_END_CURSOR: Indicates that the shard is consumed from its tail, that is, from the latest data of the shard.
- Constellation S. MAID: indicates that the checkpoint that is saved from a particular Java group starts to consume through configconstants. specify a specific locergroup.
- UnixTimestamp: A string of an integer value, which is expressed in seconds from 1970-01-01. It indicates that the shard is consumed from this time point.

Examples of the preceding three values are as follows:

```
configProp  s . put ( ConfigCons  tants . LOG_CONSUM  ER_BEGIN_P
OSITION ,  Consts . LOG_BEGIN_  CURSOR );
configProp  s . put ( ConfigCons  tants . LOG_CONSUM  ER_BEGIN_P
OSITION ,  Consts . LOG_END_CU  RSOR );
configProp  s . put ( ConfigCons  tants . LOG_CONSUM  ER_BEGIN_P
OSITION , " 1512439000 ");
configProp  s . put ( ConfigCons  tants . LOG_CONSUM  ER_BEGIN_P
OSITION ,  Consts . LOG_FROM_C  HECKPOINT );
```

 Note:

If you have set up recovery from the statebackend of flink itself when you start the flink task, then connector ignores the configuration above and uses checkpoint saved in statebackend.

3 set up consumer progress monitoring (optional)

The Flink log consumer supports consumption progress monitoring. The consumption progress is to obtain the real-time consumption position of each shard,

which is expressed in the timestamp. For more information, see *View consumer group status* and *Consumer group - Monitoring alarm*.

```
configProp  s . put ( ConfigCons  tants . LOG_CONSUM  ERGROUP , "
your  consumer  group  name ");
```



Note:

The preceding code is optional. If set, the consumer creates a consumer group first. If the consumer group already exists, no further operation is required. Snapshots in the consumer are automatically synchronized to the consumer group of Log Service. You can view the consumption progress of the consumer in the Log Service console.

4 Support disaster tolerance and exactly once syntax

If the checkpoint function of Flink is enabled, the Flink log consumer periodically stores the consumption progress of each shard. When a job fails, Flink resumes the log consumer and starts consumption from the latest checkpoint that is stored.

The period of writing checkpoint defines the maximum amount of data to be rolled back (that is, re-consumed) if a failure occurs. The code is as follows:

```
 final  StreamExec  utionEnvir  onment  env  =  StreamExec
 utionEnvir  onment . getExecuti  onEnvironm  ent ();
//  Enable  the  exactly - once  syntax  on  Flink
 env . getCheckpo  intConfig (). setCheckpo  intingMode ( Checkpoint
 ingMode . EXACTLY_ON  CE );
//  Store  the  checkpoint  every  5s
 env . enableChec  kpointing ( 5000 );
```

For more information about the Flink checkpoint, see the Flink official document *Checkpoints*.

Log Producer

The Flink log producer writes data into Alibaba Cloud Log Service.



Note:

The producer supports only the Flink at-least-once syntax. It means that when a job failure occurs, data written into Log Service may be duplicated but never lost.

User Permission

The producer uses the following APIs of Log Service to write data:

- Log: postlogstorelogs
- log:ListShards

If a RAM sub-user uses the producer, the preceding two APIs must be authorized.

Action	Resources
Log: postlogstorelogs	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/alert/${alarmName}
log:ListShards	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/alert/${alarmName}

Procedure

1. Initialize the producer.
   a. Initialize the configuration parameter Properties for the producer,

      which is similar to that for the consumer. The producer has some custom parameters. Generally, set these parameters to the default values. You can customize the values in special scenarios.

```
// The number of I / O threads used for sending
 data . The default value is 8 .
 ConfigCons tants . LOG_SENDER _IO_THREAD _COUNT
// The time when the log data is cached . The
 default value is 3000 .
 ConfigCons tants . LOG_PACKAG E_TIMEOUT_ MILLIS
// The number of logs in the cached package . The
   default value is 4096 .
 ConfigCons tants . LOG_LOGS_C OUNT_PER_P ACKAGE
// The size of the cached package . The default
 value is 3Mb .
 ConfigCons tants . LOG_LOGS_B YTES_PER_P ACKAGE
// The total memory size that the job can use .
 The default value is 100Mb .
```

```
ConfigCons tants . LOG_MEM_PO OL_BYTES
```

The preceding parameters are not mandatory. You can retain the default values.

b. Reload LogSerializationSchema to define the method for serializing data to RawLogGroup.

RawLogGroup is a collection of logs. For more information about the meaning of each field, see *Data model*.

To use the shardHashKey function of Log Service, specify the shard into which data is written. You can use LogPartitioner in the following way to generate the HashKey of data:

Example:

```
FlinkLogPr oducer < String > logProduce r = new
FlinkLogPr oducer < String >( new SimpleLogS erializer (),
configProp s );
logProduce r . setCustomP artitioner ( new LogPartiti oner <
String >() {
    // Generate a 32 - bit hash value
     public String getHashKey ( String element ) {
         try {
             MessageDig est md = MessageDig est .
getInstanc e (" MD5 ");
             md . update ( element . getBytes ());
             String hash = new BigInteger ( 1 , md .
digest ()). toString ( 16 );
             while ( hash . length () < 32 ) hash = " 0 " +
hash ;
             return hash ;
         } catch ( NoSuchAlgo rithmExcep tion e ) {
         }
          return " 0000000000 0000000000 0000000000
0000000000 0000000000 0000000000 0000 ";
     }
 });
```

Note:
LogPartitioner is optional. If this parameter is not set, data is randomly written into a shard.

2. The following usage example writes a string that is generated by simulation into Log Service:

```
// Serialize data to the data format of Log
 Service
 class SimpleLogS erializer implements LogSeriali
 zationSche ma < String > {
     public RawLogGrou p serialize ( String element ) {
         RawLogGrou p rlg = new RawLogGrou p ();
         RawLog rl = new RawLog ();
```

```
        rl . setTime (( int )( System . currentTim  eMillis () /
1000 ));
        rl . addContent (" message ",   element );
        rlg . addLog ( rl );
        return   rlg ;
    }
}
public   class   ProducerSa  mple  {
    public   static   String   sEndpoint  = " cn - hangzhou . log .
aliyuncs . com ";
    public   static   String   sAccessKey  Id  = "";
    public   static   String   sAccessKey  = "";
    public   static   String   sProject  = " ali - cn - hangzhou -
sls - admin ";
    public   static   String   sLogstore  = " test - flink -
producer ";
    private   static   final   Logger   LOG  =  LoggerFact  ory .
getLogger ( ConsumerSa  mple . class );
    public   static   void   main ( String []  args )  throws
Exception  {
        final   ParameterT  ool   params  =  ParameterT  ool .
fromArgs ( args );
        final   StreamExec  utionEnvir  onment   env  =
StreamExec  utionEnvir  onment . getExecuti  onEnvironm  ent ();
        env . getConfig (). setGlobalJ  obParamete  rs ( params );
        env . setParalle  lism ( 3 );
        DataStream < String >  simpleStri  ngStream  =  env .
addSource ( new   EventsGene  rator ());
        Properties   configProp  s  =  new   Properties ();
       //  Set   the   name   of   the   domain   used   to
access   Log   Service .
        configProp  s . put ( ConfigCons  tants . LOG_ENDPOI  NT
,   sEndpoint );
       //  Set   the   AccessKey   to   access   Log   Service
        configProp  s . put ( ConfigCons  tants . LOG_ACCESS
SKEYID ,   sAccessKey  Id );
        configProp  s . put ( ConfigCons  tants . LOG_ACCESS  KEY
,   sAccessKey );
       //  Set   the   Log   Service   project   into   which
logs   are   written
        configProp  s . put ( ConfigCons  tants . LOG_PROJEC  T ,
sProject );
       //  Set   the   Log   Service   LogStore   into   which
logs   are   written
        configProp  s . put ( ConfigCons  tants . LOG_LOGSTO  RE
,   sLogstore );
        FlinkLogPr  oducer < String >  logProduce  r  =  new
FlinkLogPr  oducer < String >( new   SimpleLogS  erializer (),
configProp  s );
        simpleStri  ngStream . addSink ( logProduce  r );
        env . execute (" flink   log   producer ");
    }
    //  Simulate   log   generation
    public   static   class   EventsGene  rator   implements
SourceFunc  tion < String > {
        private   boolean   running  =  true ;
       @ Override
        public   void   run ( SourceCont  ext < String >  ctx )
throws   Exception  {
            long   seq  =  0 ;
            while  ( running ) {
                Thread . sleep ( 10 );
                ctx . collect (( seq ++) + "-" +  RandomStri
ngUtils . randomAlph  abetic ( 12 ));
```

```
            }
        }
        @ Override
         public   void   cancel () {
             running  =  false ;
        }
    }
}
```

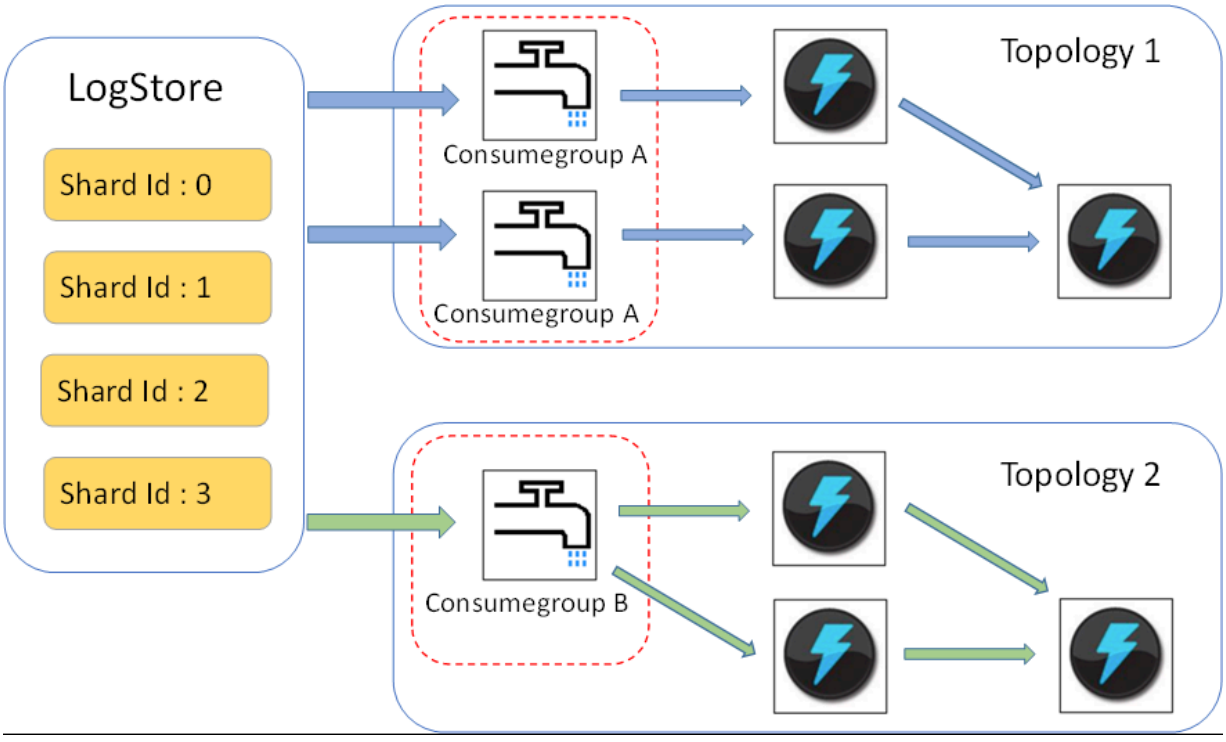
9.6 Use Storm to consume LogHub logs

LogHub of Log Service provides an efficient and reliable log channel for collecting log data in real time by using Logtail and SDKs. After collecting logs, you can consume the data written to LogHub by using real-time systems such as Spark Stream and Storm.

Log Service provides LogHub Storm spout to read data from LogHub in real time, reducing the cost of LogHub consumption for Storm users.

Basic architecture and process

Figure 9-14: Basic architecture and process



- In the preceding figure, the LogHub Storm spout is enclosed in the red dotted box. Each Storm topology has a group of spouts to read all the data from a Logstore. The spouts in different topologies are independent of each other.

- Each topology is identified by a unique LogHub consumer group name. Spouts in the same topology use the *Consumer Library* to achieve load balancing and automatic failover.
- Spouts read data from LogHub in real time, send data to the bolt nodes of the topology, and periodically save consumption endpoint as checkpoint to LogHub.

Limits

- To prevent misuse, each Logstore supports up to five consumer groups. You can use the DeleteConsumerGroup interface of the Java SDK to delete unused consumer groups.
- We recommend that the number of spouts is the same as that of shards. Otherwise, a single spout may not process a large amount of data.
- If a shard contains a large amount of data exceeding the processing capability of a single spout, you can use the shard split interface to split the shard and reduce the data volume of each shard.
- Dependency on the Storm ACK is required in LogHub spouts to confirm that spouts correctly send messages to bolts. Therefore, bolts must call ACK for confirmation.

Usage example

- Spout (used to build topology)

```
     public   static   void   main (  String []  args  )
   {
        String   mode  = " Local "; //  Use   the   local   test
mode .
          String   conumser_g  roup_name  = ""; //  Specify   a
unique   consumer   group   name   for   each   topology .  The
value   cannot   be   empty .  The   value   can   be   3 - 63
characters   long ,  contain   lowercase   letters ,  numbers ,
hyphens  (-),  and   underscore  s  ( _ ),  and   must   begin
and   end   with   a   lowercase   letter   or   number .
        String   project  = ""; //  The   Log   Service   project
.
        String   logstore  = ""; //  The   Log   Service   Logstore
.
        String   endpoint  = "";   //  Domain   of   the   Log
Service
        String   access_id  = "";  //  User ' s   access   key
         String   access_key  = "";
       //  Configurat  ions   required   for   building   a
LogHub   Storm   spout .
        Loghubspou  tconfig   Config  =  new   loghubspou  tconfig
( conumser_g  roup_name ,
                endpoint ,  project ,  logstore ,  access_id ,
                access_key ,  LogHubCurs  orPosition . END_CURSOR
);
        Topologybu  ilder   builder  =  new   topologybu  ilder
();
```

```
        //   loghub   storm   spout
         Loghubspou  t   spin  =  new  ( config  );
         //  The   number   of   spouts   can   be   the   same   as
  that   of   Logstore   shards   in   actual   scenarios .
         builder . setSpout (" spout ",  spout ,  1 );
         builder . setBolt (" exclaim ",  new   SampleBolt ()).
shuffleGro  uping (" spout ");
         Config   conf  =  new   Config ();
         conf . setDebug ( false );
         conf . setMaxSpou  tPending ( 1 );
        //  The   serializat  ion   method   LogGroupDa  taSerializ
Serializer   of   LogGroupDa  ta   must   be   configured
explicitly   when   Kryo   is   used   for   data   serializat  ion
  and   deserializ  ation .
         Config . registerSe  rializatio  n ( conf ,  LogGroupDa  ta
. class ,  LogGroupDa  taSerializ  Serializer . class );
         if  ( mode . equals (" Local ")) {
             logger . info (" Local   mode ...") ;
             LocalClust  er   cluster  =  new   LocalClust  er ();
             cluster . submitTopo  logy (" test - jstorm - spout ",
conf ,  builder . createTopo  logy ());
             try  {
                 Thread . sleep ( 6000  *  1000 ); // waiting   for
  several   minutes
             }  catch  ( Interrupte  dException   e ) {
                 //  TODO   Auto - generated   catch   block
                  e . printStack  Trace ();
             }
              cluster . killTopolo  gy (" test - jstorm - spout ");
              cluster . shutdown ();
         }  else   if  ( mode . equals (" Remote ")) {
              logger . info (" Remote   mode ...");
              conf . setNumWork  ers ( 2 );
              try  {
                  StormSubmi  tter . submitTopo  logy (" stt - jstorm
- spout - 4 ",  conf ,  builder . createTopo  logy ());
              }  catch  ( AlreadyAli  veExceptio  n   e ) {
                  //  TODO   Auto - generated   catch   block
                   e . printStack  Trace ();
              }  catch  ( InvalidTop  ologyExcep  tion   e ) {
                  //  TODO   Auto - generated   catch   block
                   e . printStack  Trace ();
              }
         }  else  {
              logger . error (" invalid   mode : " +  mode );
         }
     }
}
```

- **The following bolt code example consumes data and only prints the contents of each log.**

```
public   class   SampleBolt   extends   BaseRichBo  lt  {
    private   static   final   long   serialVers  ionUID  =
4752656887  774402264L ;
    private   static   final   Logger   logger  =  Logger .
getLogger ( BaseBasicB  olt . class );
    private   OutputColl  ector   mCollector ;
   @ Override
    public   void   prepare (@ SuppressWa  rnings (" rawtypes ")
Map   stormConf ,  TopologyCo  ntext   context ,
           OutputColl  ector   collector ) {
```

```
        mCollector  =  collector ;
    }
    @ Override
     public   void   execute ( Tuple   tuple ) {
        String   shardId  = ( String )  tuple
              . getValueBy  Field ( LogHubSpou  t . FIELD_SHAR
D_ID );
       @ SuppressWa  rnings (" unchecked ")
        List < LogGroupDa  ta >  logGroupDa  tas  = ( ArrayList
< LogGroupDa  ta >)  tuple . getValueBy  Field ( LogHubSpou  t .
FIELD_LOGG  ROUPS );
        for  ( LogGroupDa  ta   groupData  :  logGroupDa  tas ) {
           //  Each   LogGroup   consists   of   one   or   more
logs .
            LogGroup   logGroup  =  groupData . GetLogGrou  p ();
            for  ( Log   log  :  logGroup . getLogsLis  t ()) {
               StringBuil  der   sb  =  new   StringBuil  der ();
              //  Each   log   has   a   time   field   and
multiple   key :  value   pairs ,
               int   log_time  =  log . getTime ();
               sb . append (" LogTime :"). append ( log_time );
               for  ( Content   content  :  log . getContent
sList ()) {
                   sb . append ("\ t "). append ( content . getKey
()). append (":")
                           . append ( content . getValue ());
               }
               logger . info ( sb . toString ());
            }
       }
       //  The   dependency   on   the   Storm   ACK   mechanism
  is   mandatory   in   LogHub   spouts   to   confirm   that
spouts   send   messages   correctly
       //  to   bolts .  Therefore ,  bolts   must   call   ACK
for   such   confirmati  on .
        mCollector . ack ( tuple );
    }
    @ Override
     public   void   declareOut  putFields ( OutputFiel  dsDeclarer
  declarer ) {
       // do   nothing
    }
}
```

Maven

Use the following code for versions earlier than Storm 1.0 (for example, 0.9.6):

```
< dependency >
  < groupId > com . aliyun . openservic  es </ groupId >
  < artifactId > loghub - storm - spout </ artifactId >
  < version > 0 . 6 . 5 </ version >
</ dependency >
```

Use the following code for Storm 1.0 and later versions:

```
< dependency >
  < groupId > com . aliyun . openservic  es </ groupId >
  < artifactId > loghub - storm - 1 . 0 - spout </ artifactId >
  < version > 0 . 1 . 2 </ version >
```

```
</ dependency >
```

9.7 Use Spark Streaming to consume LogHub logs

E-MapReduce provides a set of universal interface to consume LogHub logs in real time by using Spark Streaming. For more information, see *GitHub*.

9.8 Use CloudMonitor to consume LogHub logs

CloudMonitor can directly consume Logstore data under LogHub to provide monitoring functions,

such as:

- Alarm on keywords in logs
- Statistics of QPS and RT in unit time
- Statistics of PV and UV in unit time

10 Data shipping

10.1 Overview

After you access a log source to Log Service, Log Service starts to collect logs in real time and allows you to consume and ship logs in the console or by using SDKs/APIs. Log Service can ship logs collected to LogHub to Alibaba Cloud storage products such as Object Storage Service (OSS) and Table Store in real time. You can configure to ship logs in the console and LogShipper provides a complete status API and automatic retry function.

Application scenarios

Interconnection with the data warehouse

Log source

The LogShipper function of Log Service ships logs that are collected to LogHub. After logs are generated, Log Service collects these logs in real time and ships them to other cloud products for storage and analysis.

Targets

- OSS (large-scale object storage)
  - *Ship logs to OSS*
  - Formats in OSS can be processed by using Hive. E-MapReduce is recommended.
- Table Store (NoSQL data storage service)
- Maxcompute (large data computing services):
  - Delivery via dataworks Data Integration-operation -*Ship data to MaxCompute by using DataWorks*

10.2 Ship logs to OSS

10.2.1 Ship logs to OSS

Log Service can automatically archive Logstore data to Object Storage Service (OSS) to achieve more functions of logs.

- OSS data supports lifecycle configuration for long-term log storage.

- You can consume OSS data by using self-built programs and other systems (for example, E-MapReduce).

Function advantages

Using Log Service to ship logs to OSS has the following advantages:

- Ease of use. You can configure to synchronize Logstore data of Log Service to OSS in the console.
- Improved efficiency. The log collection of Log Service centralizes logs of different machines, without repeatedly collecting logs from different machines to import to OSS.
- Ease of management Shipping logs to OSS can fully reuse the log grouping in Log Service. Logs in different projects and Logstores can be automatically shipped to different OSS bucket directories, which facilitates the OSS data management.

Prerequisites

1. Activate Log Service, create a project and Logstore, and collect log data.
2. Activate OSS, create a bucket in the region where the Log Service project resides.
3. Activate RAM access control.
4. The Log Service project and OSS bucket must be located in the same region. Cross-region data shipping is not supported.

Procedure

Step 1. Resource Access Management (RAM) authorization

Before you perform a shipping task, Log Service must be granted a permission to write to OSS .

Go to *RAM quick authorization* page, on the displayed page, click Agree to Authorize. After authorization is complete, Log Service has a corresponding write permission to OSS.



Note:

- For more information about how to modify the authorization policy and configure cross-account shipping task, see OSS Shipper - Advanced RAM authorization.
- For more information about how to authorize sub-account to perform a shipping task, see *Grant RAM sub-accounts permissions to access Log Service* to access Log Service.

Step 2. Configure an OSS shipping rule in Log Service

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. Select a Logstore, and click OSS in the left-side navigation pane.
4. Click Enable, set the OSS LogShipper configurations, and click Confirm.

   See the following table to complete the OSS shipping configurations.

Configuration item	Description	Value range
OSS Shipping Name	The name of the OSS shipping.	The name can be 3–63 characters long, contain lowercase letters, numbers, hyphens (-), and underscores (_), and must begin and end with a lowercase letter or number.
OSS Bucket	The name of the OSS bucket.	Must be an existing bucket name, and make sure the OSS bucket is in the same region as the Log Service project.
OSS Prefix	The prefix of OSS. Data synchronized from Log Service to OSS is stored in this bucket directory.	Must be an existing OSS prefix.
Partition Format	Use %Y, %m, %d, %H, and %M to format the creation time of the LogShipper task to generate the partition string. This defines the directory hierarchy of the object files written to OSS, where a forward slash (/) indicates a level of OSS directory. The following table describes how to define the OSS target file path by using OSS prefix and partition format.	For more information about formatting, see *Strptime API*.

Configuration item	Description	Value range
RAM Role	The Arn and name of the RAM role. The RAM role is used to control the access permissions and is the identity for the OSS bucket owner to create a role. The ARN of the RAM role can be viewed in the basic information of this role.	For example, `acs : ram :: 45643 : role / aliyunlogd efaultrole`.
Shipping Size	Automatically control the interval of creating LogShipper tasks and configure the maximum size of an OSS object (not compressed).	The value range is 5–256. The unit is MB.
Storage Format	The storage format after log data is shipped to OSS.	Three formats are supported (*JSON storage*, *Parquet storage*, and *CSV storage*).
Compression	The compression method of OSS data storage.	· Do Not Compress: The raw data is not compressed. · Compress (snappy): Use *snappy* algorithm to compress data, reducing the usage of OSS bucket storage space.

Configuration item	Description	Value range
Shipping Time	The time interval between LogShipper tasks.	The default value is 300. The value range is 300–900 . The unit is second.

