Alibaba Cloud Log Service

User Guide

Issue: 20180820

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.	 **Note:** 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]*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *{stand | slave}*

Contents

1 Preparation

1.2 Manage a Project

In the Log Service console, you can: create a project and 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 10 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.

Configuration items	Description
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 may delete a project in some situations, such as disabling Log Service and destroying 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** at the right of the project you are about to 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:

- A Logstore must be created under a project.
- At most 10 Logstores can be created for each project in Log Service.

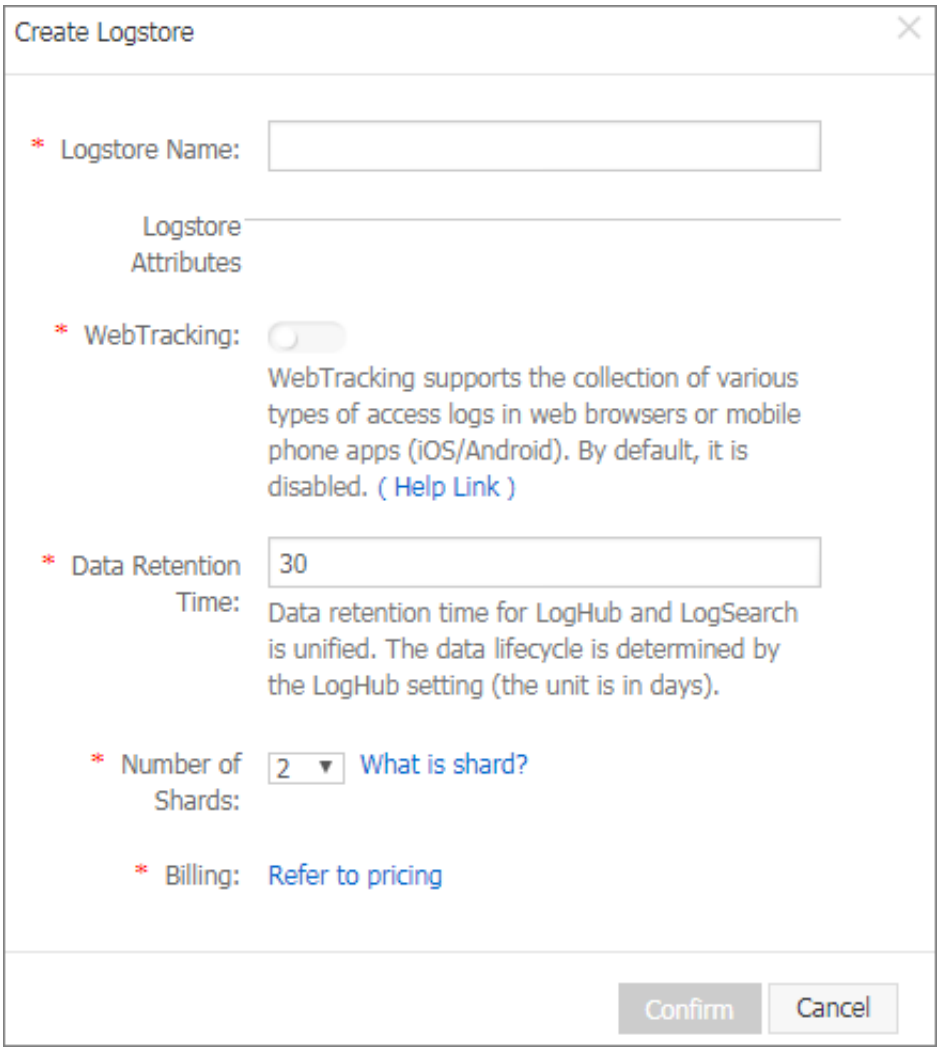
- The Logstore name must be unique in the project where it belongs.
- The data retention time can be modified after a Logstore is created. On the **Logstore List page**, click**Modify** > **Modify** , to change the **Data Retention Time**. And click **Modify**, and then close the dialog box.

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**.

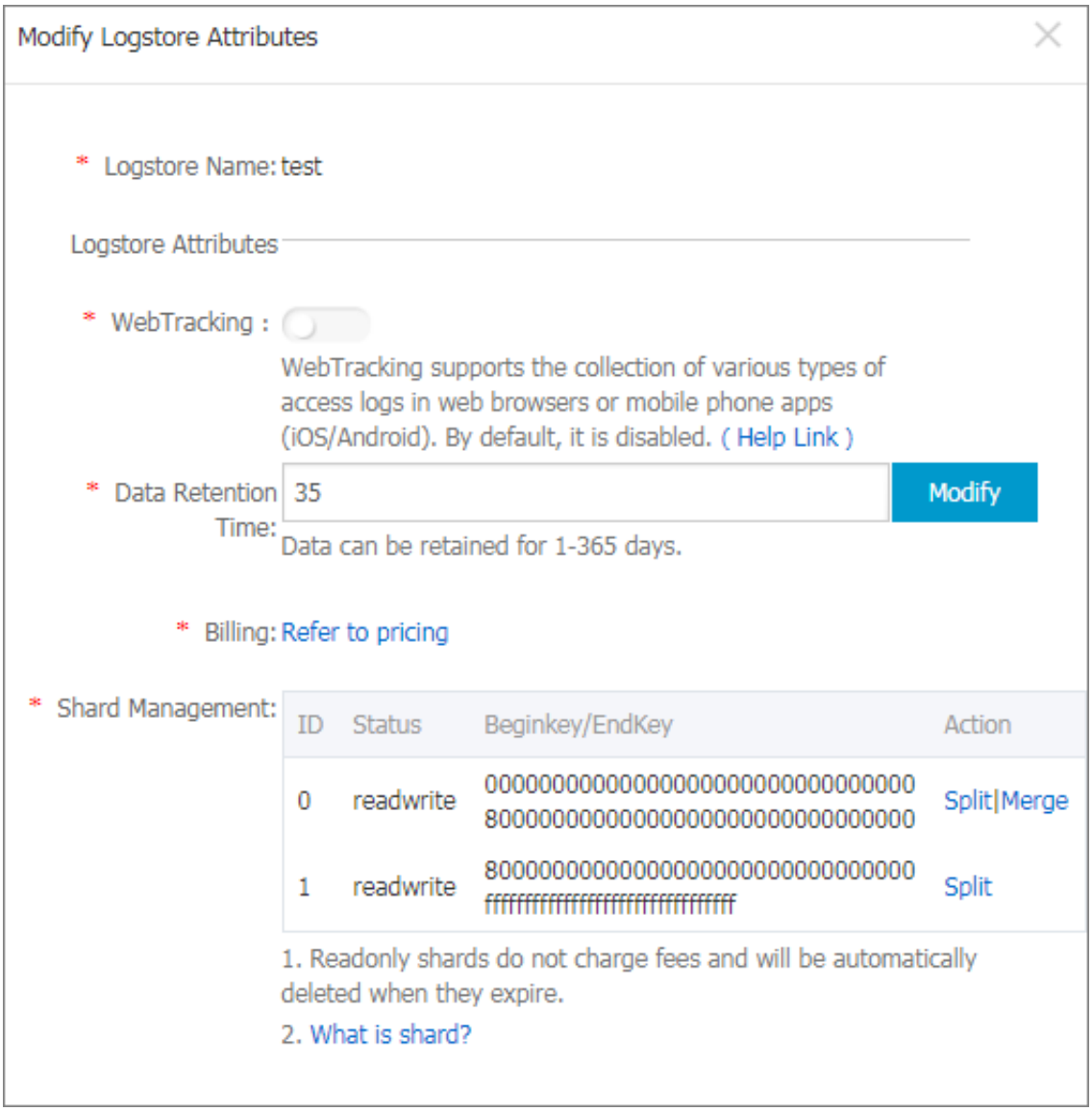
Configuration item	Description
Logstore name	The Logstore name, which must be unique in the project where it belongs. The name can be 3–63 characters long, contain lowercase letters, numbers, hyphens (-), and underscores (_), and must begin and end with a lowercase letter or number. Logstore The name must be unique in the project to which 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. The default is off.
Data Retention Time	The time (in days) the collected logs are kept in the Logstore. It can be 1–3650 days. Logs are deleted if the specified time is exceeded.
The number of shards for the Logstore.	Each Logstore can create 1–10 shards and each project can create at most 200 shards.