Figure 10-1: Delivery log

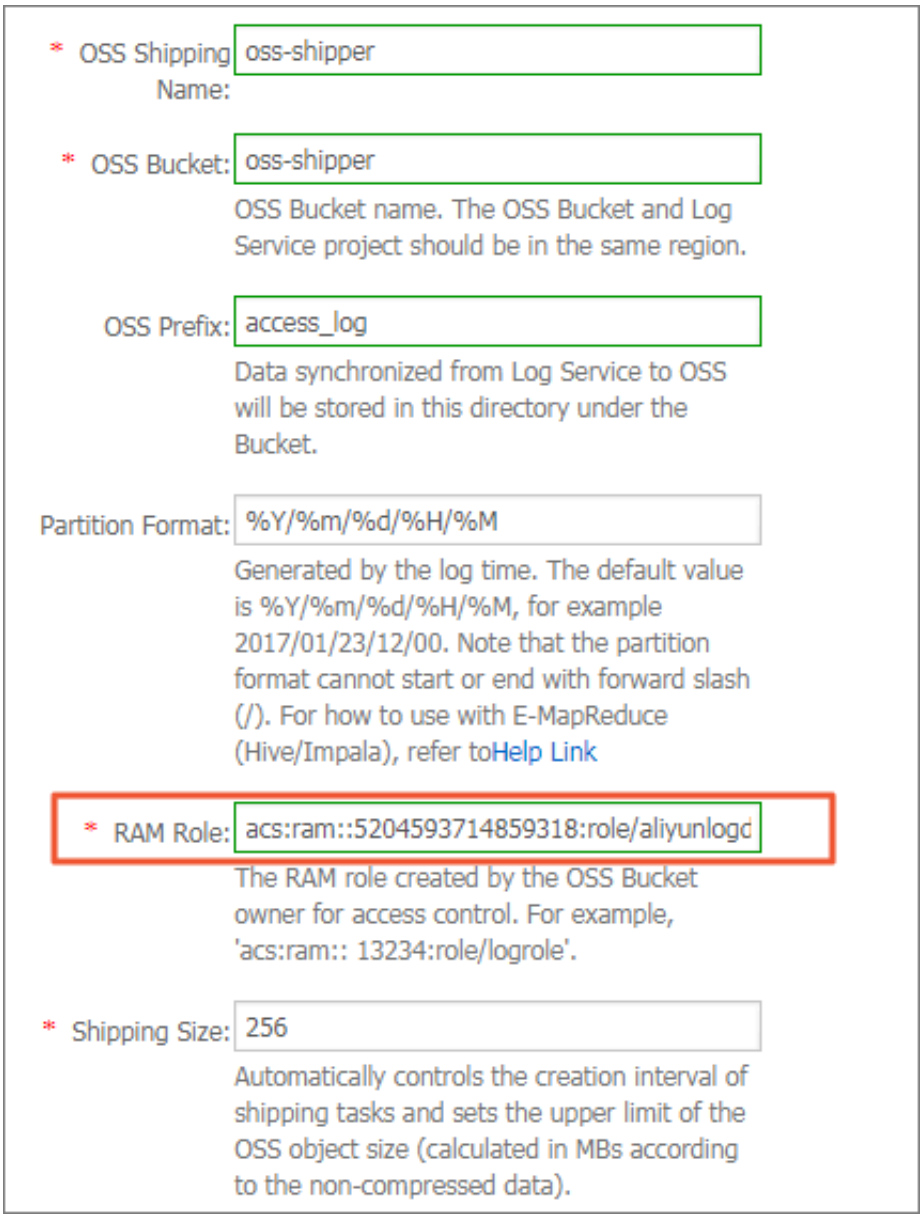
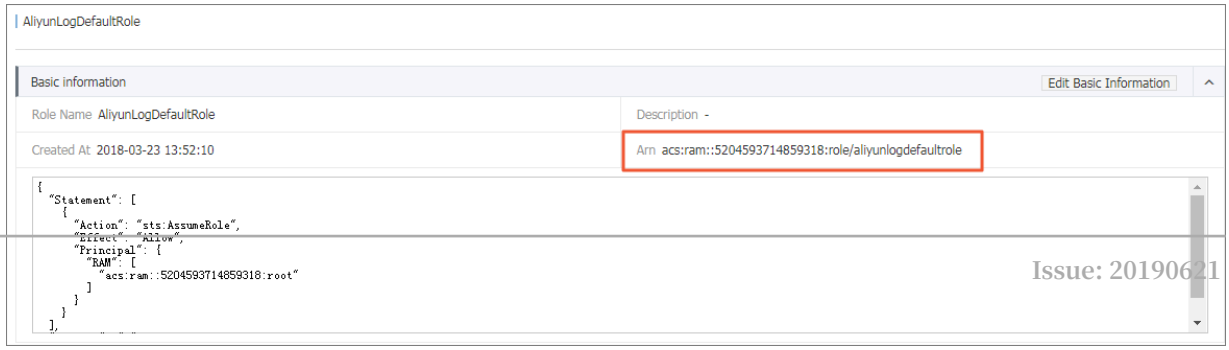


Figure 10-2: Role arn



Note:
Log Service concurrently implements data shipping at the backend. Large amounts of data may be processed by multiple shipping threads. Each shipping thread jointly determines the frequency of task generation based on the size and time. When any condition is met, the shipping thread creates the task.

Partition format

Each LogShipper task is written into an OSS file, with the path format of `oss://OSS-BUCKET/OSS-PREFIX/PARTITION-FROMAT_RAN DOM-ID`. Use the LogShipper task created at 2017-01-20 19:50:43 as an example to describe how to use the partition format.

OSS Bucket	OSS Prefix	Partition format	OSS file path
test-bucket	test-table	%Y/%m/%d/%H/%M	`oss://test-bucket/test-table/2017/01/20/19/50/43_1484913043351525351_2850008`
test-bucket	log_ship_oss_example	year=%Y/mon=%m/day=%d/log_%H%M%s	`oss://test-bucket/log_ship_oss_example/year=2017/mon=01/day=20/log_195043_1484913043351525351_2850008.parquet`

OSS Bucket	OSS Prefix	Partition format	OSS file path
test-bucket	log_ship_oss_example	ds=%Y%m%d/%H	oss :// test - bucket / log_ship_oss_example / ds = 20170120 / 19_1484913043351525351_2850008 . snappy
test-bucket	log_ship_oss_example	%Y%m%d/	oss :// test - bucket / log_ship_oss_example / 20170120 / _148491304 3351525351 _2850008
test-bucket	log_ship_oss_example	%Y%m%d%H	oss :// test - bucket / log_ship_oss_example / 2017012019 _148491304 3351525351 _2850008

Analyze the OSS data by using big data platforms such as Hive and MaxCompute. To use the partition data, set each level of directory to key=value format (Hive-style partition).

For example, oss://test-bucket/log_ship_oss_example/year=2017/mon=01/day=20/log_195043_1484913043351525351_2850008.

parquet can be set to three levels of partition columns: year, month, and day.

LogShipper tasks management

After the LogShipper function is enabled, Log Service regularly starts the LogShipper tasks in the backend. You can view the status of the LogShipper tasks in the console. With LogShipper tasks management, you can:

View all the LogShipper tasks

- in the last two days and check their status. The status of a LogShipper task can be Success, Failed, and Running. The status Failed indicates that the LogShipper task has encountered an error because of external reasons and cannot be retried. In this case, you must manually solve the problem.
- For the failed LogShipper tasks created within two days, you can view the external reasons that cause the failure in the task list. After fixing the external errors, you can retry all the failed tasks separately or in batches.

Procedure

1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. Select a Logstore, and click OSS in the left-side navigation pane.

   You can view the information such as task start time, task end time, time when logs are received, data lines, and task status.

   If the LogShipper task fails, a corresponding error message is displayed in the console. The system retries the task based on the policy by default. You can also manually retry the task.

Retry a task

Generally, log data is synchronized to OSS within 30 minutes after being written to the Logstore.

By default, Log Service retries the tasks in the last two days based on the annealing policy. The minimum interval for retry is 15 minutes. A task that has failed once can be retried in 15 minutes, a task that has failed twice can be retried in 30 minutes (2 x 15 minutes), and a task that has failed three times can be retried in 60 minutes (2 x 30 minutes).

To immediately retry a failed task, click Retry All Failed Tasks in the console or specify a task and retry it by using APIs/SDKs.

Failed tasks errors

See the following common errors that cause the task failure.

Error Message	Error cause	Handling method
UnAuthorized	No permission.	Make sure that: - The OSS user has created a role. - The account ID in the role description is correct. - The role has been granted the permissions of writing OSS buckets. - The role-arn is correctly configured.
ConfigNotExist	The configuration does not exist.	This error is generally caused by the deletion of a shipping rule. Retry the task after reconfiguring the shipping rule.
InvalidOssBucket	The OSS bucket does not exist.	Make sure that: The OSS bucket is in the same region as the Log Service project. The bucket name is correctly configured
InternalServerError	The internal error of Log Service.	Retry the task.

OSS data storage

You can access the OSS data in the console or by using APIs/SDKs.

To access OSS data in the console, log on to the OSS console, click a bucket name in the left-side navigation pane. For more information about OSS, see OSS documentation.

For more information about OSS, see OSS documentation.

Object Address

```
oss ://  OSS - BUCKET / OSS - PREFIX / PARTITION - FROMAT_RAN  DOM -
ID
```

- Descriptions of path fields
  - OSS-BUCKET and OSS-PREFIX indicate the OSS bucket name and directory prefix respectively, and are configured by the user. INCREMENTID is a random number added by the system.
  - PARTITION-FORMAT is defined as %Y/%m/%d/%H/%M, where %Y, %m, %d, %H, and %M indicate year, month, day, hour, and minute respectively. They are

obtained by using strptime API to calculate the created time of the LogShipper task in Log Service.

- RANDOM-ID is the unique identifier of a LogShipper task.

· Directory time

The OSS data directory is configured according to the created time of LogShipper tasks. Assume that the data is shipped to OSS every five minutes. The LogShipper task created at 2016-06-23 00:00:00 ships the data that is written to Log Service after 2016-06-22 23:55. To analyze the complete logs of the full day of 2016-06-22, in addition to all objects in the `2016 / 06 / 23 / 00 /` directory, you must check whether the objects in the first 10 minutes in the 2016/06/23/00/directory contain the log of 2016-06-22.

Object storage format

· JSON

For more information, see*JSON storage*.

· Parquet

For more information, see*Parquet storage*.

· CSV

For more information, see*CSV storage*.

10.2.2 JSON storage

This document introduces the configurations about JSON storage for Log Service logs that are shipped to Object Storage Service (OSS). For more information about shipping logs to OSS, see *Ship logs to OSS*.

The compression types and file addresses of OSS files are as follows.

Compression type	File suffix	Example of OSS file address
Do Not Compress	None	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937
snappy	.snappy	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.snappy

Do Not Compress

An object is combined by multiple logs. Each line of the file is a log in the JSON format . See the following example:

```
{" __time__ ": 1453809242 ," __topic__ ":"," __source__ ":" 10 . 170
 . ***.***"," ip ":" 10 . 200 . **.***"," time ":" 26 / Jan / 2016 :
 19 : 54 : 02  + 0800 ," url ":" POST
              / PutData ?  Category = YunOsAccou  ntOpLog &
 AccessKeyI  d =< yourAccess  KeyId >& Date = Fri % 2C % 2028 % 20Jun
 % 202013 % 2006 % 3A53 % 3A30 % 20GMT & Topic = raw & Signature =<
 yourSignat  ure >
               HTTP / 1 . 1 ," status ":" 200 ," user - agent ":"
 aliyun - sdk - java "}
```

Compress (snappy)

Use *Snappy* C++ (Snappy.Compress method) to compress the data in none format at the file level. You can obtain the file in none format after decompress the .snappy file. You can obtain the file in none format after decompress the `. snappy` file.

Decompressing through C++ Lib

Download Lib from *Snappy official website* and use the Snappy.Uncompress method to decompress the .snappy file.

Java Lib

[xerial snappy-java], use Snappy.Uncompress or Snappy.SnappyInputStream (SnappyFramedInputStream not supported).

```
< dependency >
< groupId > org . xerial . snappy </ groupId >
< artifactId > snappy - java </ artifactId >
< version > 1 . 0 . 4 . 1 </ version >
< type > jar </ type >
< scope > compile </ scope >
</ dependency >
```

 **Note:**

Version 1.1.2.1 may not decompress parts of the compressed file because of a *bug* , which is

Snappy.Uncompress

```
 String   fileName  = " C :\\ My   download \\ 36_1474212  9631886006
 84_4451886 . snappy ";
 RandomAcce  ssFile   randomFile  =  new   RandomAcce  ssFile (
 fileName , " r ");
 int   fileLength  = ( int )  randomFile . length ();
 randomFile . seek ( 0 );
 byte []  bytes  =  new   byte [ fileLength ];
 int   byteread  =  randomFile . read ( bytes );
 System . out . println ( fileLength );
```

```
System . out . println ( byteread );
byte []  uncompress  ed  =  Snappy . uncompress ( bytes );
String   result  =  new   String ( uncompress  ed , " UTF - 8 ");
System . out . println ( result );
```

Snappy.SnappyInputStream

```
String   fileName  = " C :\\ My   download \\ 36_1474212  9631886006
84_4451886 . snappy ";
SnappyInpu  tStream   sis  =  new   SnappyInpu  tStream ( new
FileInputS  tream ( fileName ));
byte []  buffer  =  new   byte [ 4096 ];
int   len  =  0 ;
while  (( len  =  sis . read ( buffer )) ! = - 1 ) {
    System . out . println ( new   String ( buffer ,  0 ,  len ));
}
```

Unzipping tool under Linux environment

For Linux environment, a tool used to decompress .snappy file is provided. Click to download the *snappy_tool*.

```
./ snappy_too  l   03_1453457  0065480787  22_44148 . snappy
 03_1453457  0065480787  22_44148
 compressed . size :  2217186
 snappy :: Uncompress   return :  1
 uncompress  ed . size :  25223660
```

10.2.3 CSV storage

This document introduces the configurations about CSV storage for Log Service logs that are shipped to Object Storage Service (OSS). For more information about shipping logs to OSS, see *Ship logs to OSS*.

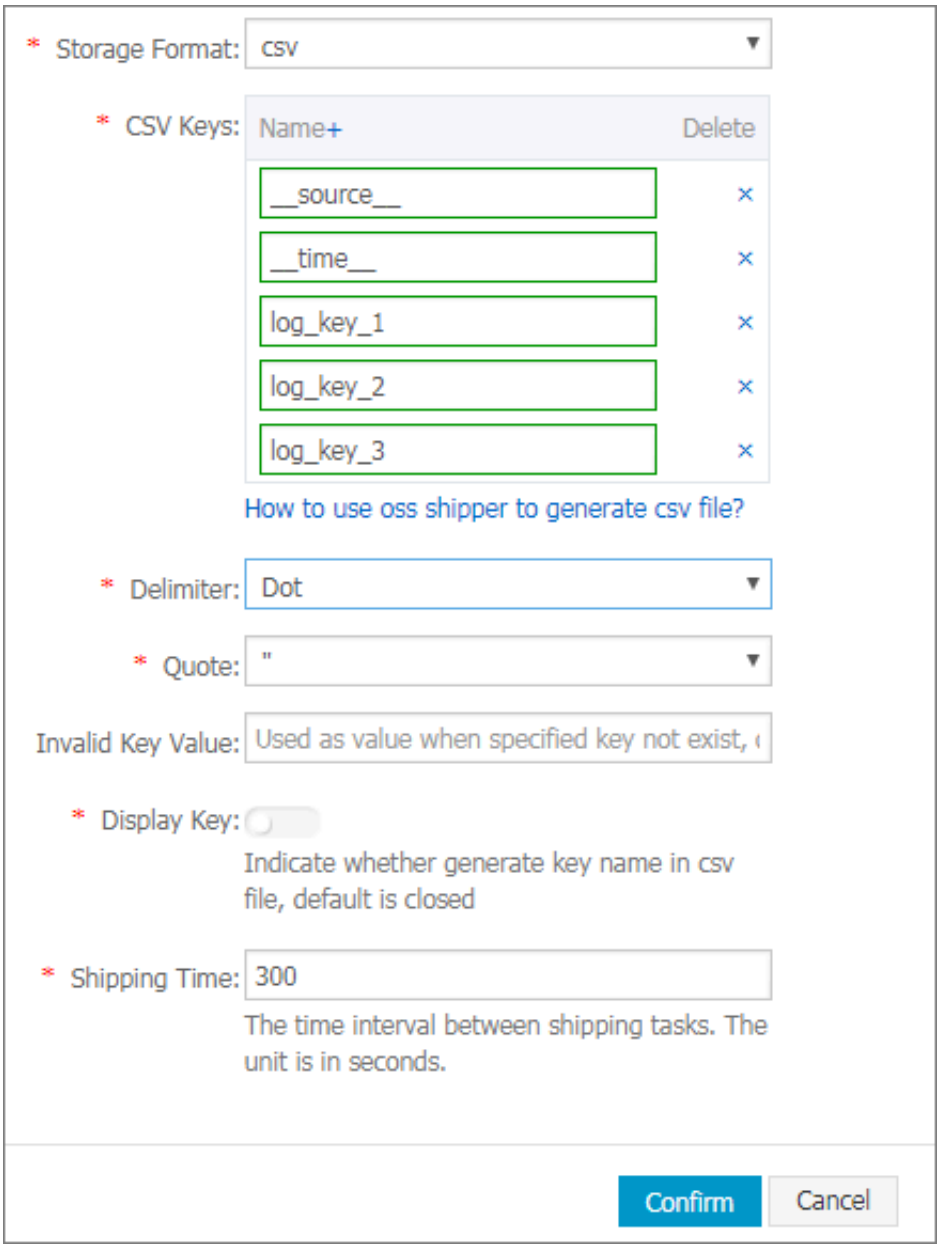
Configure CSV storage fields

Configuration page

You can view multiple key-value pairs of one log on the Log Service data preview page or index query page. Enter the field names (keys) you want to ship to OSS in sequence .

If the key name you entered cannot be found in the log, the corresponding column is set to null.

Figure 10-3: Configuration item



Configuration item

Configuration item	Value	Note
Delimiter	character	A one-character string used to separate different fields.

Configuration item	Value	Note
Quote	character	A one-character string. If a field contains a delimiter or a line break, use quote to enclose this field to avoid incorrect field separation in data reading.
Escape	character	A one-character string . The default settings are the same as those of quote. Modification is not supported currently. If a field contains a quote (used as a regular character instead of an escape character), an escape character must be added before this quote.
Invalid Key Value	string	If the specified key value does not exist, this string is entered in the field to indicate the field is null.
Display Key header	boolean	Indicates whether or not to add the field name to the first line of the CSV file.

For more information, see *CSV* standard and *postgresql CSV description*.

Configurable reserved fields

Besides the key-value pairs of the log, Log Service also provides the following optional reserved fields when shipping logs to OSS.

Reserved field	Description
`__time__`	The UNIX timestamp of a log (the number of seconds since 1970-01-01), which is calculated according to the time field of your log.
`__topic__`	The log topic.
`__source__`	The IP address of the client from which a log comes.

The preceding fields are included by default in JSON storage.

You can select which fields you want to include in the CSV storage as per your needs. For example, you can enter the field name `__topic__` if you need the log topic.

OSS storage address

Compression type	File suffix	Example of OSS file address
Do Not Compress	.csv	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.csv
snappy	.snappy.csv	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.snappy.csv

Consume data

HybridDB

We recommend that you configure as follows:

- Delimiter: comma (,)
- Quote: double quotation marks (“)
- Invalid Key Value: empty
- Display Key: not selected (no field name in the first line of the CSV file for HybridDB by default)

For more information, see HybridDB document.

CSV is a readable format, which means that a file in CSV format can be directly downloaded from OSS and viewed in text form.

If Compress (snappy) is used as the compression type, see the decompression descriptions of snappy in *JSON storage*.

10.2.4 Parquet storage

This document introduces the configurations about Parquet storage for Log Service logs that are shipped to Object Storage Service (OSS). For more information about shipping logs to OSS, see *Ship logs to OSS*.

Configure Parquet storage fields

Data types

The Parquet supports the storage in six formats, including string, boolean, int32, int64, float, and double.

Log Service data will be converted from strings into the target Parquet type during log shipping. If any data fails to be converted into a non-string type, the corresponding column is filled with null.

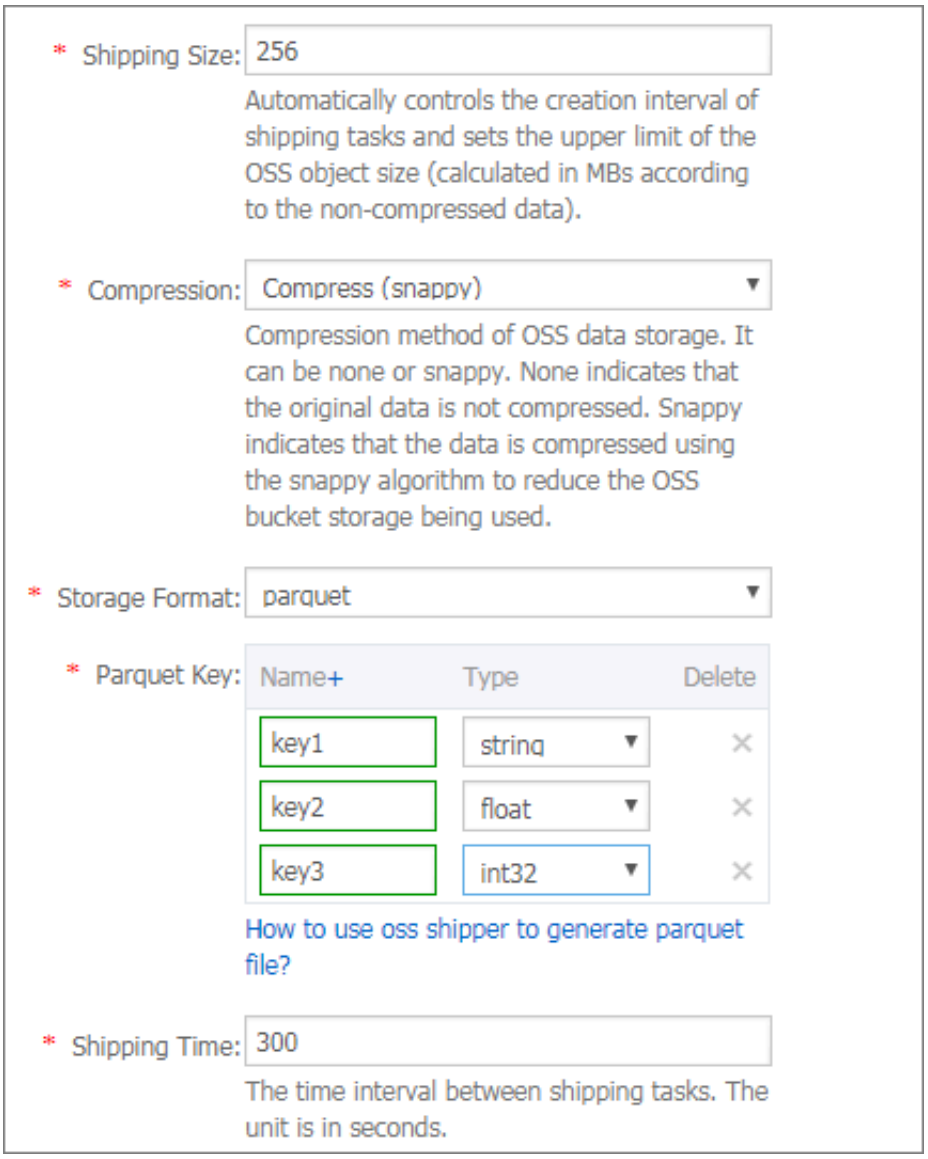
Configure columns

Configure the Log Service data field names and the target data types required by Parquet. Parquet data is organized according to this field order when being shipped. The Log Service field names are used as the Parquet data column names. The data column is set to null if:

- This field name does not exist in Log Service data.

- This field fails to be converted from a string to a non-string (such as double and int64).

Figure 10-4: Field Configuration



Configurable reserved fields

Besides the key-values of the log, the Log Service also provides the following optional reserved fields for the shipping to OSS:

Reserved field	Description
`__time__`	The UNIX timestamp of a log (the number of seconds since 1970-01-01), which is calculated according to the time field of your log.

Reserved field	Description
`__topic__`	The log topic.
`__source__`	The IP address of the client from which a log comes.

The preceding fields are carried by default in JSON storage.

You can select which fields you want to include in the Parquet or CSV storage as per your needs. For example, you can enter the field name __topic__ and select string as the type if you need the log topic.

OSS storage address

Compression type	File suffix	Example of OSS file address
Do Not Compress	.parquet	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.parquet
snappy	.snappy.parquet	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.snappy.parquet

Consume data

E-MapReduce/Spark/Hive

See *community document*.

Stand-alone verification tool

The *parquet-tools* provided by the open-source community is used to verify the Parquet format, view schema, and read data at the file level.

You can compile this tool by yourself or click *Download* to download the version provided by Log Service.

- View the schema of the Parquet file

```
$ java -jar parquet-tools-1.6.0rc3-SNAPSHOT.jar
schema -d 00_1490803_5321364704_39_124353.snappy.parquet
| head -n 30
message schema {
  optional int32 __time__;
  optional binary ip;
  optional binary __source__;
  optional binary method;
  optional binary __topic__;
  optional double seq;
```

```
   optional   int64   status ;
   optional   binary   time ;
   optional   binary   url ;
   optional   boolean   ua ;
}
 creator :  parquet - cpp   version   1 . 0 . 0
 file   schema :  schema
--------------------------------------------------------------------------------
 __time__ :  OPTIONAL   INT32   R : 0   D : 1
 ip :  OPTIONAL   BINARY   R : 0   D : 1
.......
```

- View all contents of the Parquet file

```
$  java  - jar   parquet - tools - 1 . 6 . 0rc3 - SNAPSHOT . jar
   head  - n   2   00_1490803  5321364704  39_124353 . snappy .
 parquet
 __time__  =  1490803230
 ip  =  10 . 200 . 98 . 220
 __source__  = *. *. *.*
 method  =  POST
 __topic__  =
 seq  =  1667821 . 0
 status  =  200
 time  =  30 / Mar / 2017 : 00 : 00 : 30  + 0800
 url  = / PutData ?  Category = YunOsAccou  ntOpLog & AccessKeyI
 d =************& Date = Fri % 2C % 2028 % 20Jun % 202013
 % 2006 % 3A53 % 3A30 % 20GMT & Topic = raw & Signature
 =*******************************  HTTP / 1 . 1
 __time__  =  1490803230
 ip  =  10 . 200 . 98 . 220
 __source__  = *. *. *.*
 method  =  POST
 __topic__  =
 seq  =  1667822 . 0
 status  =  200
 time  =  30 / Mar / 2017 : 00 : 00 : 30  + 0800
 url  = / PutData ?  Category = YunOsAccou  ntOpLog & AccessKeyI
 d =************& Date = Fri % 2C % 2028 % 20Jun % 202013
 % 2006 % 3A53 % 3A30 % 20GMT & Topic = raw & Signature
 =*******************************  HTTP / 1 . 1
```

For more operation instructions, run the j `java - jar parquet - tools - 1 . 6 . 0rc3 - SNAPSHOT . jar - h` command.

10.2.5 Advanced RAM authorization

Before perform the OSS shipping task, the owner of the OSS bucket must configure *quick authorization*. After the authorization is complete, Log Service of the current account has the permission to write to OSS bucket.

This document describes the RAM authorization for OSS shipping tasks in different scenarios.

- If you need more fine-grained access control for OSS buckets, see *Modify the authorization policy*.

- If a Log Service project and OSS bucket are not created with the same Alibaba Cloud account, see *Cross-account shipping*.
- If a sub-account must ship log data to OSS bucket that belongs to another Alibaba Cloud account, see *Shipping between sub-account and main account*.
- If a sub-account must ship log data of the current main account to the OSS bucket of the same account, see *Grant RAM sub-accounts permissions to access Log Service*.

Modify the authorization policy

After *quick authorization*, the role AliyunLogDefaultRole is granted to AliyunLogRolePolicy by default, and has write permission for all OSS buckets of account B.

If you need more fine-grained access control, revoke the AliyunLogRolePolicy authorization from the AliyunLogDefaultRole. See *OSS authorization* to create a more fine-grained permission policy, and authorize the AliyunLogDefaultRole.

Cross-account shipping

If your Log Service project and OSS bucket are not created with the same Alibaba Cloud account, you must configure the authorization policy in following way.

For example, Log Service data of the account A must be shipped to the OSS bucket created by the account B.

1. Using *quick authorization* account B creates the role AliyunLogDefaultRole, and grants write permission to OSS.
2. In the RAM console, click Role Management on the left-side navigation pane. Then, select AliyunLogDefaultRole, and click the role name to see the basic information.

   In the role description, `Service` configuration indicates the legal user of the role. For example, `log . aliyuncs . com` indicates that the current account can obtain the role to get OSS write permission.
3. In `Service` configuration, you can modify the role description to add `A_ALIYUN_I  D @ log . aliyuncs . com` . ID of the main account A can be viewed in the Account Management > Security Settings.

   For example, ID of the account A is 165421896534****, and modified description is as follows:

```
{
" Statement ": [
```

```
 {
   " Action ": " sts : AssumeRole ",
   " Effect ": " Allow ",
   " Principal ": {
     " Service ": [
       " 1654218965  34 ****@ log . aliyuncs . com ",
       " log . aliyuncs . com "
     ]
   }
 }
],
" Version ": " 1 "
}
```

This role description indicates that account A has the permission to use Log Service to obtain the temporary token to operate the resources of the account B. For more information about the role description, see *Authorization policy management*.

4. The account A creates a shipping task. When configuring the task, RAM role column must be filled with the RAM role identifier ARN of the OSS bucket owner, that is, the RAM role AliyunLogDefaultRole created by account B.

   The ARN of the RAM role can be viewed in the basic information. The format is as follows: `acs : ram :: 13234 : role / logrole` .

Shipping between sub-account and main account

If the sub-account a_1 of the main account A must use this role to create a shipping rule to ship logs to the OSS bucket of the account B. In this case, the main account A must grant the PassRole permission to the sub-account a_1.

The configuration is as follows:

1. Account B configures quick authorization and adds a description to the role. For more information, see *Cross-account shipping*.

2. The main account A logs on to the RAM console and grants AliyunRAMFullAccess permission to the sub-account a_1.
   a. On the User Management page, click Authorization on the right side of the sub-account a_1.
   b. Search for AliyunRAMFullAccess in the authorizable policies, and add it to selected policies. Then click Confirm.

      After successful authorization, a_1 has all RAM permissions.

      To control the permission range of a_1, the main account A can grant a_1 only the permissions required for shipping logs to OSS by modifying `Action` and `Resource` parameters.

      The contents of the `Resource` must be replaced with the ARN of AliyunLogDefaultRole. The example of authorization policy is as follows:

      ```
      {
      " Statement ": [
      {
      " Action ": " ram : PassRole ",
      " Effect ": " Allow ",
      " Resource ": " acs : ram :: 1111111 : role / aliyunlogd
       efaultrole "
      }
      ],
      " Version ": " 1 "
      }
      ```

   c. The sub-account a_1 creates a shipping task. When configuring the task, RAM role column must be filled with the RAM role identifier ARN of the OSS ARN of the OSS bucket owner, that is, the RAM role AliyunLogDefaultRole created by account B.

10.3 Ship data to MaxCompute

10.3.1 Ship data to MaxCompute by using DataWorks

You can not only ship logs to OSS storage, but also ship log data to MaxCompute by using the Data Integration function of DataWorks. Data Integration is a stable, efficient, and elastically scalable data synchronization platform provided by the Alibaba Group to external users. It provides offline batch data access channels for Alibaba Clouds big data computing engines (including MaxCompute, AnalyticDB, and OSPS).

For details about the regions in which this feature is available, see *DataWorks*.

Scenarios

- Data synchronization between data sources (LogHub and MaxCompute) across regions
- Data synchronization between data sources (LogHub and MaxCompute) with different Alibaba Cloud accounts
- Data synchronization between data sources (LogHub and MaxCompute) with the same Alibaba Cloud account
- Data synchronization between data sources (LogHub and MaxCompute) with a public cloud account and an AntCloud account

Prerequisites

1. Log Service, MaxCompute, and DataWorks have been activated.
2. Log Service has successfully collected log data and LogHub has data to ship.
3. An Access Key pair is enabled for the data source account.
4. RAM authorization is configured when shipping across accounts is involved.

   For details, see *Perform authorization for log shipping across accounts* in this document.

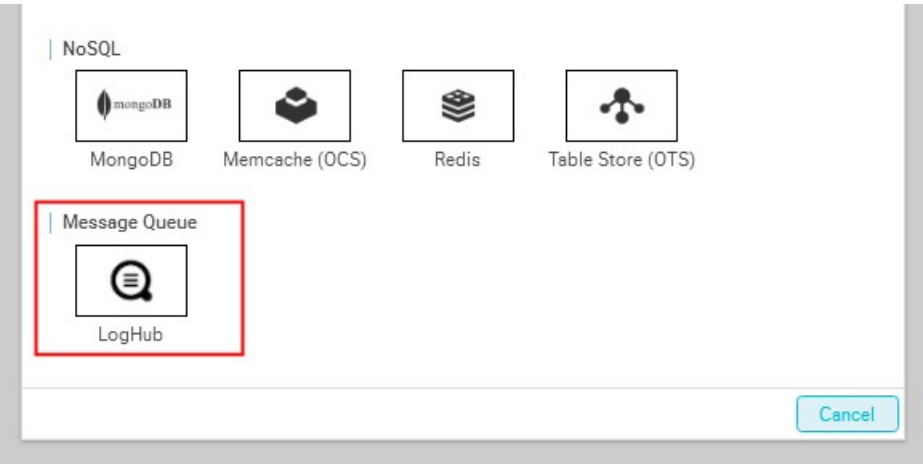
Procedure

Step 1 Create a data source

1. On the DataWorks console, open the Data Integration page. Click the Data Source tab on the navigation bar on the left.
2. On the Data Source page, click New Data Source in the upper right corner. The New Data Source page appears.

3. Click LogHub in the Message Queue list. The New LogHub data source page appears.

   Figure 10-5: Add a data source



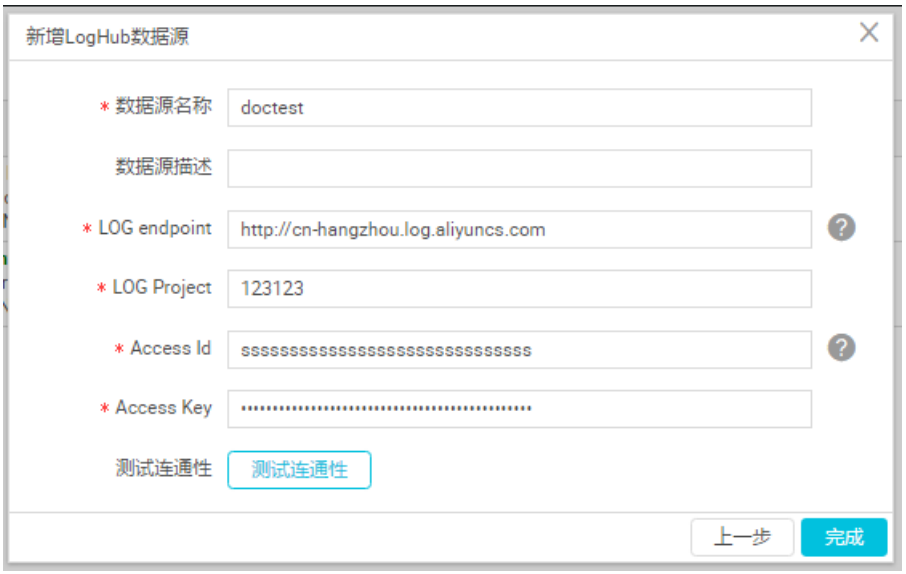
4. Set the configuration items for the data source.

   The following table describes the configuration items:

Configuration items	Description
Data source name	A data source name may consist of letters, digits, and underscores. It must begin with a letter or underscore and cannot contain more than 60 characters.
Data source description	A brief description of the data source, containing up to 80 characters.
LOG Endpoint	Endpoint of Log Service, determined by your region, in the format of `http://yyy.com`. For more information, see *Service endpoint*.
LOG Project	A Log Service project in MaxCompute to which the log data is sent. It must be an existing project. Must be a project that has been created.

Configuration items	Description
Access Id/Access Key	The Access Key of the data source account is equivalent to a logon password. You can enter the Access Key of the primary account or subaccount of the data source. After successful configuration, the current account is granted access to the account logs in the data source and thus can ship logs of the data source account through a synchronization task.

Figure 10-6: Create a LogHub data source



5. Click Test Connectivity Click Finish after Connectivity test is successful appears in the upper right corner of the page.

Step 2 Configure a synchronization task

Click Synchronization Task in the navigation bar on the left and click Step 2 Create a synchronization task to configure the synchronization task.

Select Wizard Mode to configure the task on a visualized page more easily; or select Script Mode to configure your synchronization task with more customization.

Wizard mode

The configuration items of the task synchronization node include Select a Source, Select a Target, Field Mapping, and Channel Control.

1. Select a source.

   Data source: Select the data source configured in Step 1. Set the configuration items according to the following table:

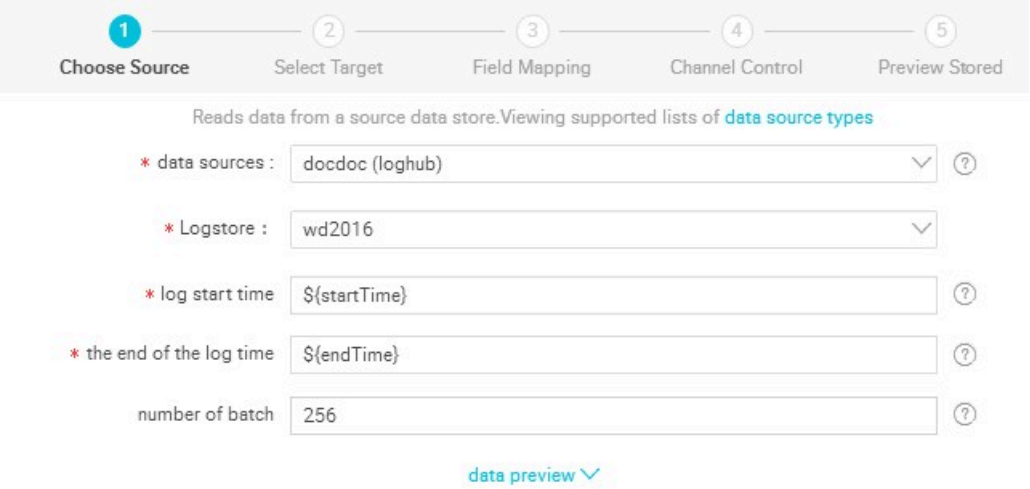
Configuration items	Description
Data source	Select the name of the LogHub data source.
Logstore	Name of the table from which the incremental data is exported. You must enable the Stream feature on the table when creating the table or using UpdateTable API later.
Log start time	Start time of data consumption. The parameter defines the left border of a time range (left closed and right open) in the format of yyyyMMddHHmmss (such as 20180111013000) and can work with the scheduling time parameter in DataWorks.
Log end time	End time of data consumption. The parameter defines the right border of a time range (left closed and right open) in the format of yyyyMMddHHmmss (such as 20180111013010) and can work with the scheduling time parameter in DataWorks.
Batch size	Number of data entries read each time. The default value is 256.

   After the configuration items are set, click the Data Preview drop-down button to show the Data Preview details. Verify that log data has been obtained, and then click Next.

   Note:

Data preview presents several data entries selected from the LogHub. The preview result may differ from the synchronization data that you configure, because the synchronization data is configured with log start time and end time.

Figure 10-7: Select a source



2. Select a target.

   a. Select a MaxCompute data source and target table.

      If you have not created any MaxCompute table, click Generate Target Table in One Click on the right. Choose Create Data Table on the dialog box-up menu.

   b. Fill in Partition information.

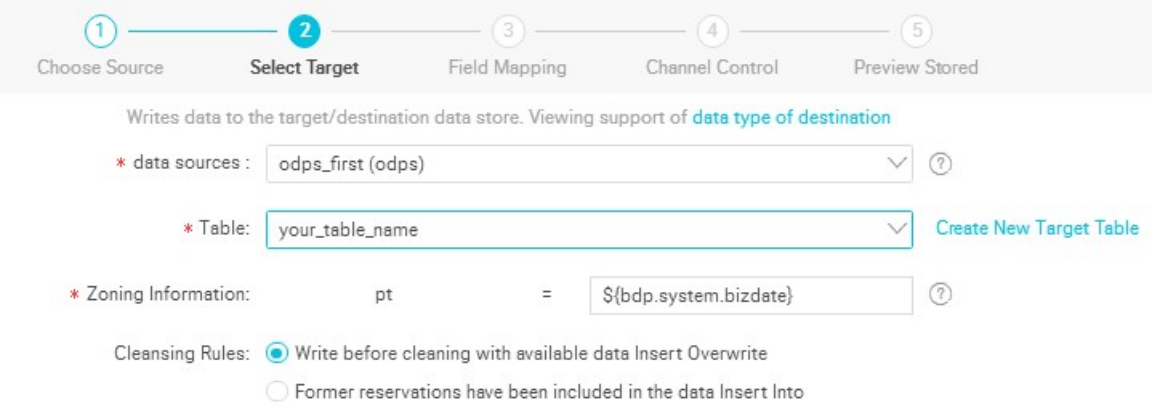
      Partition configuration supports regular expressions. For example, you can set the pt value of the partition “*” to read data in all the pt partitions.

   c. Choose Clearing Rules.

      You can choose to clear exiting data (overwrite mode) or retain existing data (insertion mode) before writing.

   After the configuration, click Next.

   Figure 10-8: Select a target



3. Map fields.

   Select the mapping between fields. You need to configure the field mapping relationship. Source Table Fields on the left correspond one-to-one with Target Table Fields on the right. You can click Same row mapping to select or deselect Same row mapping.

   Note:

   - If you need to manually add log fields as synchronous columns, use the *Script mode* configuration.
   - You can enter constants. Each constant must be enclosed in a pair of single quotes, such as abc and 123.

- You can add scheduling parameters, such as ${bdp.system.bizdate}.
- You can enter functions supported by relational databases, such as now() and count(1).
- If the value you entered cannot be parsed, the type is displayed as Not identified .

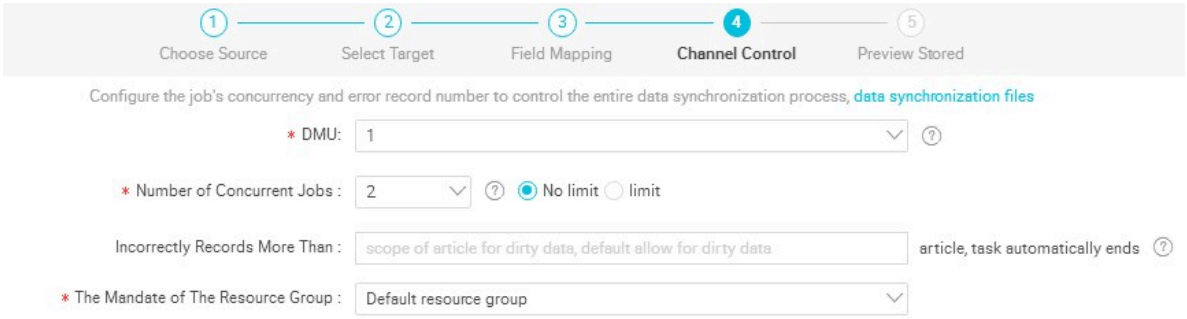
Figure 10-9: Map fields



4. Control the tunnel.

   Configure the maximum job rate and dirty data check rules, as shown in the following figure:

   Figure 10-10: Control the tunnel



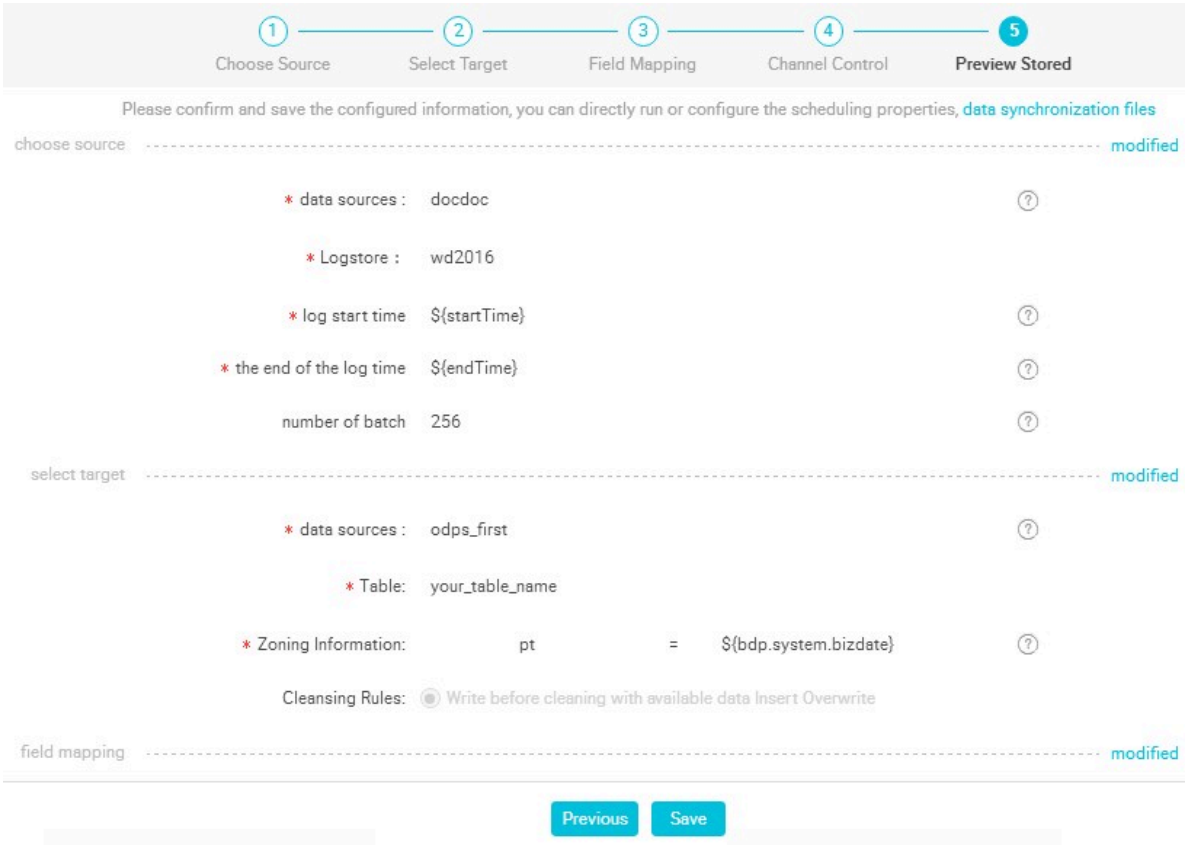
   Configuration item descriptions:

   - DMU: Data migration unit, which measures the resources (including CPU, memory, and network bandwidth) consumed during data integration.
   - Concurrent job count: Maximum number of threads used to concurrently read data from or write data into the data storage media in a data synchronization task.

5. Preview and save the configuration.

   After completing the configuration, you can scroll up or down to view the task configurations. If no error exists, click Save.

   Figure 10-11: Preview and save the configuration



Script mode

To configure the task using a script, see the following script for reference:

```
{
" type ": " job ",
" version ": " 1 . 0 ",
" configurat  ion ": {
" reader ": {
" plugin ": " loghub ",
" parameter ": {
" datasource ": " loghub_lzz ",// Data   source   name .  Use   the
 data   resource   name   you   have   added .
" logstore ": " logstore - ut2 ",// Target   Logstore   name .  A
 Logstore   is   a   log   data   collection ,  storage ,  and   query
   unit   in   the   Log   Service .
" beginDateT  ime ": "${ startTime }",// Start   time   of   data
 consumptio  n .  The   parameter   defines   the   left   border   of
   a   time   range  ( left   closed   and   right   open )
" endDateTim  e ": "${ endTime }",// End   time   of   data
 consumptio  n .  The   parameter   defines   the   right   border
 of   a   time   range  ( left   closed   and   right   open )
```

```
" batchSize ":  256 ,// Number   of   data   entries   read   each
 time .  The   default   value   is   256 .
" splitPk ": "",
" column ": [
" key1 ",
" key2 ",
" key3 "
]
}
},
" writer ": {
" plugin ": " odps ",
" parameter ": {
" datasource ": " odps_first ",// Data   source   name .  Use   the
 data   resource   name   you   have   added .
" table ": " ok ",// Target   table   name
" truncate ":  true ,
" partition ": "",// Shard   informatio  n
" column ": [// Target   column   name
" key1 ",
" key2 ",
" key3 "
]
}
},
" setting ": {
" speed ": {
" mbps ":  8 ,/ Maximum   job   rate
" concurrent ":  7 // Number   of   concurrent   jobs
}
}
}
}
```

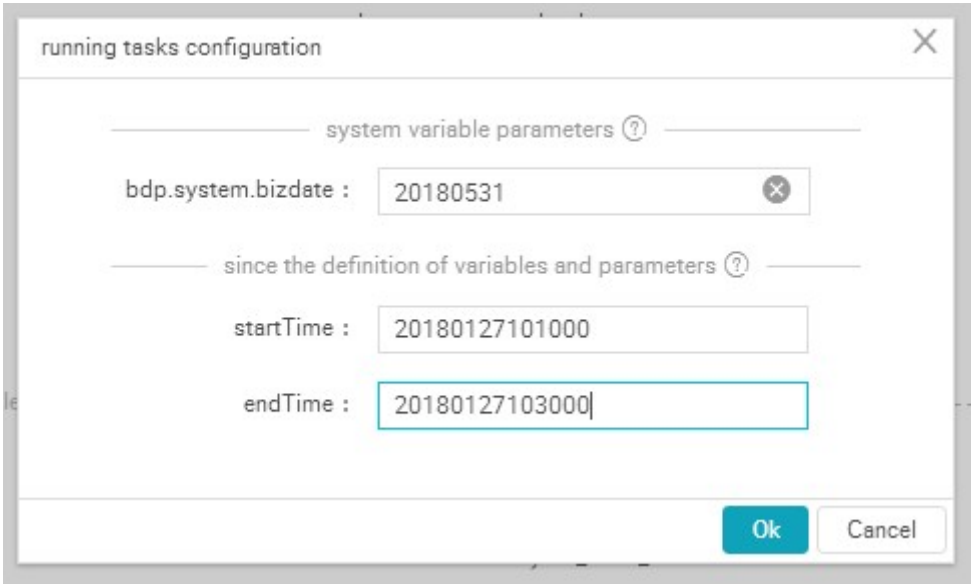
Step 3 Run the task

You can run the task in either of the following ways:

- Directly run the task (one-time running)

  Click Run above the task to run the task directly on the data integration page. Set values for the custom parameters before running the task.