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 may delete a Logstore in some situations, such as abandoning a Logstore. Log Service allows you to delete a Logstore in the console. After a Logstore is deleted, the log data stored in the Logstore is permanently deleted and unrecoverable.

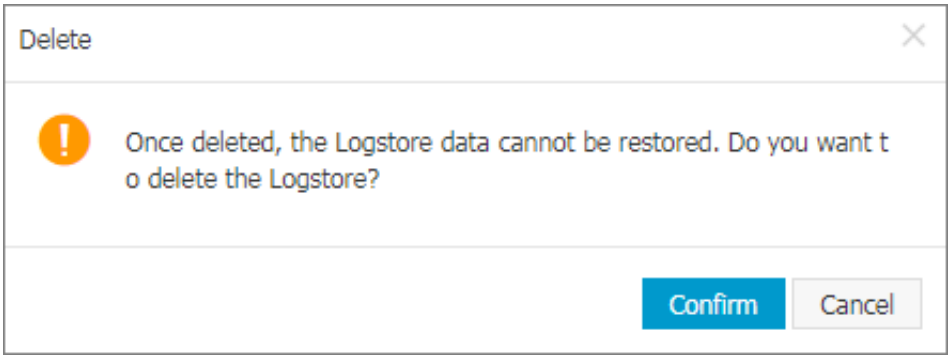
Note:

- So proceed with caution.
- Delete all the corresponding Logtail configurations before deleting a Logstore.
- If the consumption mode for log delivery is also enabled on the logstore, does not guarantee that all data in the previous logstore will be successfully delivered to maxcompute. If you need

to ensure that all data in the deleted logstore is delivered to maxcompute, follow the process below:

1. Stop writing a new log to the logstore before you delete it.
2. Verify that all log data in the logstore is successfully imported into maxcompute.
3. Delete logstore.

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 **Confirm** 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

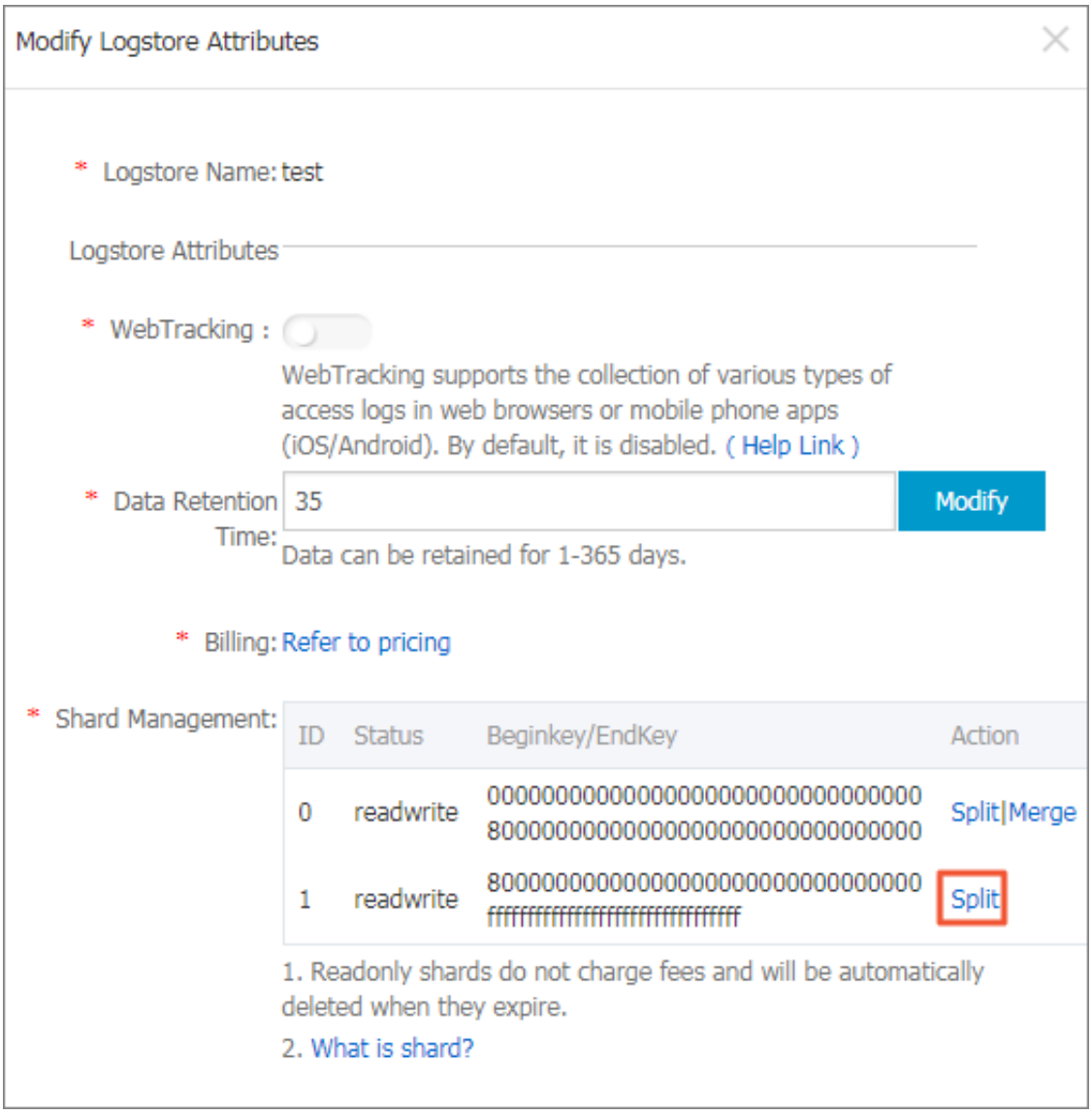
Each shard can write data at 5 MB/s and read data at 10 MB/s. When the data traffic exceeds the service capacity of the shard, we recommend that you increase the number of shards in time by splitting a shard. The expansion partition is completed by split operation.