  Figure 10-12: Running task configuration



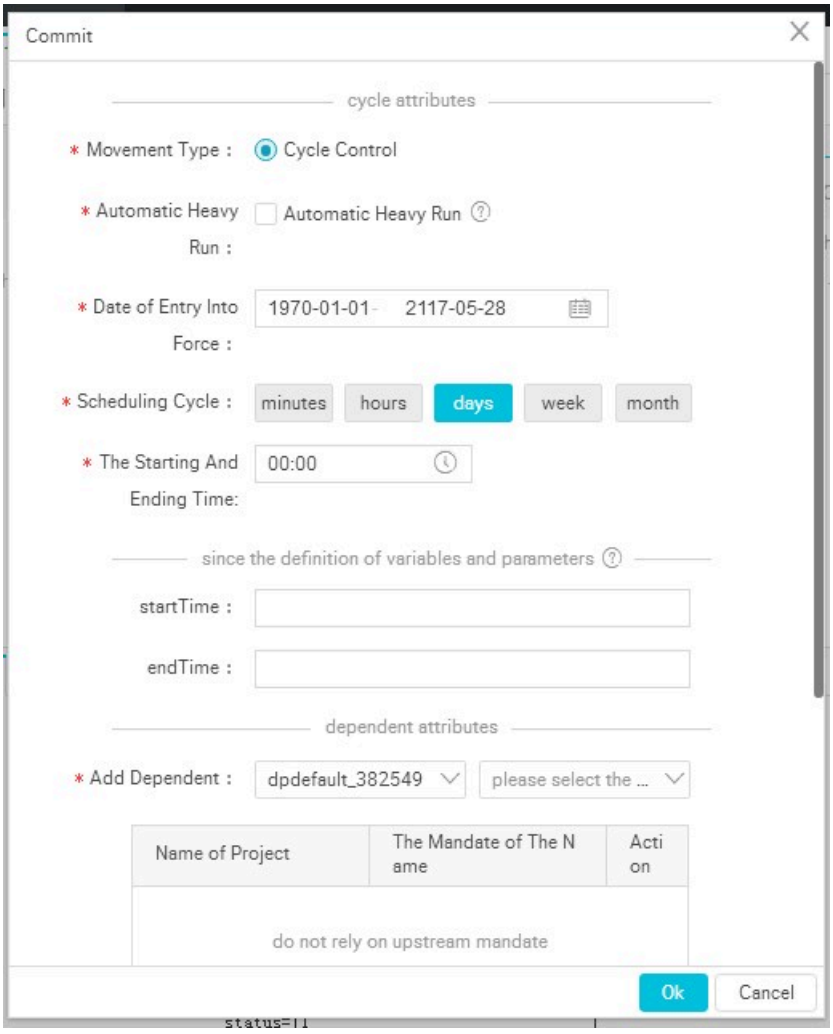
  As shown in the preceding figure, LogHub records between 10:10 and 17:30 are synchronized to MaxCompute.

- Schedule the task

  Click Submit to submit the synchronization task to the scheduling system. The scheduling system automatically and periodically runs the task from the second day according to the configuration attributes.

  Set the schedule interval to 5 minutes, at a schedule period from 00:00 to 23:59, with startTime=$[yyyymmddhh24miss-10/24/60] 10 minutes before the system to endTime=$[yyyymmddhh24miss-5/24/60] 5 minutes before the system.

  Figure 10-13: Scheduling configuration



Perform authorization for log shipping across accounts

To configure a log shipping task across accounts, perform authorization on the RAM.

- Perform authorization for log shipping across accounts

  To ship data between primary accounts, you can enter the Access Key of the primary account of the data source in the step Add LogHub Data Source. Authorization is successful if the connectivity test passes.

  For example, to ship log data under account A to a MaxCompute table of account B through the DataWorks service activated with account B, configure a data integration task with account B and enter the Access Key of the primary account of account A in the step Add LogHub Data Source. After successful configuration, account B has the permission to read all log data under account A.
- Subaccount authorization

  If you do not want to reveal the Access Key of the primary account or need to ship the log data collected by a subaccount, configure explicit authorization for the subaccount.

  - Assign management permissions to the subaccount

    If you need to ship all log data under a primary account through a subaccount, perform the following steps for authorization and Access Key configuration.

    1. Use primary account A to assign Log Service management permissions (`AliyunLogF ullAccess` and `AliyunLogR eadOnlyAcc ess`) to

subaccount A1. For details, see *Grant RAM sub-accounts permissions to access Log Service*.

2. Configure a data integration task with account B, and enter the Access Key of the subaccount of the data source in the step Add LogHub Data Source.

After successful configuration, account B has the permission to read all log data under account A.

- Assign the customization permission to the subaccount

If you need to ship specified log data under a primary account through a subaccount, perform the following steps for authorization and Access Key configuration.

1. Configure a custom authorization policy for subaccount A1 with the primary account A. For details on related authorization operations, see *Authorization - Overview* and *Overview*.
2. Configure a data integration task with account B, and enter the Access Key of the subaccount of the data source in the step Add LogHub Data Source.

When the above steps are successfully completed, account B has the permission to read specified log data under account A.

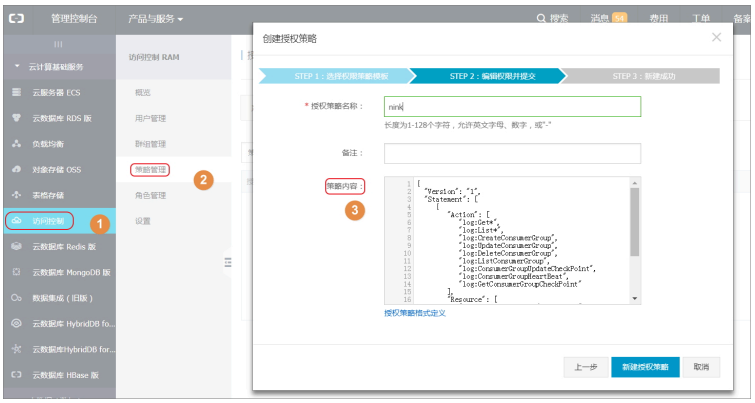
Example of custom authorization policy:

In this way, account B can synchronize only project_name1 and project_name2 data in Log Service through subaccount A1.

```
{
" Version ": " 1 ",
" Statement ": [
{
" Action ": [
" log : Get *",
" log : List *",
" log : CreateCons  umerGroup ",
" log : UpdateCons  umerGroup ",
" log : DeleteCons  umerGroup ",
" log : ListConsum  erGroup ",
" log : ConsumerGr  oupUpdateC  heckPoint ",
" log : ConsumerGr  oupHeartBe  at ",
" log : GetConsume  rGroupChec  kPoint "
],
" Resource ": [
" acs : log :*:*: project / project_na  me1 ",
" acs : log :*:*: project / project_na  me1 /*",
" acs : log :*:*: project / project_na  me2 ",
" acs : log :*:*: project / project_na  me2 /*"
],
" Effect ": " Allow "
```

```
}
]
}
```

Figure 10-14: Custom authorization policy



10.4 Send logs to an SIEM system

10.4.1 Overview

This topic describes some of the requirements and features involved if you use the POST method to send logs to an SIEM system.

Terms

- SIEM: an abbreviation of Security Information and Event Management. It is a security management approach that can be implemented with Splunk and IBM QRadar.
- Splunk HEC: the Splunk HTTP Event Collector. It is an HTTP or HTTPS interface that is used to receive logs.

Recommended specifications for the target server

- Hardware specifications
  - Linux OS (for example, Ubuntu x64)
  - At least one 8-core GPU running a minimum of 2.0 GHz
  - At least 16 GB memory
  - At least one NIC that transmits 1 GB/s
  - At least one disk of 2 GB.

Note:

- Although 16 GB memory is the minimum specification required, we recommend that you upgrade to 32 GB of memory for better performance.
- Although 2 GB is the minimum storage specification required for the disk, we recommend that you upgrade to 10 GB storage for better performance.

· Network

The bandwidth value of the network that connects your server with Alibaba Cloud must be greater than the value of the rate at which data are generated in Alibaba Cloud. Otherwise, the generated data cannot be consumed in real time.

We recommend that you set the bandwidth to 4 MB/s (32 Mbps) for the scenario that meets the following conditions:

- Data are generated at an even rate.
- The peak rate of data generation is twice of the even rate.
- Log data of 1 TB are generated each day.
- The log data to be transmitted are compressed by a 5:1 compression ratio.

· Languages

Both Python and Java can be used to compile the required program.

In the two examples (*Use HTTPS to send logs to an SIEM system* and *Use Syslog to send logs to an SIEM system*), Python is used.

For information about how the example in which Java is used, see *Use a consumer group to consume logs*.

Consumer library

The consumer library provides the advanced mode to consume logs in Log Service. Specifically, the consumer library uses consumer groups to abstract and manage the data consumption. In contrast, SDK is used to directly read data. With the consumer library, you can focus on how the data is consumed and sent. You can ignore to the details about how Log Service works, or the load balancing and failover event between consumers.

One Logstore in Log Service contains multiple shards. The consumer Library assign shards to the consumers of one consumer group by following these rules:

· Each shard can only be assigned to one consumer.
· Multiple shards can be assigned to the same consumer.

 Note:

- These rules take effect each time when shards are assigned to consumers in a consumer group.
- When a new consumer is added in a consumer group, the shards assigned to the consumers before will be reassigned to all the consumers in the consumer group. This is a method of load balancing in the consumer group.
- The process in which shards are assigned to consumers is transparent.

In addition, the consumer library can also save check-points. This feature helps data consumption continue from the point where a failure occurred, avoiding re-consumption of the same data.

Spark Streaming, Storm, and Flink Connector all works on the base of the consumer library.

Methods to send logs to an SIEM system

We recommend that you use consumer groups in Log Service to configure the required program to consume log data in real time, and then use send log data to an SIEM system through `HTTPS` or `Syslog` .

For more information, see *Use HTTPS to send logs to an SIEM system* and *Use Syslog to send logs to an SIEM system*.

10.4.2 Use HTTPS to send logs to an SIEM system

This topic uses an example to describe how to send Alibaba Cloud logs to an SIEM system (Splunk in this example) through the HTTP Event Collector (HEC).

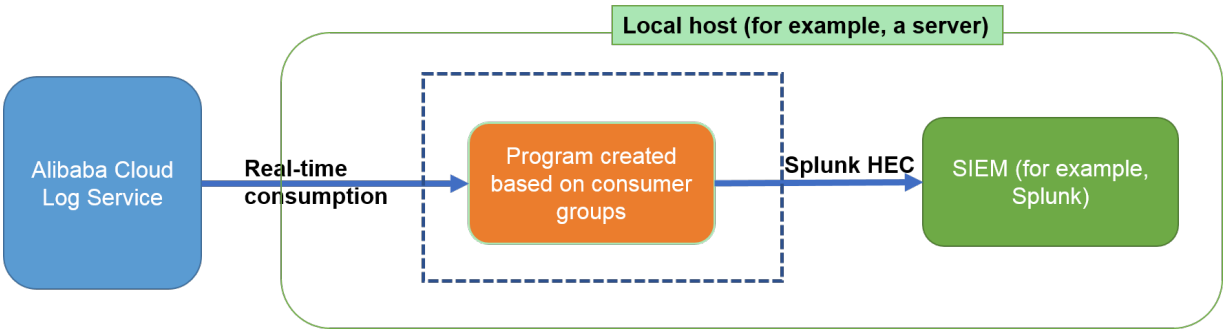
Prerequisites

- Splunk Enteprise Security is deployed on-premises to support the SIEM system, rather than in the cloud.
- This SIEM system cannot be accessed through the Internet because there are no corresponding interfaces configured for Internet access.
- PyPy3 is used to run the program, instead of CPython (the standard interpreter of Python).
- The Python SDK of Log Service is installed by using the `pypy3 -m pip install aliyun-log-python-sdk -U` command. For more information about Python SDK, see *User Guide*.

> **Note:**
> This topic uses a few of code examples. If you want to view all the latest code, you can obtain them at *Github*, or *Github (applicable to the Logstore that contains multiple data sources)*.

Workflow

We recommend that you use the consumer groups in Log Service to configure the required program to consume log data in real time. Then, the program can send logs to the SIEM system (Splunk) by using the Splunk API (that is, HEC).



Limits

For each Logstore in Log Service, up to ten consumer groups can be set. If the system displays the error message `ConsumerGr oupQuotaEx ceed`, we recommend that you log on to the Log Service console to delete the consumer groups that you do not need.

Control logic of the program

The following code shows the control logic of the program:

```
def  main ():
    option ,  settings  =  get_monito  r_option ()

    logger . info ("***  start  to  consume  data ...")
    worker  =  ConsumerWo  rker ( SyncData ,  option ,  args =(
settings ,) )
    worker . start ( join = True )

if  __name__  == ' __main__ ':
```

```
    main ()
```

Configurations and a code example of the program

- Configurations
  - The log file of the program: This file is used to debug the program or diagnose any exceptions that may occur.
  - The basic configurations of the program: These include the configurations used to connect to Log Service and those of the consumer group.
  - The advanced configurations of the consumer group: These configurations are used to debug performance. We recommend that you do not modify them.
  - The parameters and options related to the SIEM (Splunk).
- A code example of the program

  You must follow the annotations of the code to customize it to meet your needs.

```
# encoding :  utf8
 import   os
 import   logging
 from   logging . handlers   import   RotatingFi  leHandler

 root  =  logging . getLogger ()
 handler  =  RotatingFi  leHandler ("{ 0 } _ { 1 }. log ". format (
 os . path . basename ( __file__ ),  current_pr  ocess (). pid ),
 maxBytes = 100 * 1024 * 1024 ,  backupCoun  t = 5 )
 handler . setFormatt  er ( logging . Formatter ( fmt ='[%( asctime
 ) s ] - [%( threadName ) s ] - {%( module ) s :%( funcName ) s :%(
 lineno ) d } %( levelname ) s  - %( message ) s ',  datefmt ='% Y -
% m -% d  % H :% M :% S '))
 root . setLevel ( logging . INFO )
 root . addHandler ( handler )
 root . addHandler ( logging . StreamHand  ler ())

 logger  =  logging . getLogger ( __name__ )

 def   get_option ():
    ##########################
    #  Basic   options
    ##########################

    #  Load   parameters   and   options   of   SLS   from
 environmen  t   variables .
     endpoint  =  os . environ . get (' SLS_ENDPOI  NT ', '')
     accessKeyI  d  =  os . environ . get (' SLS_AK_ID ', '')
     accessKey  =  os . environ . get (' SLS_AK_KEY ', '')
     project  =  os . environ . get (' SLS_PROJEC  T ', '')
     logstore  =  os . environ . get (' SLS_LOGSTO  RE ', '')
     consumer_g  roup  =  os . environ . get (' SLS_CG ', '')

    #  This   parameter   specifies   the   starting   time   when
 data   consumptio  n   is   initialize  d .
    #  You   can   use   begin   or   end   when   you   set
 this   parameter .  The   time   can   be   set   to   use   a
 specific   ISO   format .
     cursor_sta  rt_time  = " 2018 - 12 - 26   0 : 0 : 0 "
```

```
    ##########################
    #  Advanced   options
    ##########################

    #  Do   not   configure   the   consumer   name ,  especially
when   you   need   to   run   this   program   in   parallel .
     consumer_n  ame  = "{ 0 }-{ 1 }". format ( consumer_g  roup ,
current_pr  ocess (). pid )

    #  Set   the   heartbeat   interval .  If   the   server   does
  not   receive   the   heartbeat   report   from   the   specified
  shard   within   two   times   of   the   specified   interval ,
the   server   determines   the   consumer   is   off   line   and
  will   reassign   its   task   to   another   consumer .
    #  We   recommend   that   you   set   a   greater   interval
when   the   network   provides   low   performanc  e .
     heartbeat_  interval  =  20

    #  The   maximum   interval   between   two   times   of   data
  consumptio  n .  If   data   are   generated   quickly ,  do
not   adjust   this   setting .
     data_fetch  _interval  =  1

    #  Create   a   consumer   group   that   contains   a
consumer .
     option  =  LogHubConf  ig ( endpoint ,  accessKeyI  d ,
accessKey ,  project ,  logstore ,  consumer_g  roup ,  consumer_n
ame ,
                          cursor_pos  ition = CursorPosi  tion .
SPECIAL_TI  MER_CURSOR ,
                          cursor_sta  rt_time = cursor_sta  rt_time
,
                          heartbeat_  interval = heartbeat_
interval ,
                          data_fetch  _interval = data_fetch
_interval )

    #  Splunk   options
     settings  = {
                " host ": " 10 . 1 . 2 . 3 ",
                " port ":  80 ,
                " token ": " a023nsdu12  3123123 ",
                ' https ':  False ,              #  optional .
bool
                ' timeout ':  120 ,             #  optional .  int
                ' ssl_verify ':  True ,         #  optional .  bool
                " sourcetype ": "",            #  optional .
                " index ": "",                #  optional .
                " source ": "",               #  optional .
           }

     return   option ,  settings
```

An example of the code used to consume and send data

The following code shows the settings about for to obtain data from Log Service and how to send the data to the Splunk:

```
from   aliyun . log . consumer   import  *
from   aliyun . log . pulllog_re  sponse   import   PullLogRes
ponse
```

```
from   multiproce  ssing   import   current_pr  ocess
import   time
import   json
import   socket
import   requests

class   SyncData ( ConsumerPr  ocessorBas  e ):
   """
    This   consumer   obtains   data   from   Log   Service   and
send   the   data   to   Splunk .
   """
    def   __init__ ( self ,  splunk_set  ting ):
     """ Initiate   the   Splunk   server   and   its   network
connectivi  ty ."""
        super ( SyncData ,  self ). __init__ ()

        assert   splunk_set  ting ,  ValueError (" You   need   to
configure   settings   of   remote   target ")
        assert   isinstance ( splunk_set  ting ,  dict ),  ValueError
(" The   settings   should   be   dict   to   include   necessary
address   and   confidenti  als .")

        self . option  =  splunk_set  ting
        self . timeout  =  self . option . get (" timeout ",  120 )

       #  Test   Splunk   connectivi  ty .
        s  =  socket . socket ()
        s . settimeout ( self . timeout )
        s . connect (( self . option [" host "],  self . option ['
port ']))

        self . r  =  requests . session ()
        self . r . max_redire  cts  =  1
        self . r . verify  =  self . option . get (" ssl_verify ",
True )
        self . r . headers [' Authorizat  ion '] = " Splunk  {}".
format ( self . option [' token '])
        self . url  = "{ 0 }://{ 1 }:{ 2 }/ services / collector /
event ". format (" http "  if   not   self . option . get (' https
')  else  " https ",  self . option [' host '],  self . option ['
port '])

        self . default_fi  elds  = {}
        if   self . option . get (" sourcetype "):
            self . default_fi  elds [' sourcetype '] =  self . option
. get (" sourcetype ")
        if   self . option . get (" source "):
            self . default_fi  elds [' source '] =  self . option .
get (" source ")
        if   self . option . get (" index "):
            self . default_fi  elds [' index '] =  self . option .
get (" index ")

    def   process ( self ,  log_groups ,  check_poin  t_tracker ):
        logs  =  PullLogRes  ponse . loggroups_  to_flatter  n_list (
log_groups ,  time_as_st  r = True ,  decode_byt  es = True )
        logger . info (" Get   data   from   shard  { 0 },  log
count : { 1 }". format ( self . shard_id ,  len ( logs )))
        for   log   in   logs :
           #  Send   data   to   Splunk .
            event  = {}
            event . update ( self . default_fi  elds )
            event [' time '] =  log [ u ' __time__ ']
            del   log [' __time__ ']
```

```
            json_topic  = {" actiontrai  l_audit_ev  ent ": [" event
"] }
            topic  =  log . get (" __topic__ ", "")
            if  topic  in  json_topic :
                try :
                    for  field  in  json_topic [ topic ]:
                        log [ field ] =  json . loads ( log [ field
])
                except  Exception  as  ex :
                    pass
            event [' event '] =  json . dumps ( log )

            data  =  json . dumps ( event ,  sort_keys = True )

                try :
                    req  =  self . r . post ( self . url ,  data =
data ,  timeout = self . timeout )
                    req . raise_for_  status ()
                except  Exception  as  err :
                    logger . debug (" Failed  to  connect  to
remote  Splunk  server  ({ 0 }).  Exception : { 1 }",  self . url
,  err )

                   #  TODO :  Add  the  code  to  process
errors .  For  example ,  you  can  add  the  code  to  retry
  or  send  a  notificati  on  in  response  to  an  error .
            raise  err

        logger . info (" Complete  send  data  to  remote ")

        self . save_check  point ( check_poin  t_tracker )
```

A example of the code used to start the program

Assume that the program is named `sync_data . py` . The code used to start the program is as follows:

```
export  SLS_ENDPOI  NT =< Endpoint  of  your  region >
export  SLS_AK_ID =< YOUR  AK  ID >
export  SLS_AK_KEY =< YOUR  AK  KEY >
export  SLS_PROJEC  T =< SLS  Project  Name >
export  SLS_LOGSTO  RE =< SLS  Logstore  Name >
export  SLS_CG =< SLS  consumer  group  name .  You  can  set
it  to  syc_data .>

pypy3  sync_data . py
```

An example code of the Logstore that contains multiple sources

For the Logstore that contains multiple sources, you must set an executor to avoid process broken down. For more information about related examples, see *Send logs from a multi-source Logstore to Splunk*.

 **Note:**

You must pay attention to the changes to the primary functions.

```
exeuctor , options , settings = get_option ()

    logger . info ("*** start to consume data ...")
    workers = []

    for option in options :
        worker = ConsumerWo rker ( SyncData , option , args =(
settings ,) )
        workers . append ( worker )
        worker . start ()

    try :
        for i , worker in enumerate ( workers ):
            while worker . is_alive ():
                worker . join ( timeout = 60 )
            logger . info (" worker project : { 0 } logstore : { 1
} exit unexpected , try to shutdown it ". format (
                options [ i ]. project , options [ i ]. logstore ))
            worker . shutdown ()
    except KeyboardIn terrupt :
        logger . info ("*** try to exit **** ")
        for worker in workers :
            worker . shutdown ()

       # wait for all workers to shutdown before
shutting down executor
        for worker in workers :
            while worker . is_alive ():
                worker . join ( timeout = 60 )

    exeuctor . shutdown ()
```

View and monitor data consumption

- You can log on to the Log Service console to view the status of a consumer group. For more information, see *View consumer group status*.
- You can log on to the CloudMonitor to view delays associated with consumer groups and configure corresponding alarms. For more information, see *Consumer group delay*.

Run multiple consumers

For a program that is based on consumer groups, you can start the program multiple times to run multiple consumers.

```
nohup pypy3 sync_data . py &
nohup pypy3 sync_data . py &
nohup pypy3 sync_data . py &
...
```

 Note:

The data of one shard can be consumed by only one consumer. If a Logstore contains ten shards, then up to ten consumers can consume the data of all of the shards.

Throughput test scenario

Assume that you have an instance with a single core CPU. The data of raw logs is consumed at a rate of 5 MB/s only if the following conditions are met:

- The bandwidth and rate to receive data is not limited.
- PyPy3 is used to run the following example codes.
- One consumer uses about 10% resources of a single-core CPU.

Therefore, ten consumers that use the same single-core CPU can consume the data of raw logs at the rate of 50 MB/s. Then, up to 4 TB data of raw logs can be consumed on such a CPU.

High availability scenario

A consumer group saves check-points on the server end. When one consumer stops, another consumer automatically takes over the data consumption task and continues the task from the check-point where the preceding consumer stopped.

Therefore, to achieve high availability, start multiple consumers that are located in different servers. That way, if one of these servers stops or is damaged, a consumer located in one of the other servers can take over the consumption task and continue the task from the check-point where the preceding consumer stopped.

To ensure sufficient consumers are available, set the number of the consumers that you start in different servers to be greater than the number of the shards.

HTTPS

To use HTTPS to encrypt the data transmitted between your configured program and Log Service, you must set the prefix of the service endpoint to `https://`. For example, `https://cn-beijing.log.aliyuncs.com`.

The server certificate *.aliyuncs.com issued by GlobalSign. Generally, most Linux or Windows server automatically trusts this certificate. If a server does not trust this certificate, you can download the valid certificate at *Certificates*.

10.4.3 Use Syslog to send logs to an SIEM system

This topic describes the required configurations and related information for how to use Syslog to send logs to your Security Information and Event Management (SIEM)

software. Syslog is a log path through which most physical devices (such as switches, routers, and servers) send logs. In addition, almost SIEM software (including IBM Qradar and HP Arcsight) can receive logs that are sent with Syslog.

Background information

- Syslog: This protocol defines the standard format of logs according to the standard protocols *RFC5424* and *RFC3164*.

  The RFC3164 protocol was released in 2001 and the RFC5424 protocol released in 2009 is an upgraded version. We recommend that you use the RFC5424 protocol. It has fixed many of the issues of the previous version and is compatible with the RFC3164 protocol.
- Syslog over TCP/TLS: If the TCP and TLS protocols are supported by your SIEM software, this method can be used to transfer data to your software. We recommend that you use this method.

  Both the TCP and UDP protocols support Syslog. These two protocols can guarantee the stable data transmission. The *RFC5425* protocol also defines the TLS protocol.
- Syslog facility: This is a program component defined by the earlier versions of Unix. For more information, see *Facility*. You can select `user-level messages` as the default facility.
- Syslog severity: This refers to *Message priorities*. You can set the priority for the specified logs as needed. By default, `Informational: informational messages` is used.

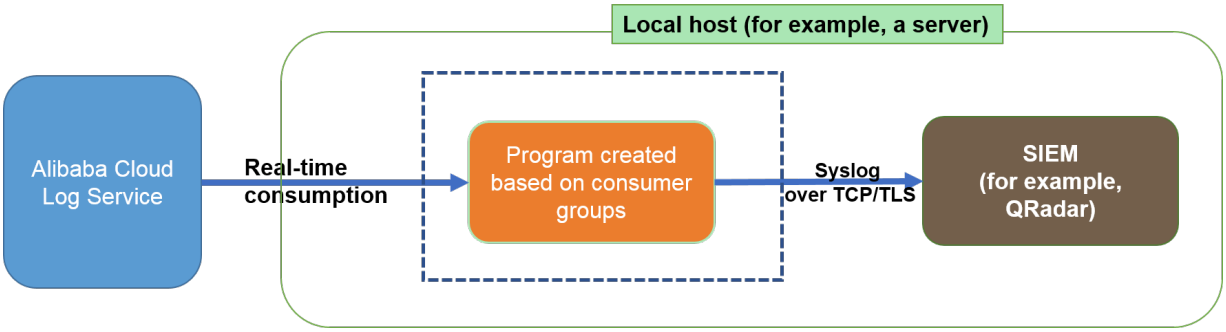
> **Note:**
> This topic uses a few of code examples. If you want to view all the latest code, you can obtain them at *Github*.