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.

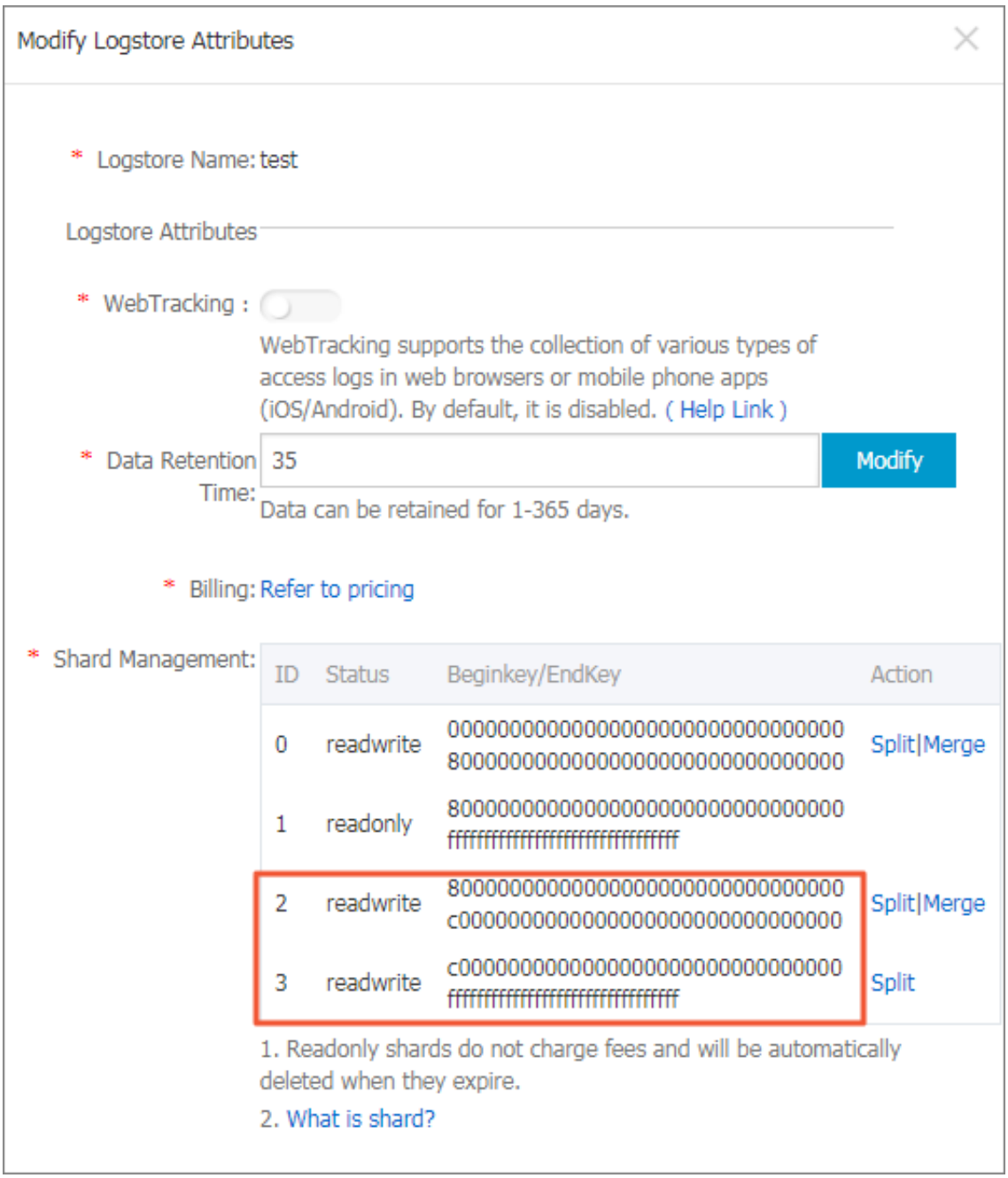
Split operations can split two other shards from one, that is, the number of shards is increased by 2 after the split. After the split, the status of the original shard specified to be split is changed from readwrite to readonly. Data can still be consumed, while new data cannot be written. The two newly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

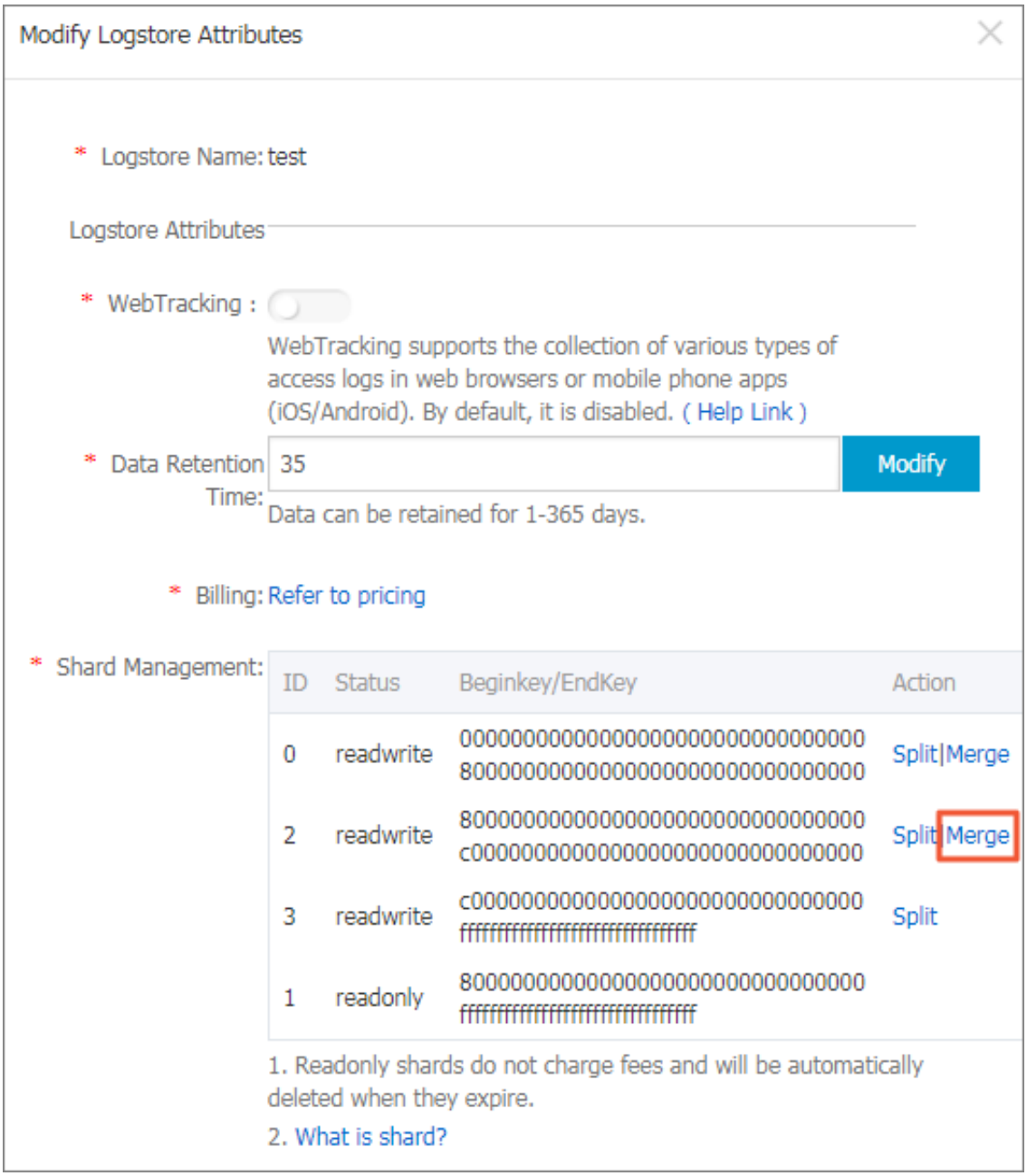
You can reduce the number of shards by merging shards. The ranges of the specified shard and the adjacent shard on the right are merged. A new shard in readwrite status is generated and its MD5 range covers the total range of the original two shards. The original two shards are now in the readonly status.

Instructions

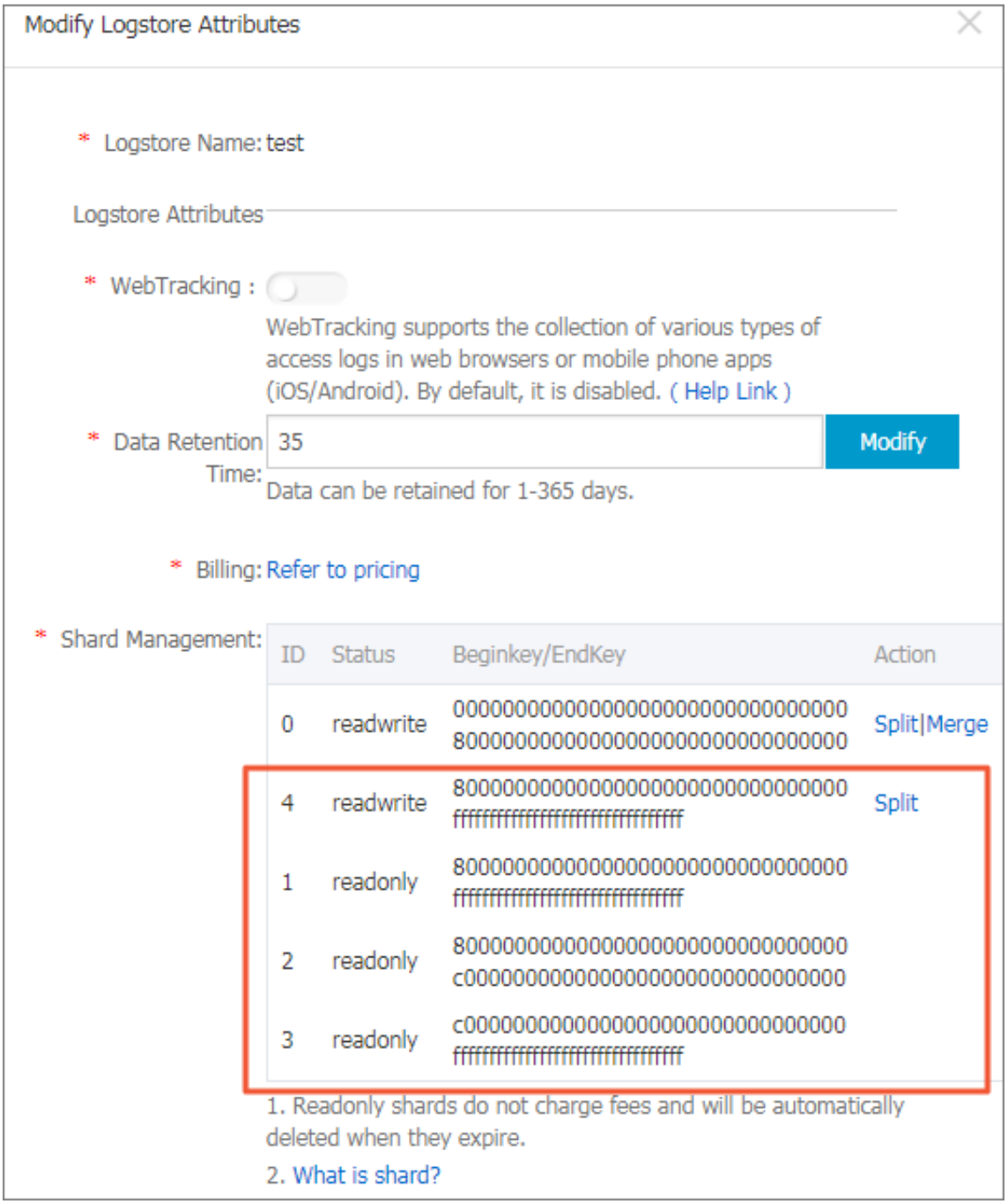
When merging shards, you must specify a shard in readwrite status. Make sure the specified shard is not the last shard in readwrite status. The server automatically finds the adjacent shard at the right of the specified shard and merges these two shards. After the merge, the specified shard and the adjacent 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 multiple methods to collect logs, such as by using clients, Web pages, protocols , and SDKs/APIs (mobile devices and games). All the collection methods are based on Restful API. You can also implement new collection methods by using APIs/SDKs.