Prerequisites

- PyPy3 is used to run the program, instead of CPython (the standard interpreter of Python).
- The Python SDK of Log Service is installed by using the `pypy3 -m pip install aliyun-log-python-sdk -U` command. For more information about Python SDK, see *User Guide*.

Workflow

We recommend that you configure the required program that is based on the consumer groups in Log Service. Then, the program can send logs to the SIEM system by using Syslog.



Limits

Up to ten consumer groups can be set for each Logstore in Log Service. If the system displays the error message `ConsumerGr oupQuotaEx ceed`, we recommend that you log on to the Log Service console to delete the consumer groups that you do not need.

Control logic of the program

The following code shows the control logic of the program:

```
def   main ():
    option ,  settings  =  get_monito  r_option ()

    logger . info ("***  start   to   consume   data ...")
    worker  =  ConsumerWo  rker ( SyncData ,  option ,  args =(
settings ,) )
    worker . start ( join = True )

if   __name__  == ' __main__ ':
```

```
    main ()
```

Configurations and a code example of the program

- Configurations
  - The log file of the program: This file is used to debug the program or diagnose any exceptions that may occur.
  - The basic configurations of the program: These include the configurations used to connect to Log Service and those of the consumer group.
  - The advanced configurations of the consumer group: These configurations are used to debug performance. We recommend that you do not modify them.
  - The parameters and options related to the Syslog server.

    Note:

    If SIEM supports the Syslog path that is based on TCP or TLS, set the `proto` parameter of the program to TLS and set a valid SSL certificate.
- A code example of the program

  You must follow the annotations of the code to customize it to meet your needs.

```
# encoding :  utf8
 import   os
 import   logging
 from   logging . handlers   import   RotatingFi  leHandler

 root  =  logging . getLogger ()
 handler  =  RotatingFi  leHandler ("{ 0 } _ { 1 }. log ". format (
 os . path . basename ( __file__ ),  current_pr  ocess (). pid ),
 maxBytes = 100 * 1024 * 1024 ,  backupCoun  t = 5 )
 handler . setFormatt  er ( logging . Formatter ( fmt ='[%( asctime
 ) s ] - [%( threadName ) s ] - {%( module ) s :%( funcName ) s :%(
 lineno ) d } %( levelname ) s  - %( message ) s ',  datefmt ='% Y -
% m -% d  % H :% M :% S '))
 root . setLevel ( logging . INFO )
 root . addHandler ( handler )
 root . addHandler ( logging . StreamHand  ler ())

 logger  =  logging . getLogger ( __name__ )

 def   get_option ():
    ##########################
    #  Basic   options
    ##########################

    #  Load   parameters   and   options   of   SLS   from
 environmen  t   variables .
     endpoint  =  os . environ . get (' SLS_ENDPOI  NT ', '')
     accessKeyI  d  =  os . environ . get (' SLS_AK_ID ', '')
     accessKey  =  os . environ . get (' SLS_AK_KEY ', '')
     project  =  os . environ . get (' SLS_PROJEC  T ', '')
     logstore  =  os . environ . get (' SLS_LOGSTO  RE ', '')
```

```
    consumer_g  roup  =  os . environ . get (' SLS_CG ', '')

   #  This   parameter   specifies   the   starting   time   when
data   consumptio  n   is   initialize  d .
   #  You   can   use   begin   or   end   when   you   set
this   parameter .  The   time   can   be   set   to   use   a
specific   ISO   format .
    cursor_sta  rt_time  = " 2018 - 12 - 26   0 : 0 : 0 "

   ##########################
   #  Advanced   options
   ##########################

   #  Do   not   configure   the   consumer   name ,  especially
when   you   need   to   run   this   program   in   parallel .
    consumer_n  ame  = "{ 0 }-{ 1 }". format ( consumer_g  roup ,
current_pr  ocess (). pid )

   #  Set   the   heartbeat   interval .  If   the   server   does
  not   receive   the   heartbeat   report   from   the   specified
  shard   within   two   times   of   the   specified   interval ,
the   server   determines   the   consumer   is   off   line   and
  will   reassign   its   task   to   another   consumer .
   #  We   recommend   that   you   set   a   greater   interval
when   the   network   provides   low   performanc  e .
    heartbeat_  interval  =  20

   #  The   maximum   interval   between   two   times   of   data
  consumptio  n .  If   data   are   generated   quickly ,  do
not   adjust   this   setting .
    data_fetch  _interval  =  1

   #  Create   a   consumer   group   that   contains   a
consumer .
    option  =  LogHubConf  ig ( endpoint ,  accessKeyI  d ,
accessKey ,  project ,  logstore ,  consumer_g  roup ,  consumer_n
ame ,
                          cursor_pos  ition = CursorPosi  tion .
SPECIAL_TI  MER_CURSOR ,
                          cursor_sta  rt_time = cursor_sta  rt_time
,
                          heartbeat_  interval = heartbeat_
interval ,
                          data_fetch  _interval = data_fetch
_interval )

   #  syslog   options
    settings  = {
                " host ": " 1 . 2 . 3 . 4 ", #  required .
                " port ":  514 ,       #  required .
                " protocol ": " tcp ", #  required .  Valid :  tcp
|  udp  |  tls  ( only   applicable   to   Python3 )
                " sep ": "||",       #  required .  Set   the
separator   to   separate   key - value   pairs .
                " cert_path ":  None , #  optional .  Set   the
directory   where   the   TLS   certificat  e   is   located .

                " timeout ":  120 ,    #  optional .  Set   the
timeout .  The   default   timeout   is   120   seconds .
                " facility ":  syslogclie  nt . FAC_USER ,  #
optional .  You   can   refer   to   the   values   of   other
syslogclie  nt . FAC_ *.
```

```
                " severity ":  syslogclie  nt . SEV_INFO ,  #
optional .  You   can   refer   to   the   values   of   other
syslogclie  nt . SEV_ *.
                " hostname ":  None ,  #  optional .  Set   the
host   name .  By   default ,  the   name   of   the   host   that
  you   use   is   used .
                " tag ":  None         #  optional .  Set   the
tag .  By   default ,  a   hyphen  (-)  is   used .
            }

    return   option ,  settings
```

An example of the code used to consume and send data

The following code shows the settings about for to obtain data from Log Service and how to send the data to the Syslog server:



Note:

You can customize the format of the code according to your needs.

```
from   syslogclie  nt   import   SyslogClie  ntRFC5424   as
SyslogClie  nt

class   SyncData ( ConsumerPr  ocessorBas  e ):
   """
    This   consumer   obtains   data   from   Log   Service   and
send   the   data   to   the   Syslog   server .
   """
    def   __init__ ( self ,  splunk_set  ting ):
     """ Initiate   the   Syslog   server   and   its   network
connectivi  ty ."""
        super ( SyncData ,  self ). __init__ ()   #  remember   to
call   base ' s   init

        assert   target_set  ting ,  ValueError (" You   need   to
configure   settings   of   remote   target ")
        assert   isinstance ( target_set  ting ,  dict ),  ValueError
(" The   settings   should   be   dict   to   include   necessary
address   and   confidenti  als .")

        self . option  =  target_set  ting
        self . protocol  =  self . option [' protocol ']
        self . timeout  =  int ( self . option . get (' timeout ',
120 ))
        self . sep  =  self . option . get (' sep ', "||")
        self . host  =  self . option [" host "]
        self . port  =  int ( self . option . get (' port ',  514 ))
        self . cert_path = self . option . get (' cert_path ',  None
)

       #  try   connection
        with   SyslogClie  nt ( self . host ,  self . port ,  proto
= self . protocol ,  timeout = self . timeout ,  cert_path = self .
cert_path )  as   client :
            pass

    def   process ( self ,  log_groups ,  check_poin  t_tracker ):
        logs  =  PullLogRes  ponse . loggroups_  to_flatter  n_list (
log_groups ,  time_as_st  r = True ,  decode_byt  es = True )
```

```
        logger . info (" Get   data   from   shard  { 0 },  log
count : { 1 }". format ( self . shard_id ,  len ( logs )))
        try :
            with   SyslogClie  nt ( self . host ,  self . port ,
proto = self . protocol ,  timeout = self . timeout ,  cert_path =
self . cert_path )  as   client :
                for   log   in   logs :
                    #  Put   your   sync   code   here   to   send
to   remote .
                    #  the   format   of   log   is   just   a
dict   with   example   as   below  ( Note ,  all   strings   are
unicode ):
                    #     Python2 : {" __time__ ": " 12312312 ", "
__topic__ ": " topic ",  u " field1 ":  u " value1 ",  u " field2 ":
u " value2 "}
                    #     Python3 : {" __time__ ": " 12312312 ", "
__topic__ ": " topic ", " field1 ": " value1 ", " field2 ": " value2
"}
                    #  suppose   we   only   care   about   audit
log
                     timestamp  =  datetime . fromtimest  amp ( int (
log [ u ' __time__ ']))
                     del   log [' __time__ ']

                     io  =  six . StringIO ()
                     first  =  True
          #  TODO :   Modify   the   content   to   be   formatted
as   needed .  The   data   is   transmitte  d   by   using   key -
value   pairs   which   are   separated   by  ||.
                     for   k ,  v   in   six . iteritems ( log ):
                         io . write ("{ 0 }{ 1 }={ 2 }". format ( self
. sep ,  k ,  v ))

                     data  =  io . getvalue ()

          #  TODO :   Modify   the   facility   or   severity   as
needed .
                     client . log ( data ,  facility = self . option
. get (" facility ",  None ),  severity = self . option . get ("
severity ",  None ),  timestamp = timestamp ,  program = self .
option . get (" tag ",  None ),  hostname = self . option . get ("
hostname ",  None ))

        except   Exception   as   err :
            logger . debug (" Failed   to   connect   to   remote
syslog   server  ({ 0 }).  Exception : { 1 }". format ( self . option
,  err ))

            #  TODO :  Add   the   code   to   process   errors .  For
  example ,  you   can   add   the   code   to   retry   or   send
a   notificati  on   in   response   to   an   error .
            raise   err

        logger . info (" Complete   send   data   to   remote ")
```

```
            self . save_check  point ( check_poin  t_tracker )
```

A example of the code used to start the program

Assume that the program is named `sync_data . py` . The code used to start the program is as follows:

```
export   SLS_ENDPOI  NT =< Endpoint   of   your   region >
export   SLS_AK_ID =< YOUR   AK   ID >
export   SLS_AK_KEY =< YOUR   AK   KEY >
export   SLS_PROJEC  T =< SLS   Project   Name >
export   SLS_LOGSTO  RE =< SLS   Logstore   Name >
export   SLS_CG =< SLS   consumer   group   name .  You   can   set
it   to   syc_data .>

pypy3   sync_data . py
```

View and monitor data consumption

- You can log on to the Log Service console to view the status of a consumer group. For more information, see *View consumer group status*.
- You can log on to the CloudMonitor to view delays associated with consumer groups and configure corresponding alarms. For more information, see *Consumer group delay*.

Run multiple consumers

For a program that is based on consumer groups, you can start the program multiple times to run multiple consumers.

```
 nohup   pypy3   sync_data . py  &
 nohup   pypy3   sync_data . py  &
 nohup   pypy3   sync_data . py  &
...
```



Note:

The data of one shard can be consumed by only one consumer. If a Logstore contains ten shards, then up to ten consumers can consume the data of all of the shards.

Throughput test scenario

Assume that you have an instance with a single core CPU. The data of raw logs is consumed at a rate of 5 MB/s only if the following conditions are met:

- No limit of the bandwidth and rate to receive data exists.
- PyPy3 is used to run the following example codes.
- One consumer uses about 10% resources of a single-core CPU.

Therefore, ten consumers that use the same single-core CPU can consume the data of raw logs at the rate of 50 MB/s. Then, up to 4 TB data of raw logs can be consumed on such a CPU.

High availability scenario

A consumer group saves check-points on the server end. When one consumer stops, another consumer automatically takes over the data consumption task and continues the task from the check-point where the preceding consumer stopped.

Therefore, to obtain high availability, you can start multiple consumers that are located in different servers. That way, if one of these servers stops or is damaged, a consumer located in one of the other servers can take over the consumption task and continue the task from the check-point where the preceding consumer stopped.

To ensure sufficient consumers are available, set the number of the consumers that you start in different servers to be greater than the number of the shards.

10.5 Manage LogShipper tasks

LogShipper is a function in Log Service that allows you to maximize your data value . You can ship the collected logs to Object Storage Service (OSS) in the console to store data for a long term or consume data together with other systems such as E-MapReduce. After the LogShipper function is enabled, Log Service backend regularly ships the logs written to the Logstore to the corresponding cloud products. The Log Service console provides the OSS Shipper page for you to query the data shipping status within a specified time range, which allows you to know the shipping progress and handle online issues in time.

On the Logstore List page, click OSS in the left-side navigation pane. The OSS Shipper page appears. You can manage your LogShipper tasks in the following ways.

Enable/disable LogShipper tasks

1. Select the target Logstore on the OSS Shipper page.
2. Click Enable or Disable to enable or disable the tasks.

   You must reconfigure the shipping rule after you enable the tasks again.

You must reconfigure the shipping rule after you enable the tasks again.

Configure a shipping rule

After enabling the LogShipper tasks, click Setting to modify the shipping rule.

View details of a LogShipper task

You can filter the LogShipper tasks to be viewed based on the Logstore, time range, and task shipping status. Then, you can view the details of a specific LogShipper task on this page, such as the status, start time, end time, time when logs are received, and type.

A LogShipper task has three kinds of status.

Status	Description	Operation
Success	Logs are successfully shipped.	No need to pay attention.
Running	Logs are being shipped.	Check whether or not logs are successfully shipped later.
Failed	Logs failed to be shipped. The LogShipper task has encountered an error because of external reasons and cannot be retried.	For more information, see Manage LogShipper tasks in Ship logs to OSS.

Delete shipping configuration

Procedure

1. On the Logstore list page, click Delete rule.
2. Click Confirm in the dialog box.

   Once deleted, you will no longer be able to create an offline archive configuration with the same name. Please choose carefully.

11 Log Service Monitor

11.1 Monitor Log Service

You can view the monitoring data of Log Service in the CloudMonitor console or Log Service console.

- In the CloudMonitor console, you can view:
  - Log reading/writing in Logstores
  - Logs collected by agents (Logtail)
- In the Log Service console, you can view:
  - Current point of real-time subscription consumption (Spark Streaming, Storm, and consumer library)
  - Log shipping status

This document describes how to view monitoring data in the Alibaba Cloud CloudMonitor console. For how to view monitoring data in the Log Service console, see *View consumer group status*, *Manage LogShipper tasks* and *Configure an alarm*.

Procedure



Note:
You must authorize the sub-accounts before using them to configure the cloud monitoring.

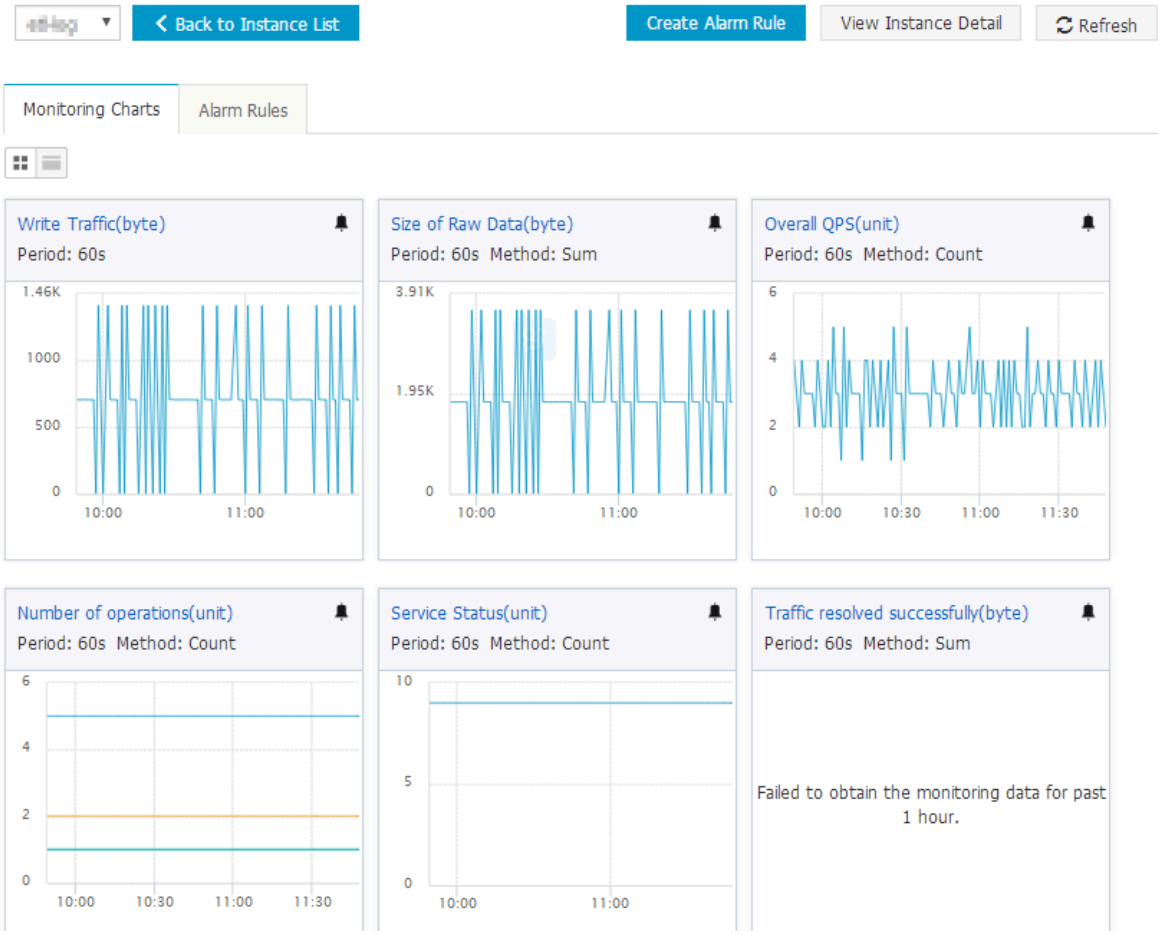
1. Log on to the Log Service console.
2. On the Project List page, click the project name.

3. Click the Monitor icon at the right of the Logstore to enter the CloudMonitor console.

   You can log on to the CloudMonitor console directly and then click Cloud Service > Log Servicein the left-side navigation pane to enter the monitoring configuration page.

   Monitor the log data in CloudMonitor. For more information, see *Log Service monitoring*.

   Figure 11-1: Monitoring item description



See

Log Service monitoring metrics.

Set alarm rules

Click Create Alarm Rule in the upper-right corner of the Monitoring Charts page. Configure the related resource, alarm rules, and notification method. For more information, see *Use CloudMonitor to set alarm rules*.

11.2 Service log

11.2.1 Service log overview

Alibaba Cloud Log Service provides the service log function, which supports recording various types of logs (including operational logs) and provides several dashboards for a variety of analysis dimensions. This function can help you gain real-time insights into the resource usage of Log Service, allowing you improve your overall O&M efficiency.

Limits

- A dedicated Logstore only stores logs generated by Log Service and does not support reading other data. Currently, there are no limits for queries, statistics, alarms, and stream consumption.
- RAM users can activate Log Service only after they are authorized by their corresponding Alibaba Cloud accounts.
- Logs generated by a Project can be stored in other Projects that are in the same region. However, storage across regions is not currently supported.
- Logs generated by the service log function follows the standard pricing policy of Log Service. The billing method is Pay-As-You-Go and a free quota is provided each month. For more information, see *Billing method*.
- If you want to disable the service log function, you can deselect the service logs check box of the Enable Operations Logs field. Log Service then forbids service log writing but retains historical service logs, which may result in fees. If you want to delete historical service logs, you can directly delete the Logstore that stores the logs.

Default configurations

Table 11-1: Default configurations

Default configuration item	Details
Logstore	By default, Log Service creates five Logstores for you, each of which stores a different log type. For example: · `internal-operation_log`: Records operational logs and stores them for 30 days by default. This log type uses the common billing method. · `internal-diagnostic_log`: Records metering logs, consumption group delays, and Logtail-related logs based on topics. This Logstore stores the logs and information for 30 days by default and can be used free of charge. For more information about log types and fields, see *Log types*.
Region	· If you select Automatic creation (recommended) for the Log Storage field, Log Service creates a Project within the same region. · Service logs can be stored only in Projects within the current region.
Shard	By default, Log Service creates two Shards for each Logstore and enables the *automatic sharding* function.
Log storage duration	By default, logs are stored for 30 days. However, you can modify this setting. For more information, see *Manage a Logstore*.
Index	By default, the index function is enabled for all collected logs. If you do not need query, analysis, or alarm settings, you can disable the index function on the query page.

Default configuration item	Details
Dashboard	By default, Log Service creates five dashboards pertaining to the following: · User operations · Metering data · Logtail log collection · Logtail exception monitoring · Consumer group monitoring For more information, see *Dashboards*.

Scenarios

· View metering data

After you Log Service, Log Service immediately begins to calculate the storage space occupied by logs and indexes on an hourly basis and collects billing data (including the number of read and write operations and index traffic) for the current collection period and stores this information as metering logs. Metering logs are then stored in an independent Logstore. You can view the collected metering logs to gain insight into your storage and consumption information.

· Balance Shard writers and your overall consumption

You can compare write operations into Shards and overall consumption trends through predefined dashboards to determine whether your Shard write operations and consumption are balanced.

If there are multiple Logstores under a Project, the same Shards may be repeated for several Logstores. In such case, if you want to view the write distribution of a Shard in a specific Logstore, you can add the target Project and Logstore as filter conditions in the upper-left corner of the dashboard.

· Monitor API request status

All user operations (such as log writes, consumption, and the creation of Projects and Logstores) are performed through API requests. Every user operation generates a log in the internal-operation_log Logstore.

If a request fails, the `Status` field of the corresponding log is of a 3xx, 4xx, or 5xx status code. Therefore, you can determine whether API requests are normal

by monitoring the number of logs with a `Status` field of the preceding code formats.

- View the Logtail status

  By default, Log Service creates two Logtail-related dashboards, which are used for exception monitoring and data statistics. The exception monitoring dashboard shows specific exceptions, such as log parsing failure and regular expression mismatch.

11.2.2 Use the service log function

This topic describes how you can use the service log function. Specifically, it outlines how you can disable or enable the service log function, modify the log type and storage location. After you enable the server log function, Log Service stores all logs generated in the current Project to the specified Project. Note that this function is disabled by default.

Prerequisites

1. A Project is created.
2. Your RAM user account is authorized by your Alibaba Cloud account.

Overview

- This function records operational logs generated in a specified Project and Logtail alarm logs. It also allows you to store the logs in a new Project or a specified existing Project.
- This function automatically creates Logstores in the storage location you specified, separately storing operational logs and other logs.
- Log Service provides five types of dashboards so that you can view and monitor the running status of logs in real time.



Note:

- Log Service creates Logstores and dashboards in your specified storage location. Note that the Logstore that is used to store operational logs is charged according to the standard billing method, whereas the Logstore that is used to store other logs can be used free of charge.
- We recommend that logs generated within the same region be stored in the same Project automatically created by Log Service.

Enable the service log function

1. Log on to the Log Service console and find the target Project.
2. In the Actions column, click Operations Log.
3. In the displayed dialog box, select a log type or types, which best meet your requirements.

   You can select Operational logs or Other logs or both for the Enable Operations Log field.

   - Operational logs: Record all operations conducted on resources in a Project (including creation, modification, deletion, update, writes, and reads). They are stored in the internal-operation_log Logstore of a specified Project.
   - Other logs: Include metering logs at a Logstore granularity, consumption group delay logs, Logtail error information, heartbeat information, and statistical logs. They are stored in the internal-diagnostic_log Logstore of a specified Project.
4. Specify the Log Storage.
   - If you choose Automatic creation (recommended), Log Service will automatically create a Project named `log - service -{ user ID }-{ region }` in the current region. We recommend that all logs generated in this region be stored in this Project.
   - If you choose an existing Project, service logs will be stored in the specified Project.
5. Click Confirm.

Modify the log type and storage location

1. Log on with your Alibaba Cloud account to the Log Service console and find the target Project.
2. In the Actions column, click Operations Log.
3. Select the target option for the Enable Operations Log field.
4. Select the target Project for the Log Storage field.



Note:

- We recommend that service logs be stored in the Project that is automatically created by Log Service and that logs generated in the same region be stored in the same Project.

- After you modify the log storage location, newly generated logs will be stored in the specified Project. The logs stored in the original Project will not be migrated to the new Project or deleted. If you do not need historical logs, you can manually delete them.

Disable the service log function

 Note:
If the service log function is disabled, logs and dashboard information stored in the Project will not be deleted automatically. If you do not need them any more, you can delete the Project or Logstore where logs are stored manually.

1. Log on with your Alibaba Cloud account to the Log Service console and find the target Project.
2. In the Actions column, click Operations Log.
3. Deselect all options for the Enable Operations Log field.
4. Click Confirm.