Collection source

Type	Source	Access method	Others
Application	Program output	*Logtail*	*Use case*
	Access log	*Logtail*	*##Nginx##*
	Link tracking	Jaeger Collector, *Logtail*	
Language	Java	*SDK*,*C Producer Library*	
	Log4J Appender	*1.x*, *2.x*	
	LogBack Appender	*LogBack*	
	C.	*Native*	
	Python	*Python*	
	Python Logging	*Python logging Handler*	
	PHP	*PHP*	
	C#	*Go*	
	Go	*Go*	
	NodeJS	*NodeJs*	
	JS	*JS/Web Tracking*	
OS	Linux	*Logtail*	
	Windows	*Logtail*	
	Mac/Unix	*Native C*	
	Docker file	*Logtail Logtail file collection*	

Type	Source	Access method	Others
	Docker output	*Logtail Logtail container output*	
Mobile	iOS/Android	*Android SDK*,*Android SDK*	
	Web Page	*JS/Web Tracking*	
	Intelligent IoT	*C producer Library*	
	Syslog	*Logtail Syslog*	
Cloud Product	Elastic Compute Service (ECS) instance	*Logtail collection Introduction*	
	Container Service	*Logtail*, *FluentBit* (provided by the customer)	*Text*, *output*
	Object Storage Service (OSS)	Open a ticket to apply for a whitelist.	
	Server Load Balancer (SLB)	Activate on the product page.	*Introduction*
	Function Compute (FC)	Activate on the product page.	*Introduction*
	API Gateway (API)	Activate on the product page.	*Introduction*
	Message Service (MNS)	Open a ticket to apply for a whitelist.	
	MaxCompute	*Import DataWorks data*	
	WAF	*Open a ticket* to apply for a whitelist.	
	Situation Awareness	*Open a ticket* to apply for a whitelist.	
	Content Delivery Network (CDN)	*Open a ticket* to apply for a whitelist.	
Third-party software	Logstash	*Logstash*	

Select network and access point

Log Service provides #### in each region and each region provides two network access methods:

- Intranet (classic network)/private network (Virtual Private Cloud (VPC)): The service access in the current region, which has the best quality of link bandwidth (**recommended**).
- Internet (classic network): Accessible without any limits. The access speed depends on the link quality. We recommend that you use HTTPS to guarantee the transmission security.

FAQ:

- Q: **How to select the network for private line access?**

  A: Select the intranet/private network access point.
- Q: **How to select the network if I want to collect ECS logs from region A to the Log Service project in region B?**

  A: Install the Internet version Logtail of region B on the ECS instance in region A for Internet transmission.
- Q: **How to quickly determine whether it is accessible or not?**

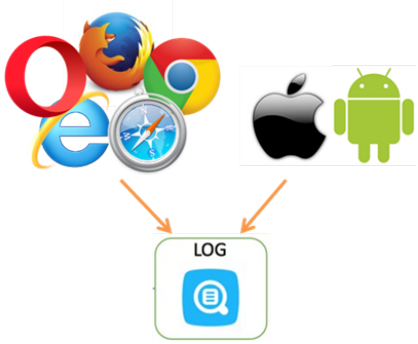
  A: Run the following command. It is accessible if any information is returned.

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

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

2.2 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 might be generated.

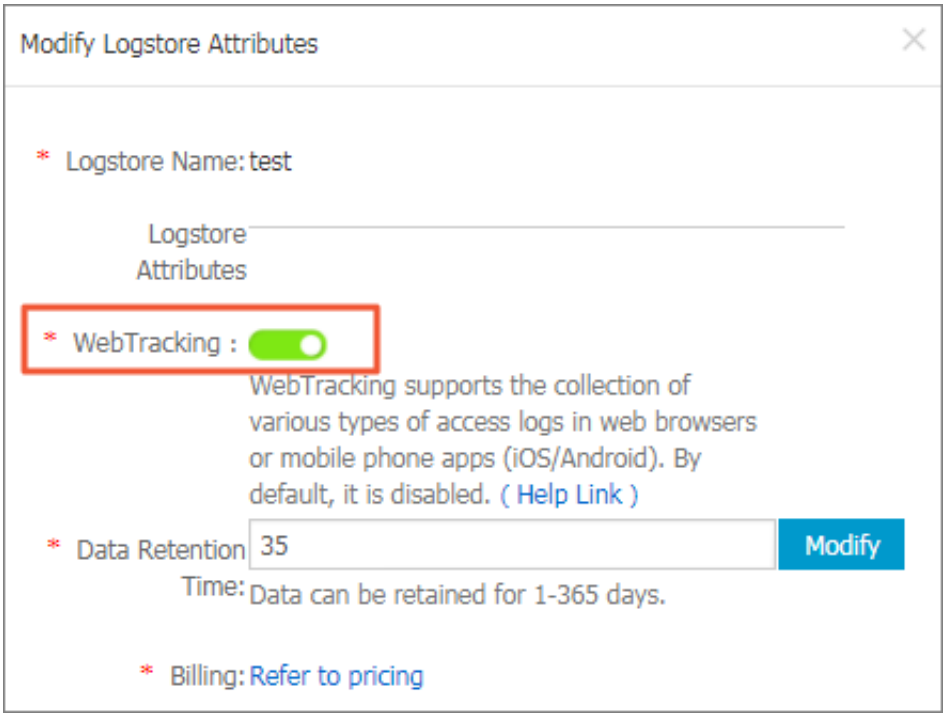
Procedure

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 needs to enable the Web Tracking function.
2. Turn on the Web Tracking switch.



- **Enable Web Tracking by using**

Java SDK:

```
import com.aliyun.openservices.log.Client;
```

```
import com.aliyun.openservices.log.common.LogStore;
import com.aliyun.openservices.log.exception.LogException;
Public class webtracking {
  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.GetLogStore(project, logStore).
GetLogStore();
          client.UpdateLogStore(project, new LogStore(logStore,
logSt.GetTtl(), logSt.GetShardCount(), true));
                    //Disable the Web Tracking function.
                    //client.UpdateLogStore(project, new LogStore(
logStore, logSt.GetTtl(), logSt.GetShardCount(), false));
                    //Create a Logstore that supports the Web
Tracking function.
                    //client.UpdateLogStore(project, new LogStore(
logStore, 1, 1, true));

      catch (LogException e){
          e.printStackTrace();
```

Step 2 Collect logs

After the Web Tracking function is enabled for Logstore, you can use any of the following three methods to upload data to the Logstore.

- **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	Meaning
`${project}`	The name of the project created in Log Service.
`${host}`	The domain name of the region where your Log Service is located.
`${logstore}`	Under `${project}` , the name of the . Logstore with the Web Tracking function enabled
`APIVersion=0.6.0`	The reserved field, which is required.

Field	Meaning
`__topic__=yourtopic`	The key-value pairs to be uploaded to Log Service. Multiple key-value pairs are supported,
such as `key1=val1` and `key2=val2`.	But you must make sure that the URL length is less than 16 KB.

- **Use 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.

- **Use 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 avoid this problem. To send requests synchronously, replace the statement in the script.

Original script:

```
this.httpRequest_.open("GET", url, true)
```

Replace the last parameter to send requests synchronously:

```
this.httpRequest_.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 meaning of each of these parameters is as follows:

Field	Meaning
`${host}`	The endpoint of the region where your logging service is located.
`${project}`	The name of the project that you opened in the log service.
`${logstore}`	The name of the logstore in `${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
```

After data is uploaded to Log Service, you can use Log Service to *ship* data to Object Storage Service (OSS). You can also use the LogHub Client Library provided by Log Service to consume data.

2.3 Logstash

2.3.1 Quick installation

You can choose to install logtash quickly on your server by default.

Context

Log Service provides an installation package based on Logstash 2.2.2, which integrates with JRE 1.8, Log Service write plug-in, and NSSM 2.24. The deployment process by using this package is simpler than *Custom installation*. You can select the custom installation for complex requirements.

Procedure

1. Download and extract the *installation package* to the C: drive.
2. Confirm the Logstash startup program path is *`C:\logstash-2.2.2-win\bin\logstash.bat`*.

2.3.2 Custom installation

You can install Logstash by using quick installation or custom installation methods.

Context

When you have other requirements for logstroudsburg's installation configuration, you can choose how you want to customize the installation, modify the default installation configuration.

Procedure

1. Install Java
   1. Download the installation package.

      Go to the *Java official website* to download JDK for installation.
   2. 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*
   3. 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
   1. Download the installation package from the official website.

      Select version 2.2 or later on the *Logstash* home page.
   2. Install Logstash.

      Extract *logstash-2.2.2.zip* to the *C:\logstash-2.2.2* directory.

      Confirm the Logstash startup program path is *C:\logstash-2.2.2\bin\logstash.bat* .
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-2.2.2> .\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 the *C:\logstash-2.2.2* directory of the machine. Run *PowerShell* or *cmd.exe* to go to the Logstash installation directory. Perform the following command to install lLogstash:

  ```
  PS C:\logstash-2.2.2> .\bin\plugin install C:\logstash-2.2.2\
  logstash-output-logservice-0.2.0.gem
  ```

- Perform verification.

  ```
  PS C:\logstash-2.2.2> .\bin\plugin list
  ```

  Verify that logstash-output-logservice exists in the installed plug-in list of the machine.

4. Install NSSM

   Download from the official website. Go to the *NSSM official website* to download the NSSM installation package.

   fter you download the installation package to the local machine, extract it to the *C:\logstash-2.2.2\nssm-2.24*.