Grant permissions to RAM users

Before using your RAM user account to use Log Service functions, you must use your Alibaba Cloud account to grant related permissions to the RAM user. For more information, see *Grant RAM sub-accounts permissions to access Log Service*. The following shows the corresponding policy:

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : CreateDash  board ",
        " log : UpdateDash  board "
      ],
      " Resource ": " acs : log :*:*: project /{ Project   where
 logs   are   stored }/ dashboard /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : GetProject ",
        " log : CreateProj  ect ",
        " log : ListProjec  t "
      ],
      " Resource ": " acs : log :*:*: project /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : List *",
```

```
        " log : Create *"
        " log : Get *",
        " log : Update *",
      ],
      " Resource ": " acs : log :*:*: project /{ Project   where
 logs   are   stored }/ logstore /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log :*"
      ],
      " Resource ": " acs : log :*:*: project /{ Project   for   which
   the   service   log   function   is   enabled }/ logging ",
      " Effect ": " Allow "
    }
  ]
}
```

11.2.3 Log types

The service log function records multiple log types. This topic describes the fields used by each type of log in detail.

Log types

When you enable the service log function, you can choose one of the following log types:

- Operational logs: Record all operations to yourresources in a Project, including create, modify, delete, update, write, and read operations. They are stored in the internal-operation_log Logstore of a specified Project.
- Other logs: Include metering logs at a Logstore granularity, which are consumption group delay logs, Logtail error information, heartbeat information, and statistical logs. They are stored in the internal-diagnostic_log Logstore of a specified Project.

Log type	Logstore	Log source	Description
Operational logs	internal-operation_log	*User operational logs*	All API requests and operational logs, including requests sent from consoles, consumer groups, SDKs, and clients.

Log type	Logstore	Log source	Description
Other logs	internal-diagnostic_log	*Consumer group snapshot logs*	The consumption delay logs of a consumption group, which are reported every 2 minutes. To query snapshot logs of a certain consumption group, you need to specify `__topic__ : consumergr oup_log` in the query statement.
		Logtail alarm logs	The Logtail error logs, which are recorded every 30 seconds. The errors of the same type occurring within 30 seconds only accumulate the total number of errors, but only one error message is randomly selected and sent. To query a certain alarm log, you need to specify `__topic__ : logtail_al arm` in the query statement.
		Logtail collection logs	The Logtail collection logs, which are recorded every 10 minutes. To query a certain Logtail collection log, you need to specify `__topic__ : logtail_pr ofile` in the query statement.
		Metering logs	The user metering logs, which are collected every hour. The logs include information relating to storage space at a Logstore granularity, read and write traffic, index traffic, and the number of requests. To query a certain metering log, you need to specify `__topic__ : metering` in the query statement.

Log type	Logstore	Log source	Description
		Logtail status logs	The status logs that are regularly reported by Logtail. The logs are recorded every minute. To query a certain Logtail status log, you need to specify `__topic__ : logtail_st atus` in the query statement.

User operational logs

Operational logs involve data read and write operations and other operations on various resources according to the `Method` field.

Type	Method
Data read operations	By calling the following APIs: · GetLogStoreHistogram · GetLogStoreLogs · PullData · GetCursor · GetCursorTime
Data write operations	By calling the following APIs: · PostLogStoreLogs · WebTracking
Other operations on resources	By calling the CreateProject and DeleteProject APIs.

Common fields

The following table lists common fields that can be used by various operations.

Field	Description	Example
APIVersion	The API version	0.6.0
InvokerUid	The account ID of the user who performs the operation	1759218115323050
NetworkOut	The inbound Internet read traffic in bytes	10

Field	Description	Example
Latency	The request delay in microseconds	123279
LogStore	The name of a Logstore	logstore-1
Method	The method being used	GetLogStoreLogs
Project	The name of a Project	project-1
NetOutFlow	The read traffic in bytes	120
RequestId	The request ID	8AEADC8B0AF2FA2592C9509E
SourceIP	The IP address of the client that sent the request	1.2.3.4
Status	The response status code	200
UserAgent	The user agent on the client	sls-java-sdk-v-0.6.1

Data read fields

The following tables lists the fields specific to read requests.

Field	Description	Example
BeginTime	The request start time in Unix timestamps	1523868463
DataStatus	The response status, including `Complete`, `OK`, and `Unknown`.	OK
EndTime	The request end time in Unix timestamps	1523869363
Offset	The offset of the GetLog request	20
Query	The original query statement	UserAgent: [consumer-group-java]*
RequestLines	The number of lines that are expected to be returned	100
ResponseLines	The number of lines of the query results	100

Field	Description	Example
Reverse	Indicates whether to return logs in reverse order of log timestamps, where: · 1 indicates the reverse order. · 0 indicates the normal order. The default value is 0.	0
TermUnit	The number of words included in a search statement	0
Topic	The name of the read topic	topic-1

Data write fields

The following tables lists the fields specific to write requests.

Field	Description	Example
InFlow	The number of the original write bytes	200
InputLines	The number of requested write lines	10
NetInflow	The number of write bytes after compression	100
Shard	The ID of the Shard to which data is written	1
Topic	The name of the topic to which data is written	topic-1

Consumption group snapshot logs

Field	Description	Example
consumer_group	The name of a consumption group	consumer-group-1

Field	Description	Example
fallbehind	The period of time from the current consumption point to the most recent write log (in seconds)	12345
logstore	The name of a Logstore	logstore-1
project	The name of a Project	project-1
shard	The ID of a Shard	1

Logtail alarm logs

Field	Description	Example
alarm_count	The number of alarms in the sampling window	10
alarm_message	The sample of original logs that triggered the alarm	M_INFO_COL,all_status_monitor,T22380,0,2018-04-17 10:48:25.0,AY66K,AM5,2018-04-17 10:48:25.0,2018-04-17 10:48:30.561,i-23xebl5ni.1569395.715455,901,00789b
alarm_type	The alarm type	REGISTER_INOTIFY_FAIL_ALARM
logstore	The name of a Logstore	logstore-1
source_ip	The IP address of the server at which Logtail runs	1.2.3.4
os	The operating system, such as Linux or Windows	Linux
project	The name of a Project	project-1
version	The Logtail version	0.14.2

Logtail collection logs

Based on the `file_name` field, Logtail collection logs can be divided into two types: single-file statistics and Logstore-level logs. For the second type, `logstore_statistics` in `file_name` indicates that the log collects statistics for the entire Logstore. For this log type, file-related fields, such as `file_dev` and `file_inode`,

can be disregarded. The following tables lists fields that are used in Logtail collection logs.

Field	Description	Example
logstore	The name of a Logstore	logstore-1
config_name	The name of a Logtail Config, which is unique and consists of `## Config version##Project name$Config name`.	##1.0##project-1$logstore-1
error_line	The raw log that caused the error	M_INFO_COL,all_status_monitor,T22380,0,2018-04-17 10:48:25.0,AY66K,AM5,2018-04-17 10:48:25.0,2018-04-17 10:48:30.561,i-23xebl5ni.1569395.715455,901,00789b
file_dev	The device ID of the log file	123
file_inode	The inode of the log file	124
file_name	The storage path of the log file or `logstore_statistics`	/abc/file_1
file_size	The size of the log file (in bytes)	12345
history_data_failures	The number of historical processing failures	0
last_read_time	The latest read time in the window (in Unix timestamps)	1525346677
project	The name of a Project	project-1
logtail_version	The Logtail version	0.14.2
os	The OS	Windows
parse_failures	The number of lines with a parsing failure in the log in the current window	12

Field	Description	Example
read_avg_delay	The average value of the difference between the offset at every read and the file size in the current window	65
read_count	The number of log reads in the current window	10
read_offset	The offset of the current file read (in bytes)	12345
regex_match_failures	The number of regular expression matching failures	1
send_failures	The number of request sending failures in the current window	12
source_ip	The IP address of the server where Logtail runs	1.2.3.4
succeed_lines	The number of log lines that are successfully processed	123
time_format_failures	The number of log time matching failures	122
total_bytes	The total number of reads (in bytes)	12345

The following table lists the fields that are valid only when `file_name` is `logstore_s tatistics`.

Field	Description	Example
send_block_flag	Indicates whether the sending queue is blocked at the end of the window.	false
send_discard_error	The number of discarded data packets due to data exceptions or permission unavailability in the current window	0

Field	Description	Example
send_network_error	The number of data packets that were not sent due to network errors in the current window	12
send_queue_size	The number of unsent data packets in the current send queue at the end of the window	3
send_quota_error	The number of data packets that were not sent due to quota insufficiency in the current window	0
send_success_count	The number of data packets that are successfully sent in the current window	12345
sender_valid_flag	Indicates whether the sending flag of the current Logstore is valid at the end of the window, where: · The value `true` indicates that the flag is valid. · The value `false` indicates that the flag may be forbidden due to network or quota errors.	true
max_send_success_time	The maximum time period during which data can be successfully sent in the window (in Unix timestamps)	1525342763
max_unsend_time	The maximum time period during which no data is sent in the sending queue at the end of the window (in Unix timestamps). The field is 0 when the queue is empty.	1525342764

Field	Description	Example
min_unsend_time	The minimum time period during which no data is sent in the sending queue at the end of the window (in Unix timestamps). The field is 0 when the queue is empty.	1525342764

Metering logs

Field	Description	Example
begin_time	The start time of a statistics window (in Unix timestamps)	1525341600
index_flow	The index traffic in the statistics window (in bytes)	12312
inflow	The write traffic in the statistics window (in bytes)	12345
logstore	The name of a Logstore	logstore-1
network_out	The outbound traffic from the statistics window to the Internet (in bytes)	12345
outflow	The read traffic in the statistics window (in bytes)	23456
project	The name of a Project	project-1
read_count	The number of data reads in the statistics window	100
shard	The average number of Shards used in the statistics window	10.0
storage_index	The total amount of the index storage in a Logstore at the statistics time point (in bytes)	10000000
storage_raw	The total number of logs in a Logstore at the statistics time point (in bytes)	20000000

Field	Description	Example
write_count	The number of data writes in the statistics window	199

Logtail status logs

Field	Description	Example
cpu	The load of a CPU	0.001333156
hostname	The host name	abc2.et12
instance_id	The instance ID, which is randomly assigned	05AFE618-0701-11E8-A95B-00163E025256_10.11.12.13_1517456122
ip	The IP address	1.0.1.0
load	The average system load	0.01 0.04 0.05 2/376 5277
memory	The amount of the memory occupied by the Logtail process (in MB)	12
detail_metric	The value of a metered item (in JSON format). For more information, see *detail_metric*.	*detail_metric*
os	The OS	Linux
os_cpu	The overall CPU usage	0.004120005
os_detail	Details relating to the OS	2.6.32-220.23.8.tcp1.34.el6.x86_64
status	The client status, which can be: · ok · busy · many_log_files · process_block · send_block · send_error For more information, see *Logtail Run Status*.	busy
user	The user name	root
user_defined_id	The user-defined ID	aliyun-log-id

Field	Description	Example
uuid	The UUID of the server	64F28D10-D100-492C-8FDC-0C62907F1234
version	The Logtail version	0.14.2
project	The Project to which the Logtail Config belongs	my-project

The following table lists the fields that are included in `detail_met ric`.

Field	Description	Example
config_count	The number of Logtail Configs	1
config_get_last_time	The time when the Config information was last obtained	1525686673
config_update_count	The number of Config updates after Logtail is started	1
config_update_item_count	The total number of configuration item updates after Logtail is started	1
config_update_last_time	The time when the Config is last updated after Logtail is started	1
event_tps	The TPS	1
last_read_event_time	The time when the event was last obtained	1525686663
last_send_time	The time when data was last sent	1525686663
open_fd	The number of files that are currently open	1
poll_modify_size	The number of files with listener modification events	1
polling_dir_cache	The number of scanned folders	1
polling_file_cache	The number of scanned files	1

Field	Description	Example
process_byte_ps	The number of logs processed per second (in bytes)	1000
process_lines_ps	The number of logs processed per second	1000
process_queue_full	The number of sending queues that have reached the maximum queue length	1
process_queue_total	The number of processing queues	10
process_tps	The processing TPS	0
reader_count	The number of files being processed	1
region	The region to which Logtail belongs	cn-hangzhou,cn-shanghai
register_handler	The number of folders registered with listeners	1
send_byte_ps	The number of raw logs sent per second (in bytes)	11111
send_line_ps	The number of logs sent per second	1000
send_net_bytes_ps	The amount of network data sent per second (in bytes)	1000
send_queue_full	The number of sending queues that have reached the maximum queue length	1
send_queue_total	The number of sending queues	12
send_tps	The sending TPS	0.075
sender_invalid	The number of abnormal sending queues	0

11.2.4 Service log dashboards

After the service log function is enabled, Log Service automatically creates five dashboards for displaying statistics relating to user operations, metering, Logtail log collection, Logtail exception monitoring, and consumption group monitoring.

Dashboard description

When enabling the service log function, you can choose from the following log types:

- If you select Operational logs, Log Service creates an *operation statistics* dashboard, displaying the statistics for the number of requests and the proportion of request failures on the current day.
- If you select Other logs, Log Service creates dashboards for displaying *metering statistics*, *Logtail log collection statistics*, *Logtail running monitoring*, and *consumption group monitoring*.

Operation statistics dashboard

This dashboard displays user access and operation information, such as the queries per second (QPS) and request delay for API requests and other operations on any of the resources under a Project.

Metering statistics dashboard

This dashboard displays metering statistics, which include data storage, index traffic, data read and write operations, and read and write traffic.

Logtail log collection statistics dashboard

This dashboard displays statistics relating to Logtail log collection.

Logtail running monitoring dashboard

This dashboard displays statistics for Logtail errors and alarms for you to monitor the health status of Logtail in real time.

Consumption group monitoring dashboard

This dashboard displays statistics for consumption groups, including Shard consumption data, consumption delays, and the consumption group list.

11.3 Log moniroring by CloudMonitor

11.3.1 Log Service monitoring metrics

For details about metric data, see *Monitor Log Service*.

1. Read/Write traffic
   - Meaning: Data traffic that is written to and read from each Logstore in real time . It makes statistics on the traffic that is written to and read from the specified Logstore through iLogtail, SDKs, and APIs in real time. The traffic volume is the volume of transferred data (or compressed data). The measurement period is one minute.
   - Unit: Bytes/min
2. Raw data size
   - Meaning: Volume of the raw data (before compression) written to each Logstore.
   - Unit: Byte/min
3. Total QPS
   - Meaning: Number of QPSs of all operations. The measurement period is one minute.
   - Unit: Count/min

4. Operation count

   · Meaning: Number of QPSs of various operations types. The measurement period is one minute.

   · Unit: Count/min

   · The following types of operations are measured:

     - Write:

       ■ PostLogStoreLogs: API later than 0.5

       ■ PutData: API earlier than 0.4

     - Keyword query:

       ■ GetLogStoreHistogram: Query of keyword distribution, which is an API later than 0.5.

       ■ GetLogStoreLogs: Query of keyword-matched logs, which is an API later than 0.5.

       ■ GetDataMeta: Same as GetLogStoreHistogram, which is an API earlier than 0.4.

       ■ GetData: Same as GetLogStoreLogs, which is an API earlier than 0.4.

     - Batch data acquisition:

       ■ GetCursorOrData: obtains cursors and data in batches.

       ■ ListShards: obtains all shards in a Logstore.

     - List:

       ■ ListCategory: same as ListLogStoreLogs, which is an API earlier than 0.4

       ■ ListTopics: traverses all topics in a Logstore.

5. Service status

   · Meaning: This view collects statistics on the QPSs that correspond to the HTTP status codes returned for all types of operations. You can locate the operation exception based on the return error code and adjust programs in a timely manner.

   · Status codes:

     - 200: is the normal return code, indicating that the operation is successful.

     - 400: indicates an error of one of the following parameters: Host, Content -length, APIVersion, RequestTimeExpired, query time range, Reverse,

AcceptEncoding, AcceptContentType, Shard, Cursor, PostBody, Parameter, and ContentType.

- 401: indicates that authentication fails because the AccessKey ID does not exist, the signature does not match, or the signature account has no permission. Check whether the project permission list on SLSweb contains the AccessKey.
- 403: indicates a quota overrun. For example, the maximum number of Logstores, shards, or read/write operations per minute is exceeded. Locate the specific error based on the returned message.
- 404: indicates that the requested resource does not exist. Resources include projects, Logstores, topics, and users.
- 405: indicates that the operation method is incorrect. Check the URL of the request.
- 500: indicates a Log Service error. Please try again.
- 502: indicates a Log Service error. Please try again.

6. Traffic successfully parsed by the agent
   - Meaning: size of the logs (raw data) successfully collected by Logtail
   - Unite: byte
7. Number of lines successfully parsed by the agent (Logtail)
   - Meaning: number of logs (counted by lines) successfully collected by Logtail
   - Unit: line
8. Number of lines the agent fails to parse
   - Meaning: number of lines Logtail fails to collect due to an error. An error occurs if this view has data.
   - Unit: line
9. Agent error count
   - Meaning: number of IP addresses that encounter an error when Logtail collects logs
   - Unit: count
10. Number of machines with an agent error
    - Meaning: number of alarms that indicate a collection error when Logtail collects logs
    - Unit: count

11.IP address error count (measured every 5 minutes)

- Meaning: number of IP addresses under various collection error categories, including:
  - LOGFILE_PERMINSSION_ALARM: The agent has no permission to access the log file.
  - SENDER_BUFFER_FULL_ALARM: Data is discarded because the data collection speed exceeds the network transfer speed.
  - INOTIFY_DIR_NUM_LIMIT_ALARM (INOTIFY_DIR_QUOTA_ALARM): The number of monitored directories exceeds 3,000. Please set the monitored root directory to a lower-level directory.
  - DISCARD_DATA_ALARM: Data is lost because the data time is 15 minutes earlier than the system time. Ensure that the time of the data written to log files is less than 15 minutes before the system time.
  - MULTI_CONFIG_MATCH_ALARM: When multiple configurations are applied to collect the same file, Logtail selects a configuration randomly for collection and no data is collected by other configurations.
  - REGISTER_INOTIFY_FAIL_ALARM: Inotify event registration fails. For details , view the Logtail log.
  - LOGDIR_PERMINSSION_ALARM: The agent has no permission to access the monitored directory.
  - REGEX_MATCH_ALARM: regular expression match error. Please adjust the regular expression.
  - ENCODING_CONVERT_ALARM: An error occurs when the log encoding format is converted. For details, view the Logtail log.
  - PARSE_LOG_FAIL_ALARM: log parsing error, which may be due to an incorrect regular expression at the beginning of the line or incorrect log splitting by line because the size of a single log exceeds 512 KB. For details, view the Logtail log. Adjust the regular expression if it is incorrect.
  - DISCARD_DATA_ALARM: Data is discarded because Logtail fails to write the data to the local cached file when the data cannot be sent to the Log Service. The possible cause is that the speed at which log files are generated exceeds the speed at which data is written to the cached file.
  - SEND_DATA_FAIL_ALARM: Logtail fails to send parsed logs to the Log Service . For details, view the error code and message related to data sending failures

in the Logtail log. Common errors include Log Service quota overruns and network exceptions at the agent side.

- PARSE_TIME_FAIL_ALARM: An error occurs when the time field of the log is parsed. The time field parsed by Logtail using the regular expression cannot be parsed based on the time format configuration. Please modify the configuration.
- OUTDATED_LOG_ALARM: Logtail discards historical data. Ensure that the difference between the time of currently written data and the system time is less than 5 minutes.

· Locate the specific IP address based on the error. Log on to the machine and view the /usr/logtail/ilogtail.LOG file to identify the cause.

11.3.2 Use CloudMonitor to set alarm rules

Log Service allows you to use CloudMonitor to set alarm rules. An alarm SMS or email is sent when the service status meets the configured alarm rules. Configure the alarm rules to monitor Log Service in the CloudMonitor console. Then, you can monitor the log collection status of Logtail, shard usage status, and write traffic of projects.

Procedure

On the CloudMonitor console, click xCloudMonitor console > Log Serviceclick Alarm Rules at the right of the Logstore. Then, click Create Alarm Rule in the upper-right corner.

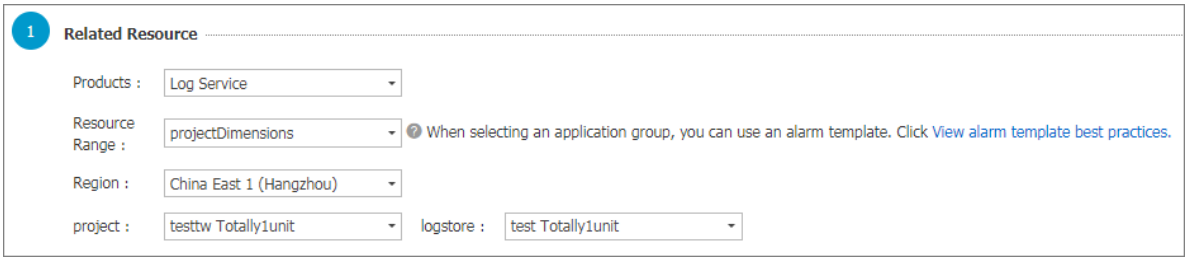
1. Configure the related resource.
   a. From the Products drop-down list, select Log Service.
   b. Select the resource range.

      You can select All Resources, Application Group, or projectDimensions.

      - All Resources – An alarm notification is sent when any instance in Log Service meets the alarm rules.
      - Application Group - An alarm notification is sent when any instance in an application group meets the alarm rules.
      - projectDimensions - An alarm notification is sent only when the selected instances meet the alarm rules.

   c. Select the region.
   d. Select one or more Projectand Logstore. You can select one or more projects and logstores.

Figure 11-2: Associated resources



2. Set the alarm rules.

   You can set one or more alarm rules.

   a. Enter the alarm rule name.
   b. Configure the rule description.

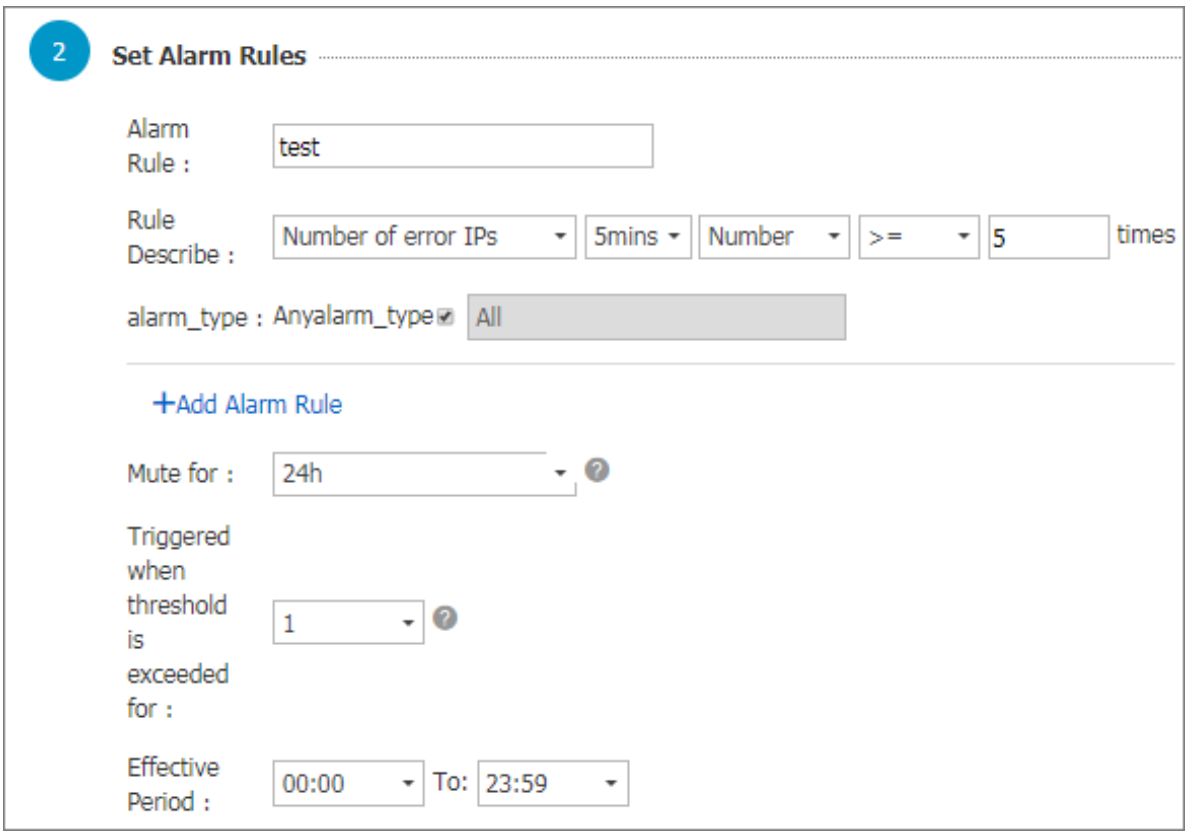
      Define your monitoring policy here by selecting the monitoring item and configuring the threshold for the monitoring item. CloudMonitor sends an alarm notification when the threshold is exceeded.

      For more information about the description of each monitoring item, see *Log Service monitoring metrics*. For more information about the statistical method, see *Monitor Log Service*.
   c. Select thealarm_type. By default, Any alarm_type is selected.
   d. Set the mute time , which is the time interval between two alarm notifications if the condition that triggers the alarm is still abnormal after an alarm notification is sent.
   e. Select a number from the Triggered when threshold is exceeded for drop-down list. The alarm is triggered after the threshold is exceeded for the selected number of times successively, that is, the alarm is triggered after the alarm

detection results meet your configured rule description for the selected number of times successively.

f. Select the effective period for your monitoring policy.The monitoring alarm policy only works within the selected period.

Figure 11-3: Set alarm rules



3. Configure the notification method.

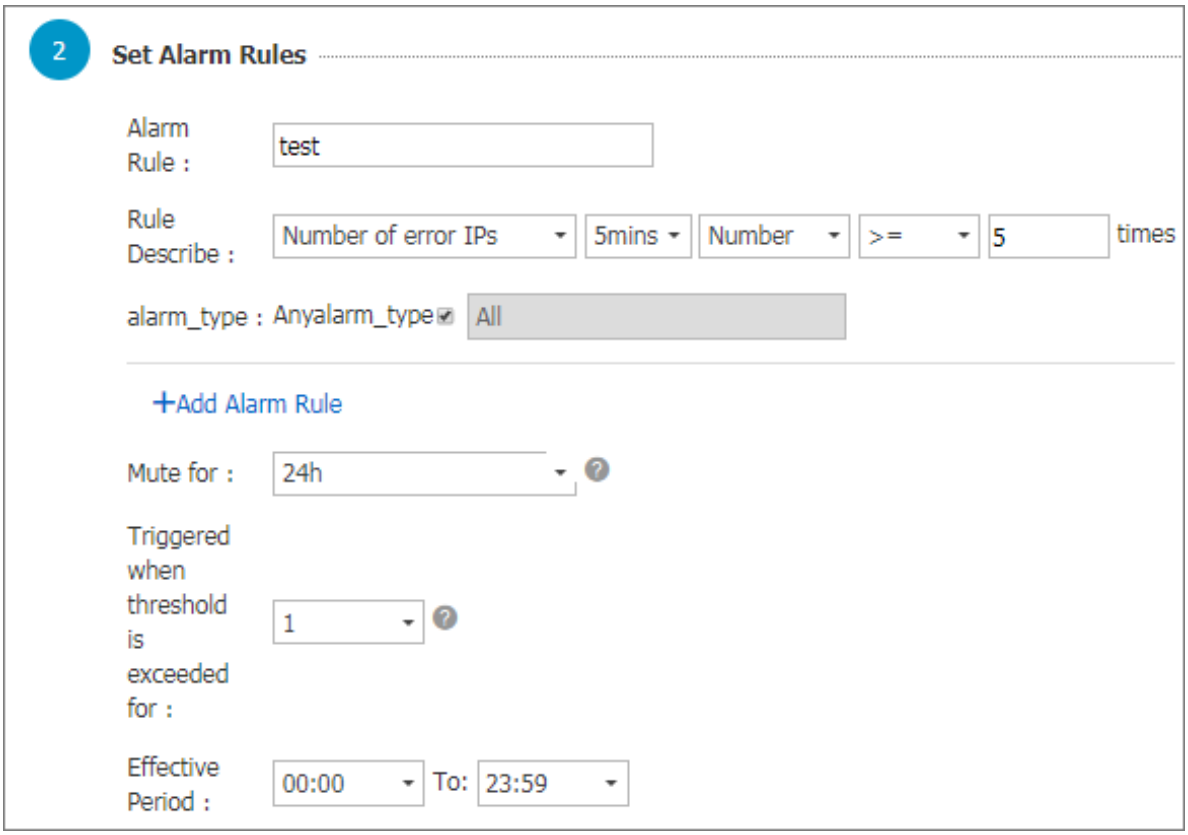
   a. Notification contact. Send a notification in the contact group level.

   b. Alarm level. Select Warningor Info as per your needs. Different levels have different notification methods.

   c. Notification subject and remark By default, the notification subject is the product name + monitoring item name + instance ID.

   d. HTTP callback. Enter a URL that can be accessed by the Internet. CloudMonitor pushes the alarm notification to this address by using the POST request. Currently, only HTTP protocol is supported.

Figure 11-4: Notification Method



Click Confirm after the configurations to complete the configuration of monitoring policy.

Example

Monitor log collection status of Logtail

Errors may occur because of incorrect configurations when Logtail is running. For example, some log formats do not match or a log file is repetitively collected. For

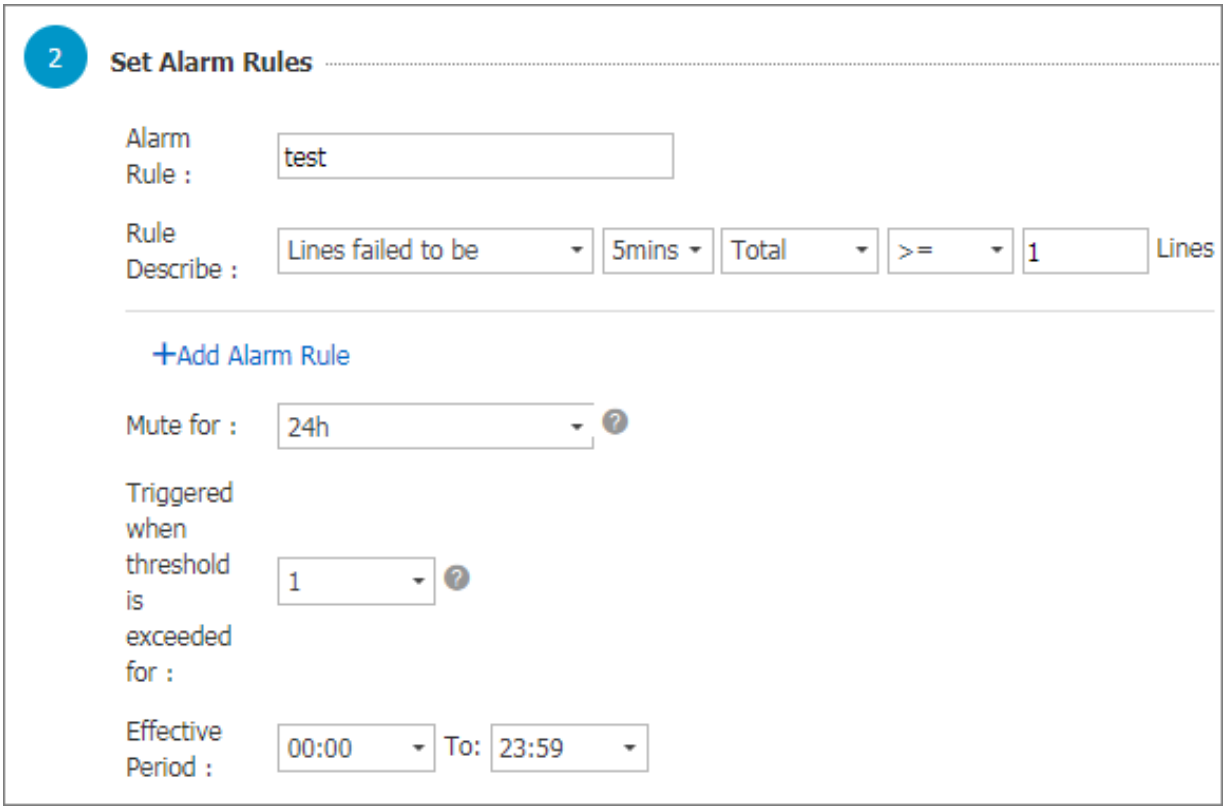
more information, see Basic questions of Logtail. To find such errors in time, you can monitor the metrics such as lines failed to be resolved and number of errors on Logtail.

The monitoring rule configuration is as follows:

Enter the alarm rule name and configure the rule description. Select Lines failed to be resolved or Number of errors as needed. Configure the rule items such as statistical period and method. You can also set alarm rules based on other errors of Logtail. Then, you can find the log collection errors in time.

The following figure shows that an alarm is triggered when the number of lines failed to be resolved within five minutes is greater than one. The monitoring lasts 24 hours.

Figure 11-5: Monitor logtail log collection status



Monitor shard usage status

Each shard in a Logstore provides the write capability of 5 MB/s (500 times per second), which is sufficient in most cases. When the capability limit is exceeded, Log Service attempts to serve (rather than deny) your requests, but does not guarantee the availability of data that exceeds the limit during peak hours. You can detect this situation by setting an alarm rule on Logstore outbound and inbound traffic. If your

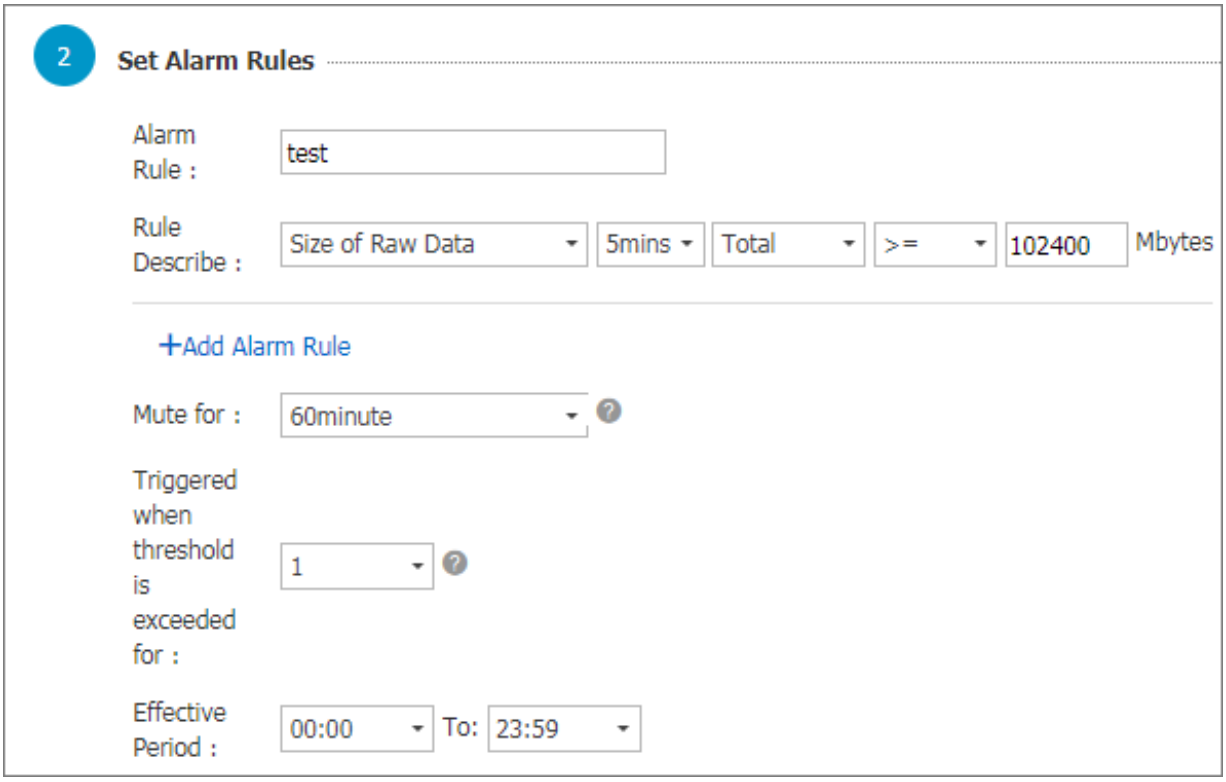
data volume is large and needs more shards, adjust the number of shards in the console in time.

Use the following solutions to set an alarm rule on Logstore traffic.

Solution 1: Set an alarm rule on traffic

Enter the alarm rule name. Select Size of Raw Data. Configure the statistical period and method. For example, to trigger the alarm when 100 GB/5 minutes is exceeded, set the rule description to 5 mins, Total, >=, and 102400, which means the alarm is triggered if the total traffic within five minutes exceeds 102400 MB.

Figure 11-6: Set up traffic alert

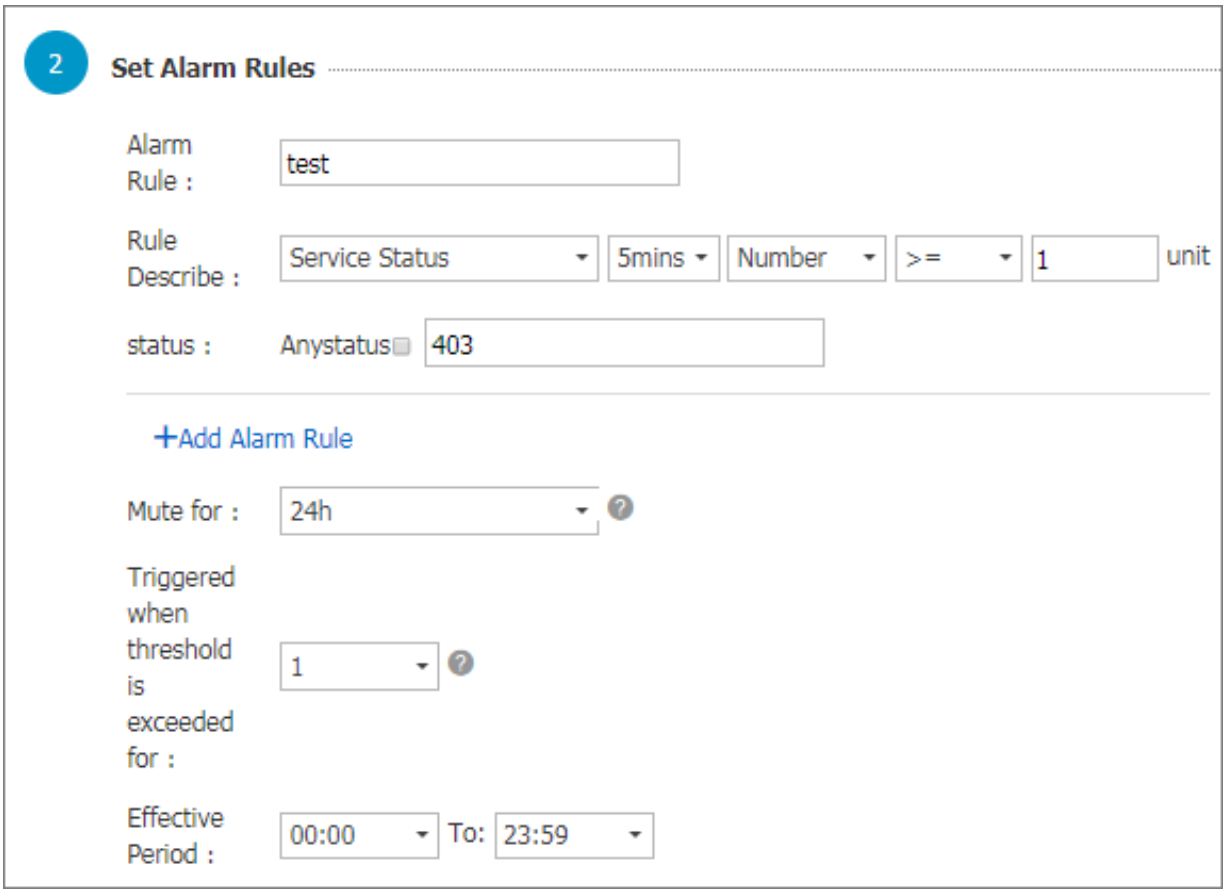


Solution 2: Set an alarm rule on service status

Enter the alarm rule name. Select Service Status. Configure the statistical period and method. For example, to trigger the alarm when 403 service status occurs more than

once within five minutes, set the rule description to 5 mins, Number of, >=, and 1, and enter 403 in the status field.

Figure 11-7: Set service status alarm



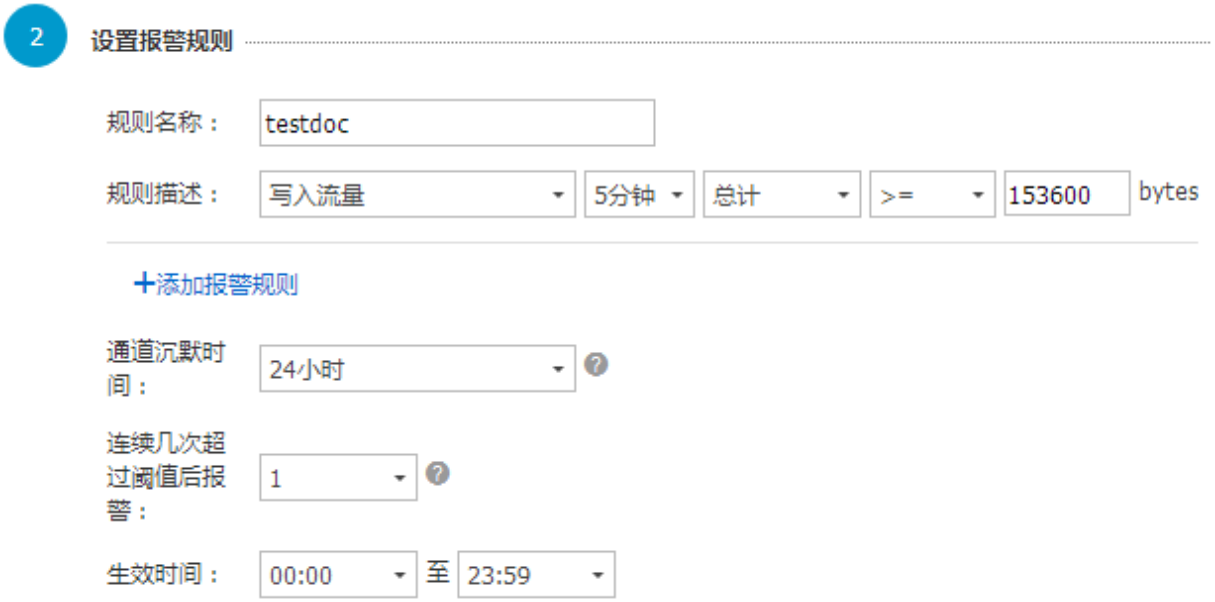
Monitor write traffic of projects

By default, each project provides the write capability of 30 GB/min (the size of raw data), which is used to protect you from generating large amounts of logs because of program errors. In most cases, this write capability is sufficient. The capability limit may be exceeded if your log volume is large. Open a ticket to increase the value.

Configure the monitoring policy of project quota as described in the following figure.

This example indicates that an alarm notification is sent when the write traffic within five minutes is greater than 150 GB.

Figure 11-8: Monitors write traffic for Project



12 Access control RAM

12.1 Authorization - Overview

Resource Access Management (RAM) is an Alibaba Cloud service designed to manage user identities and control resource access. By using RAM, you can create and manage user accounts (such as accounts of employees, systems, and applications) and control the operation permissions these user accounts have to resources under your account. If multiple users in your enterprise collaboratively work with resources, RAM allows you to avoid having to share the AccessKey of your Alibaba Cloud account with other users. Instead, you can grant users the minimum permissions necessary for them to complete their work, reducing the information security risks for your enterprise.

To precisely manage and perform operations on Log Service resources, you can use Alibaba Cloud RAM to grant corresponding access permissions to RAM service roles and user roles of Log Service, and your sub-accounts.

Manage user identities

You can use RAM to manage user identities. For example, you can create and manage user accounts or user groups under your account, create service roles to represent Log Service, and create user roles to perform resource operations and manage authorization across accounts.

Log Service supports collecting logs from cloud products such as API Gateway and Server Load Balancer. You must create and authorize the service roles in the quick authorization page before the configuration.

Role	Default permissions	Description
AliyunLogArchiveRole	AliyunLogArchiveRole Policy	Log Service uses this role by default to access your Server Load Balancer logs. By default, the authorization policy is used to export Server Load Balancer logs. For quick authorization, go to the *quick authorization page*.

Role	Default permissions	Description
AliyunLogDefaultRole	AliyunLogRolePolicy	The authorization policy is used for the default role of Log Service, including the Object Storage Service (OSS) write permission. For quick authorization, go to the *quick authorization page*.
AliyunLogETLRole	AliyunLogETLRolePolicy	Authorization Policy for the log service ETL function role, by default, the log service uses this role to access your resources in other cloud products. For quick authorization, go to the *quick authorization page*.
AliyunMNSLoggingRole	AliyunMNSLoggingRole Policy	The Log Service uses this role to access your MNS cloud product logs by default, the default Authorization Policy is used to export MNS service logs that contain write permissions for OSS. For quick authorization, go to the *quick authorization page*.

RAM

You can grant corresponding authorization policies to user accounts or groups and roles under your account.

You can also create custom authorization policies or use custom and system authorization policies as templates to edit fine-grained authorization policies. For more information, see *Overview*.

Log Service supports the following system authorization policies:

Authorization Policy	Type	Description
AliyunLogFullAccess	System policy	All management permissions of Log Service
AliyunLogReadOnlyAccess	System policy	The read-only permission to Log Service.

Scenarios

Authorize a RAM sub-account to access Log Service

In actual use cases, a primary account may allocate the O&M jobs of Log Service to its RAM sub-users, enabling the sub-users to perform routine O&M on Log Service. Alternatively, sub-users under a primary account may need to access Log Service resources. In this case, the main account must authorize its RAM sub-accounts to access or perform operations in Log Service. For security reasons, we recommend that you grant the minimum permissions to RAM sub-accounts within the required scope.

For more information about the configurations, see *Grant RAM sub-accounts permissions to access Log Service*.

Authorize a service role to read logs

Log Service currently offers an alarm function that works with your log contents. To read log data, the service account of Log Service must be given explicit authorization to access your data.

For more information about the configurations, see *Service role*.

Authorize a user role to perform operations in Log Service

A RAM user role represents a virtual user without a fixed identity authentication AccessKey, and must be assumed by a trusted real user, such as an Alibaba Cloud account, RAM-User account, and cloud service account. After assuming a role, the real user receives a temporary security token of this RAM user role. Then, the user can use this security token to access the authorized resources as a RAM user role.

- Grant a trusted real user the operation permissions to Log Service and allow RAM roles under the real user to perform operations in Log Service. For more information about the configurations, see *Service role*.

- Authorize a mobile application client to access Log Service by means of direct connection, and directly upload the application logs to Log Service. For more information about the configurations, see *Authorize a mobile application to directly connect to Log Service*.

12.2 Grant a RAM user the permissions to access Log Service

Context

In actual scenarios, a main account may allocate the Operation & Maintenance (O &M) jobs of Log Service to its Resource Access Management (RAM) sub-accounts, enabling the sub-accounts to perform routine O&M on Log Service. Alternatively, RAM sub-accounts under a main account may want to access Log Service resources . In this case, the main account must authorize its RAM sub-accounts to access or perform operations in Log Service. For security reasons, we recommend that you grant the minimum permissions to RAM sub-accounts within the required scope.

To authorize RAM sub-accounts to access Log Service resources by using a main account, follow these steps. For more information about the RAM sub-accounts, see *Introduction*.

Procedure

1. Create a RAM sub-account.
   a) Log on to the RAM console.
   b) Click Users in the left-side navigation pane. Click Create User in the upper-right corner.
   c) Enter the user information. Select the Automatically generate an AccessKey for this user check box and then click OK.
2. Grant sub-accounts permissions to access Log Service resources

   Log Service provides two system authorization policies: `AliyunLogF ullAccess` and `AliyunLogR eadOnlyAcc ess`. These two authorization policies respectively indicate the Full Access permission and Read-Only permission. You can also customize authorization policies in the RAM console. For how to create

an authorization policy, see *Create a custom authorization policy*. This document introduces how to grant the Read-Only permission to sub-accounts.

a) On the User Management page, click Authorize at the right of the sub-account. The Edit User-Level Authorization dialog box appears.

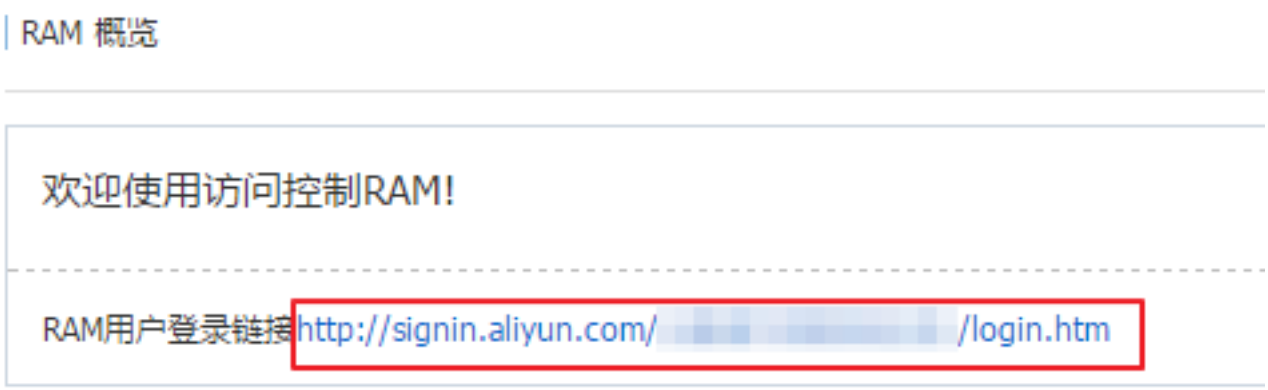
b) Select AliyunLogReadOnlyAccessunder Available Authorization Policy Names

3. Log on to the console as a sub-account

The sub-account has the permission to access Log Service console after being created and authorized. You can log on to the console as a RAM sub-account in the following ways:

a) On the RAM Overview page in the RAM console, click the RAM user logon link and use the sub-account username and password created in step 1 to log on to the console.

Figure 12-1: RAM user



b) Access the sub-account logon page and use the sub-account username and password created in step 1 to log on to the *console*.

By default, the Enterprise Alias is the account ID (ali uid) of the main account. You can view and configure your enterprise alias by navigating to Settings > Enterprise Alias Settings in the RAM console.

12.3 Custom RAM authorization

You can use RAM to grant permissions to RAM users under your Alibaba Cloud account.

Your Alibaba Cloud account can grant its RAM users the permissions to access or operate Log Service. You can grant *system policies* and *custom policies* to RAM users.

Precautions

- To maintain Log Service security, we recommend that you follow the principle of least privilege (PoLP). That is, do not grant RAM users any permissions beyond their requirements.
- In normal cases, you only need to grant RAM users the read-only permission for the project list so that they can view resources in the project list.
- `log : ListProjec t` provide the permission to view the project list.
  - RAM users with this permission can view all projects but cannot specify the project they want view.
  - RAM users without this permission cannot view any project.

This topic describes common custom authorization operations and policy content, including:

- *Read-only permission for the project list and specified project in the console*
- *Read-only permission for the specified Logstore and the permissions to create and use saved searches*
- *Read-only permission for all saved searches, dashboards, and the specified Logstore in the specified project in the console*
- *Permission to write data to the specified project through API calls*
- *Permission to consume the specified project through API calls*
- *Permission to consume the specified Logstore through API calls*

References:

- *Resources available for authorization*
- *Authorization operations*
- *Authorization rules*

Read-only permission for the project list and specified project in the console

If an Alibaba Cloud account needs to grant RAM users the following permissions:

1. Permission to view the project list under the Alibaba Cloud account
2. Read-only permission to the project specified by the Alibaba Cloud account

The policy that can grant RAM users both the permissions is as follows:

```
{
   " Version ": " 1 ",
```

```
   " Statement ": [
     {
       " Action ": [" log : ListProjec  t "],
       " Resource ": [" acs : log :*:*: project /*"],
       " Effect ": " Alow "
      },
     {
       " Action ": [
         " log : Get *",
         " log : List *"
       ],
       " Resource ": " acs : log :*:*: project /< name   of   the
 specified   project >/*",
       " Effect ": " Allow "
     }
   ]
 }
```

Read-only permission for the specified Logstore and the permissions to create and use saved searches

If an Alibaba Cloud account needs to grant RAM users the following permissions:

1. Permission to view the project list under the Alibaba Cloud account
2. Read-only permission for the specified Logstore and the permissions to create and use saved searches

The policy that can grant RAM users both the permissions is as follows:

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : ListProjec  t "
      ],
      " Resource ": " acs : log :*:*: project /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : List *"
      ],
      " Resource ": " acs : log :*:*: project /< name   of   the
 specified   project >/ logstore /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : Get *",
        " log : List *"
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ logstore /< name   of   the   specified   Logstore >"
      ],
      " Effect ": " Allow "
    },
    {
```

```
      " Action ": [
        " log : List *"
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ dashboard ",
        " acs : log :*:*: project /< name   of   the   specified
 project >/ dashboard /*"
      ],
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : Get *",
        " log : List *",
        " log : Create *"
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ savedsearc  h ",
        " acs : log :*:*: project /< name   of   the   specified
 project >/ savedsearc  h /*"
      ],
      " Effect ": " Allow "
    }
  ]
}
```

Read-only permission for all saved searches, dashboards, and the specified Logstore in the specified project in the console

If an Alibaba Cloud account needs to grant RAM users the following permissions:

1. Permission to view the project list under the Alibaba Cloud account
2. Permissions to view the specified Logstore and all saved searches and dashboards



Note:

If you want to grant the read-only permission for the specified Logstore to RAM users, you must also grant the RAM users the permission to view all saved searches and dashboards.