2.3.3 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
```

2.3.4 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.
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

   1. 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
      ```

   2. 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 {

        }
      ```

   3. 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*.

2.3.5 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.

2.3.6 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 manipulated` 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.

2.4 SDK collection

2.4.1 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

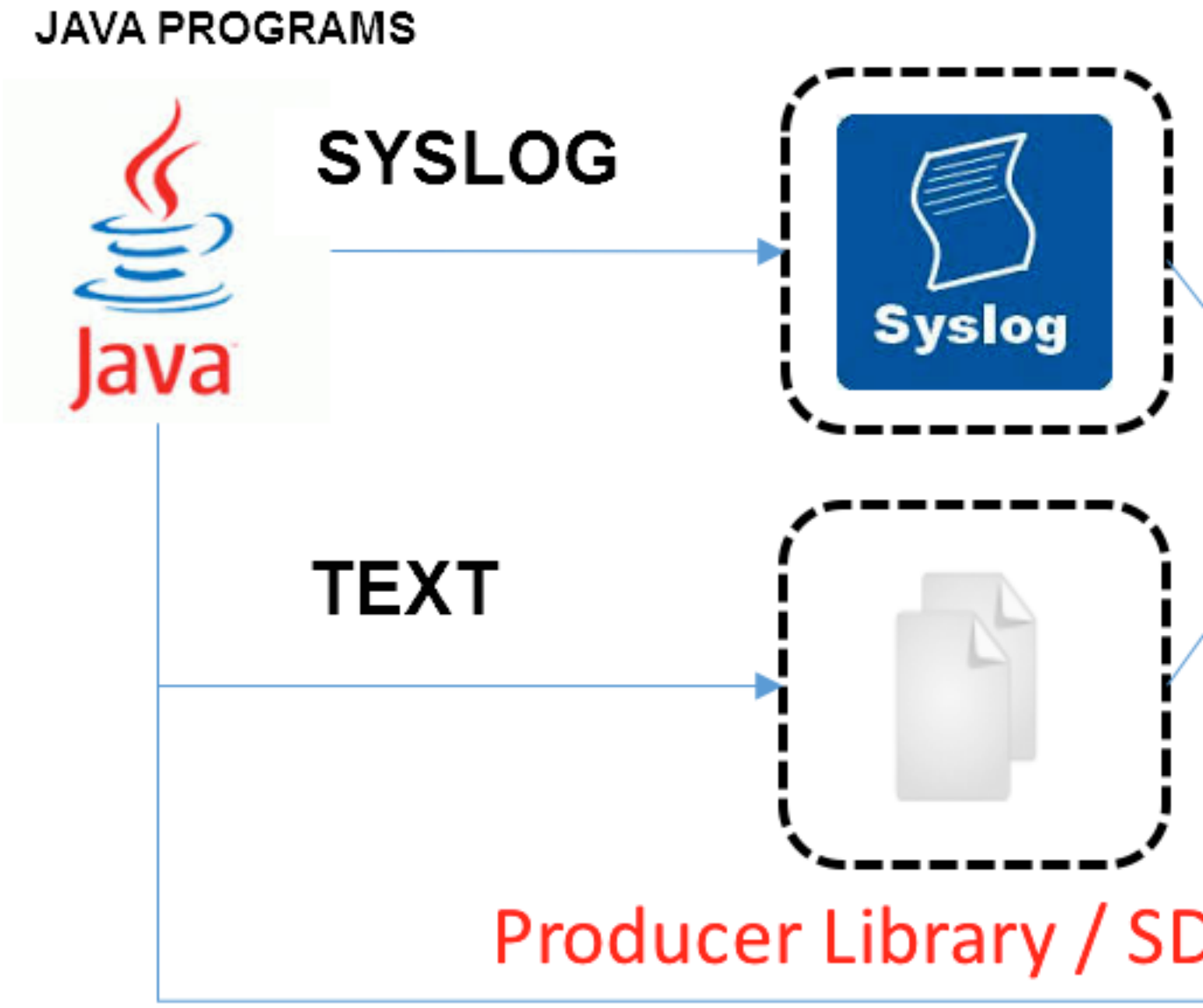
- Provides an asynchronous send interface to guarantee the thread security.
- Configurations of multiple projects can be added.
- The number of network I/O threads used for sending logs can be configured.

- 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 development steps, aggregates write requests in batches, and sends the requests to the LogHub server asynchrono usly. 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*

2.4.2 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*.

2.4.3 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)*

2.5 Common log formats

2.5.3 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 records logs in many methods, such as file, HTTP GET/POST, SMTP, and Socket. You can customize a log recording method as needed. The logging module has the same mechanism as Log4j except for the different implementation details. The logging module provides the logger, handler, filter, and formatter features.

To ccollect Python logs, we recommend you to use logging handler directly:

- *Automatically upload Python logs using log Handler*
- *Log handler automatically parses logs in Kv format*
- *Log handler automatically parses a log in JSON format*

Python log format

The log format specifies the specific output format of log recording in formatter. The construction 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. rotatingfilehandler (LOG_FILE, maxbytes
 = 1024*1024, backupcount = 5) # instated Handler
fmt = '%(asctime)s - %(filename)s:%(lineno)s - %(name)s - %(message)s'

formatter = logging.Formatter(fmt) # Instantiate the formatter
ormatter = logging.Formatter(fmt) # Instantiate the formatter
10.handler.setFormatter(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)ss` format, that is, replacing the dictionary keywords. The following keywords are provided.

Format	Description
%(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.
%(relativeCreated)d	The interval (in milliseconds) between the log created time and the time that the logging module is loaded.
%(asctime)s	The log created 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 created time in the millisecond level.
%(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
```

Configure Logtail to collect Python logs

For the detailed procedure of collecting Python logs by using Logtail, see ####### and *Apache # #*. 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.

See the following common Python logs and the corresponding regular expressions:

- Log sample:

  ```
  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 format:

  ```
  2016-02-19 11:06:52,514 - test.py:19 - 10 DEBUG test.py test <module
  > 1455851212.514271 139865996687072 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+(.
  ```

2.5.4 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.

```
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout 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">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

For how to configure Logtail to collect Log4j logs, see *#unique_49*. 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.PermanentTairDaoImpl
 - 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]
```

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]

2.5.6 WordPress logs

Default WordPress log format

Sample of raw logs:

```
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
.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-
admin/install.php" "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"
```

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

2.5.8 JSON logs

JSON logs are constructed in two structures:

- Object: A collection of key/value pairs.Object: a collection of name/value pairs).
- ray: 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 automatically parsing non-object data such as JSON arrays. Use regular expressions to extract the fields or use the simple mode to collect logs by line.

Log sample

```
{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek
********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&
Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%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=YunOsAccountOpLog&AccessKeyId=U0Ujpek
********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&
Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%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*. Select the corresponding configuration based on your network deployment and actual situation. This document only shows **how to configure data source** in Step 3 Configure data import wizard in details. Enter the Configuration Name and Log Path. Then, select JSON Mode as the log collection mode.

1. Click the data access wizard chart in the logstore list interface to enter the data access wizard.
2. Select the data type.

   Select the **text file** and click **Next**.
3. Configure the data source.

   a. Fill in the configuration name, Log Path, and select log collection mode as **JSON mode**.

   b. Select whether or not to use the system time as the log time according to your requirements. You can enable or disable the **Use System Time function**.