The policy that can grant RAM users both the permissions is as follows:

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : ListProjec  t "
      ],
      " Resource ": " acs : log :*:*: project /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : List *"
```

```
      ],
      " Resource ": " acs : log :*:*: project /< name   of   the
 specified   project >/ logstore /*",
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : Get *",
        " log : List *"
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ logstore /< name   of   the   specified   Logstore >"
      ],
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : Get *",
        " log : List *"
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ dashboard ",
        " acs : log :*:*: project /< name   of   the   specified
 project >/ dashboard /*"
      ],
      " Effect ": " Allow "
    },
    {
      " Action ": [
        " log : Get *",
        " log : List *"
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ savedsearc  h ",
        " acs : log :*:*: project /< name   of   the   specified
 project >/ savedsearc  h /*"
      ],
      " Effect ": " Allow "
    }
  ]
}
```

Permission to write data to the specified project through API calls

You can grant RAM users the permission to only write data to the specified project.

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : Post *"
      ],
      " Resource ": " acs : log :*:*: project /< name   of   the
 specified   project >/*",
      " Effect ": " Allow "
    }
  ]
```

```
}
```

Permission to consume the specified project through API calls

You can grant RAM users the permission to only consume data of the specified project .

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : ListShards ",
        " log : GetCursorO  rData ",
        " log : GetConsume  rGroupChec  kPoint ",
        " log : UpdateCons  umerGroup ",
        " log : ConsumerGr  oupHeartBe  at ",
        " log : ConsumerGr  oupUpdateC  heckPoint ",
        " log : ListConsum  erGroup ",
        " log : CreateCons  umerGroup "
      ],
      " Resource ": " acs : log :*:*: project /< name   of   the
 specified   project >/*",
      " Effect ": " Allow "
    }
  ]
}
```

Permission to consume the specified Logstore through API calls

You can grant RAM users the permission to only consume data of the specified project .

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : GetCursorO  rData ",
        " log : GetConsume  rGroupChec  kPoint ",
        " log : UpdateCons  umerGroup ",
        " log : ConsumerGr  oupHeartBe  at ",
        " log : ConsumerGr  oupUpdateC  heckPoint ",
        " log : ListConsum  erGroup ",
        " log : CreateCons  umerGroup "
      ],
      " Resource ": [
        " acs : log :*:*: project /< name   of   the   specified
 project >/ logstore /< name   of   the   specified   Logstore >",
        " acs : log :*:*: project /< name   of   the   specified
 project >/ logstore /< name   of   the   specified   Logstore >/*"
      ],
      " Effect ": " Allow "
    }
  ]
```

```
}
```

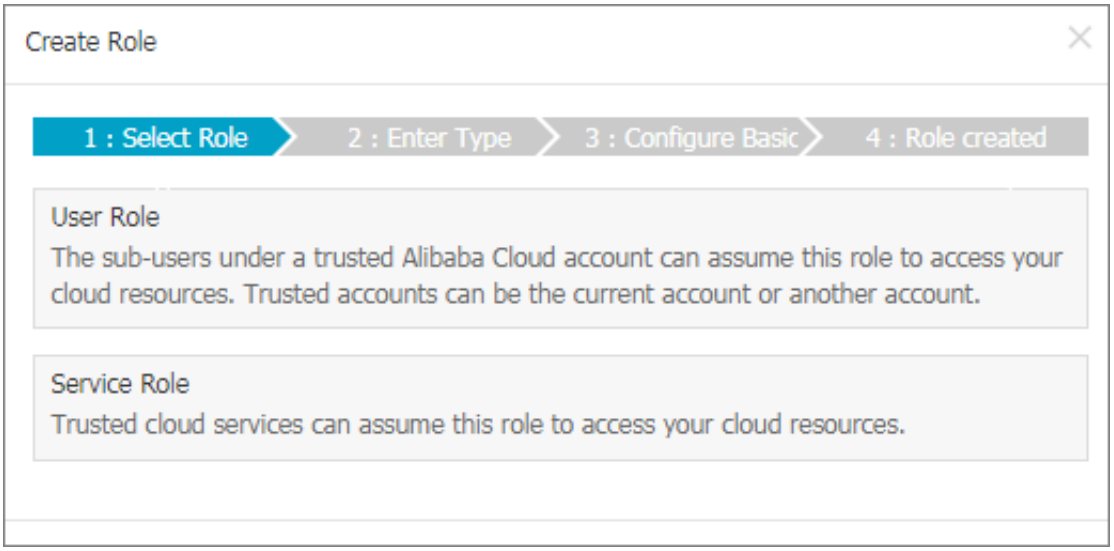
12.4 Service role

Log Service currently offers an alarm function that works with your log contents. To read log data, the service account of Log Service must be given explicit authorization to access your data. If you have read this document and completed the authorization , skip the following sections and create alarm rules directly. For how to authorize a service role, see the following sections.

Create a Resource Access Management (RAM) role

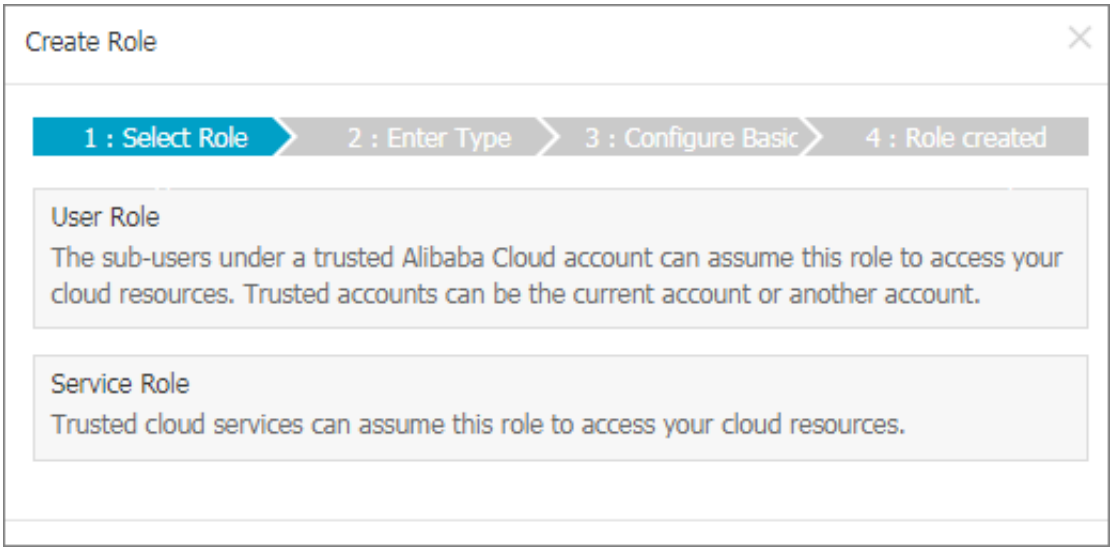
1. Log on to the Resource Access Management (RAM) console. Click Roles in the left-side navigation pane and click Create Role in the upper-right corner. The Create Role dialog box appears. Select Service Role in the Select Role Type step.

   Figure 12-2: Select the role type.



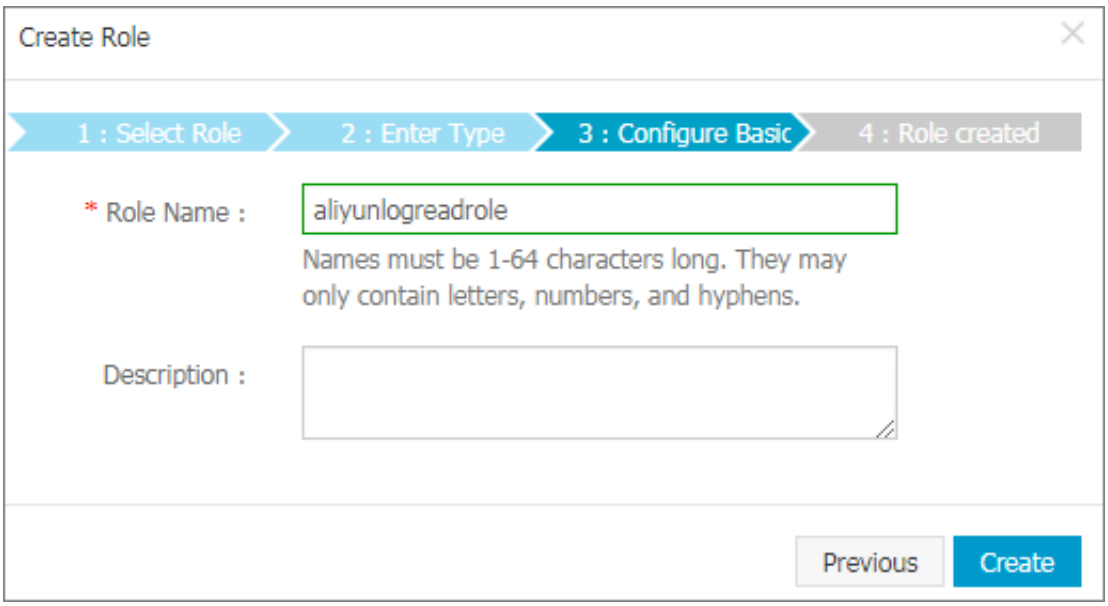
2. Select LOG Log Service in the Enter Type step.

Figure 12-3: Fill in type information



3. Enter aliyunlogreadrole in the Role Name field. (By default, this role is assumed to access data. Therefore, do not change this role name.) Then, click Create.

Figure 12-4: Configure basic role information.

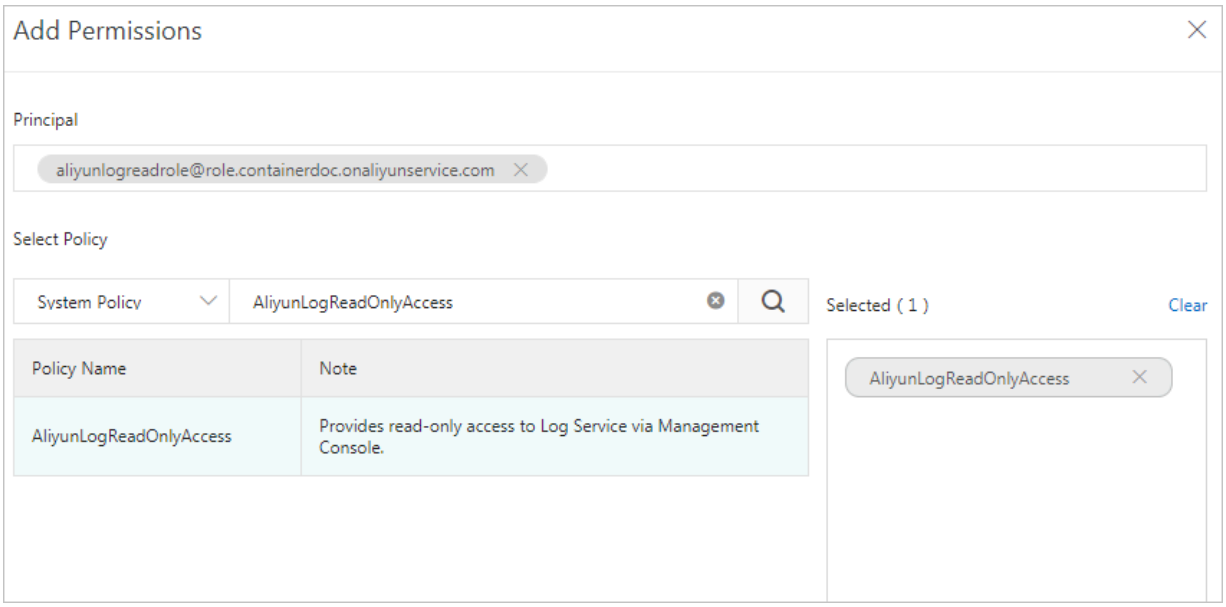


Authorize the role to access log data

After creating the role, click Authorize at the right of aliyunlogreadrole on the Role Management page. The Edit Role Authorization Policy dialog box appears. Search for the AliyunLogReadOnlyAccesspermission under Available Authorization Policy

Names. Select this permission and click 1 to add it to the Selected Authorization Policy Name. Then, click OK.

Figure 12-5: Edit Role authorization policy



Then, Log Service has the permissions to regularly read data from specified Logstores to perform alarm checks.

12.5 User Role

Rolethe same as ***User***, is an identity used in Resource Access Management (RAM). Compared with a RAM user, a RAM user role is a virtual user without a fixed authentication AccessKey, and must be assumed by a trusted real user, such as an Alibaba Cloud account, RAM user account, and cloud service account. After assuming a role, the real user receives a temporary security token of this RAM user role. Then, the user can use this security token to access the authorized resources as a RAM user role.

To grant a trusted real user the operation permissions to Log Service and allow RAM roles under the real user to perform operations in Log Service, you must create a RAM user role, specify the trusted Alibaba Cloud account, authorize the RAM user role, grant the AssumeRole permission to RAM users under the trusted account, and obtain the temporary security token of the RAM user role.

For more information, see *User*.

Step 1. Create a user role and specify the trusted Alibaba Cloud account

1. Log on to the RAM console. Click Roles in the left-side navigation pane
2. and click Create Role in the upper-right corner. The Create Role dialog box appears.
3. . Select User Role in the Select Role Type step.
4. Select the trusted Alibaba Cloud account in the Enter Type step.

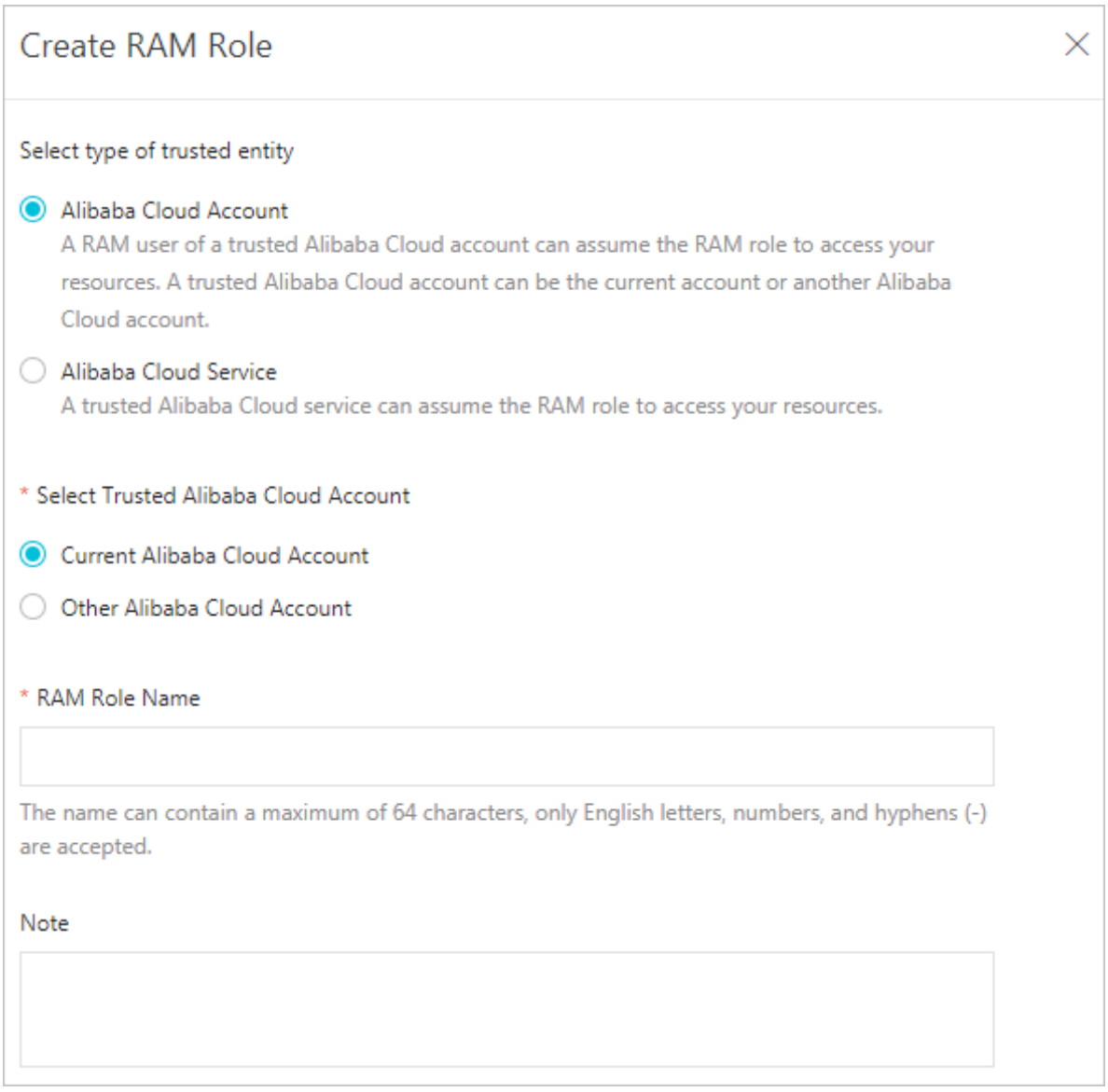


Note:

- If the role you create is to be used by the RAM users under your account, such as authorizing a mobile application client to directly perform operations on Log Service resources, select Current Alibaba Cloud Account as the trusted Alibaba Cloud account.
- If the role you create is to be used by the RAM users under another Alibaba Cloud account, such as resource authorization across accounts, select Other

Alibaba Cloud Account and enter the ID of another Alibaba Cloud account in the Trusted Alibaba Cloud Account ID field.

Figure 12-6: Create a role



5. Enter the Role Name and Description in the Configure Basic Information step, and click Create.

Step 2. Authorize the RAM user role

The created user role does not have any permission. You must grant the RAM user role the operation permissions to Log Service. The trusted Alibaba Cloud account specified in the preceding step has the permission to assume the RAM user role to perform operations in Log Service.

 **Note:**
You can grant one or more authorization policies to the RAM user role, including system authorization policies and custom authorization policies. In this document, grant the RAM user role the permissions to manage Log Service.

1. In the RAM console, click Roles in the left-side navigation pane.
2. Click Authorize at the right of the target RAM user role name.
3. Select the system authorization policy AliyunLogFullAccess, and click OK.

For more information, see *Authorization*.

Step 3. Authorize the RAM user of the trusted Alibaba Cloud account

A RAM role must be assumed by an authorized real user for normal usage. However, a trusted real user cannot assume a RAM user role using its own identity, but as a RAM user only. that is, a RAM user role must and can only be assumed by a RAM user identity.

Besides, the trusted Alibaba Cloud account must grant the AssumeRolepermission to its RAM users. A RAM user can represent the trusted Alibaba Cloud account to assume the RAM user role created in step 1 only after being granted the permission to call the Security Token Service (STS) AssumeRole API.

1. Log on to the RAM console with the trusted Alibaba Cloud account.
2. On the User management page, click Authorize at the right of the RAM user.

   If you have not created a RAM user before, see the *User* RAM users to create one.
3. Select the system authorization policy AliyunSTSAssumeRoleAccess, and click OK.

Step 4. Obtain the temporary security token of the RAM user role

After a RAM user is granted with the AssumeRole permission, the user can use the access key to call the STS AssumeRole API to obtain an on-demand security token for this role. the temporary security token of a RAM user role.

For how to call the AssumeRole API, see *Getting started*.

After AccessKeyId, AccessKeySecret, and SecurityToken are obtained using STS SDK, log services can be accessed using log service SDK.

The following example uses AccessKey ID, AccessKey Secret, and SecurityToken to initiate LogClient. For Java SDK usage, see *Java SDK*.

```
package   sdksample ;
import   java . util . ArrayList ;
import   java . util . List ;
import   java . util . Vector ;
import   java . util . Date ;
import   com . aliyun . openservic  es . log . Client ;
import   com . aliyun . openservic  es . log . common .*;
import   com . aliyun . openservic  es . log . exception .*;
import   com . aliyun . openservic  es . log . request .*;
import   com . aliyun . openservic  es . log . response .*;
import   com . aliyun . openservic  es . log . common . LogGroupDa
ta ;
import   com . aliyun . openservic  es . log . common . LogItem ;
import   com . aliyun . openservic  es . log . common . Logs . Log ;
import   com . aliyun . openservic  es . log . common . Logs . Log .
Content ;
import   com . aliyun . openservic  es . log . common . Logs .
LogGroup ;
import   com . aliyun . openservic  es . log . common . Consts .
CursorMode ;
public   class   sdksample  {
    public   static   void   main ( String   args [])  throws
LogExcepti  on ,  Interrupte  dException  {
        String   endpoint  = "< log_servic  e_endpoint >"; //  Select
  the   endpoint   that   matches   with   the   region   where   the
  project   created   in   the   preceding   steps   resides .
        String   accessKeyI  d  = ""< your_acces  s_key_id >"; //
Use   the   AccessKey   ID   of   your   Alibaba   Cloud   account .
        String   accessKeyS  ecret  = ""< your_acces  s_key_secr  et
>"; //  Use   the   AccessKey   Secret   of   your   Alibaba   Cloud
  account .
    String   securityTo  ken  = ""< your_secur  ity_token >"; // Use
  the   SecurityTo  ken   of   the   role .
        String   project  = ""< project_na  me >"; //  The   name
of   the   project   created   in   the   preceding   steps .
        String   logstore  = ""< logstore_n  ame >"; //  The   name
of   the   Logstore   created   in   the   preceding   steps .
       // Construct   a   client   instance .
        Client   client  =  new   Client ( endpoint ,  accessKeyI  d
,  accessKeyS  ecret );
   //  Set   SecurityTo  ken .
    client . SetSecurit  yToken ( securityTo  ken );
       //  Write   logs .
        String   topic  = "";
        String   source  = "";
       //  Send   10   packets   consecutiv  ely ,  with   each
packet   containing   10   logs
        for  ( int   i  =  0 ;  i  <  10 ;  i ++) {
            Vector < LogItem >  logGroup  =  new   Vector < LogItem
>();
            for  ( int   j  =  0 ;  j  <  10 ;  j ++) {
                LogItem   logItem  =  new   LogItem (( int ) ( new
Date (). getTime () /  1000 ));
                logItem . PushBack (" index "+ String . valueOf ( j
),  String . valueOf ( i  *  10  +  j ));
                logGroup . add ( logItem );
            }
             PutLogsReq  uest   req2  =  new   PutLogsReq  uest (
project ,  logstore ,  topic ,  source ,  logGroup );
```

```
                client . PutLogs ( req2 );
    }
    }
}
```