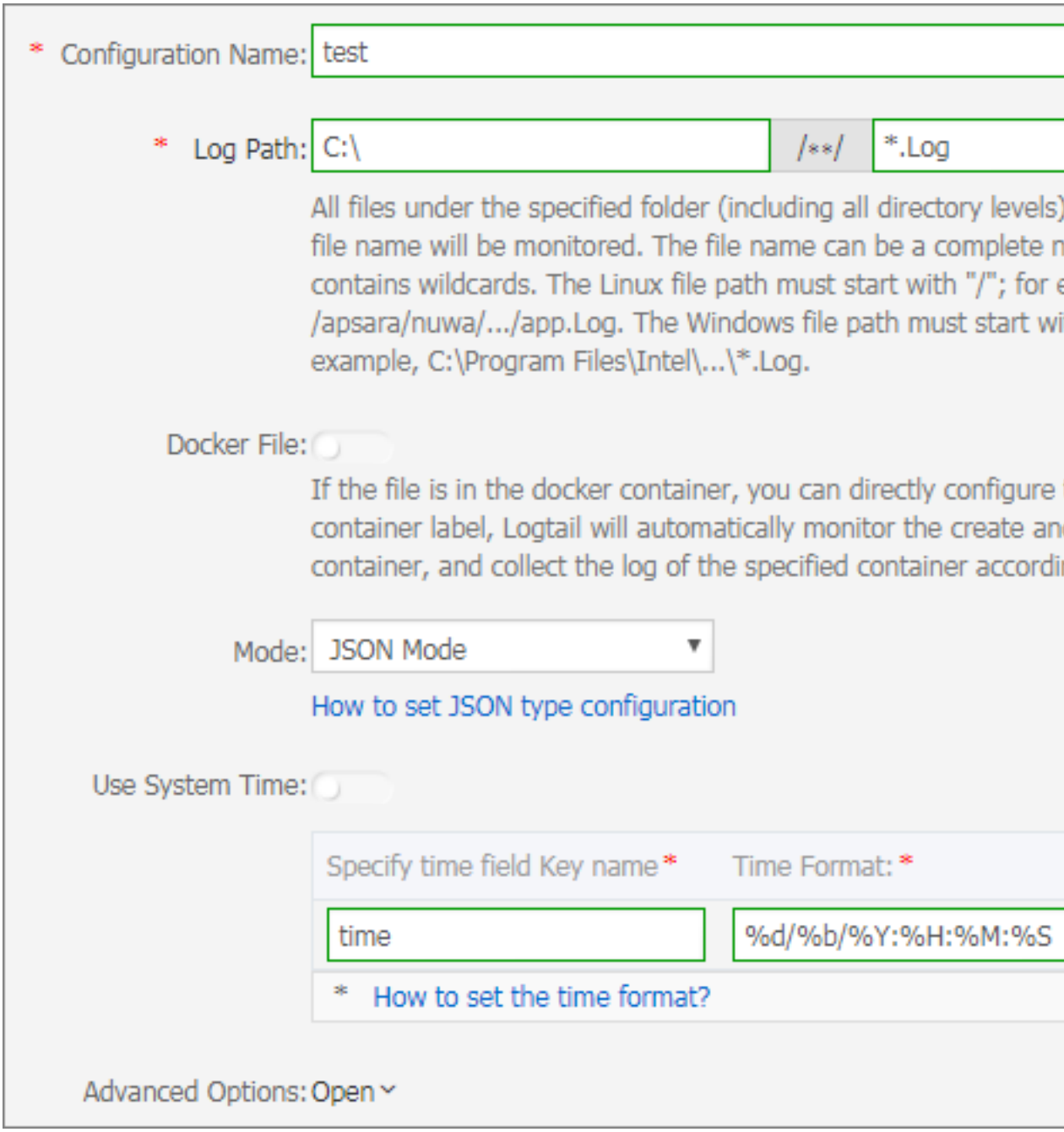
   - Enable **Use System Time function**

     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 **the 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 Logtail date format.

Figure 2-1: JSON logs



2.5.9 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 instantiation 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\BuildLiteBehavior [ RunTime:0.000014s ]
INFO: [ app_init ] --END-- [ RunTime:0.000091s ]
Info: [app_begin] -- start --
INFO: Run Behavior\ReadHtmlCacheBehavior [ RunTime:0.000038s ]
INFO: [ app_begin ] --END-- [ RunTime:0.000076s ]
INFO: [ view_parse ] --START--
INFO: Run Behavior\ParseTemplateBehavior [ RunTime:0.000068s ]
INFO: [ view_parse ] --END-- [ RunTime:0.000104s ]
INFO: [ view_filter ] --START--
INFO: Run Behavior\WriteHtmlCacheBehavior [ RunTime:0.000032s ]
INFO: [ view_filter ] --END-- [ RunTime:0.000062s ]
INFO: [ app_end ] --START--
INFO: Run Behavior\ShowPageTraceBehavior [ RunTime:0.000032s ]
INFO: [ app_end ] --END-- [ RunTime:0.000070s ]
ERR: D method instantiation does not find the model class
```

Configure Logtail to collect ThinkPHP logs

For the complete process of collecting ThinkPHP logs by using Logtail, see *#unique_49*. 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
```

2.5.10 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.

Collection configuration

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_position => "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_timestamp} %{
IPORHOST:site} %{WORD:method} %{URIPATH:page} %{NOTSPACE:querystring}
 %{NUMBER:port} %{NOTSPACE:username} %{IPORHOST:clienthost} %{NOTSPACE
:useragent} %{NUMBER:response} %{NUMBER:subresponse} %{NUMBER:scstatus
} %{NUMBER:time_taken}"]

    date {
    match => [ "log_timestamp", "YYYY-MM-dd HH:mm:ss" ]
      timezone => "Etc/UTC"

  useragent {
    source=> "useragent"
```

```
    prefix=> "browser"

  mutate {
    remove_field => [ "log_timestamp"]



output {
  if [type] == "iis_log_1" {
  logservice {
        codec => "json"
        endpoint => "***"
        project => "***"
        logstore => "***"
        topic => ""
        source => ""
        access_key_id => "***"
        access_key_secret => "***"
        max_send_retry => 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.

2.5.11 Use Logstash to collect IIS logs

View IIS log configurations, select the W3C format (default field setting), and save the format to put it into effect. 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.

Upload the system time as the 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_position => "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_retry => 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 *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 *## Logstash # 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_withtime
.log
```

- **Collection configuration**

```
input {
  file {
    type => "csv_log_2"
    path => ["C:/test/csv_withtime/*.log"]
    start_position => "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_retry => 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* 填写文件路径时请使用 UNIX *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, *C:/test/multiline/*.log* . Otherwise, fuzzy match is not supported.

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 *## Logstash # Windows Service* as a Windows service.

2.5.12 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.ArithmeticException: / 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_position => "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_retry => 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 *## Logstash # Windows Service*.

2.5.13 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 *Web Tracking* of Log Service to collect Unity3D logs conveniently. This document collect the Unity `Unity Debug.Log` . use the Web Tracking function to collect Unity logs to Log Service.

Procedure

1. Activate the Web Tracking function

   For more information, see *Web Tracking*.

2. Register Unity3D LogHandler

   Create a C# file `LogOutputHandler.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 ####.

```
using UnityEngine;
using System.Collections;
public class LogOutputHandler : MonoBehaviour

    //Register the HandleLog function on scene start to fire on
debug.log events
    public void OnEnable()

        Application.logMessageReceived += HandleLog;

    //Remove callback when object goes out of scope
    public void OnDisable()

        Application.logMessageReceived -= HandleLog;

    string project = "your project name";
    string logstore = "your logstore name";
    string serviceAddr = "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_Trace=" + WWW.EscapeURL(stackTrace);
        parameters += "&";
        //Add any User, Game, or Device MetaData that would be
useful to finding issues later
        parameters += "Device_Model=" + WWW.EscapeURL(SystemInfo.
deviceModel);
        string url = "http://" + project + "." + serviceAddr + "/
logstores/" + logstore + "/track? APIVersion=0.6.0&" + parameters;
        StartCoroutine(SendData(url));

    public IEnumerator 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 UnityEngine;
```

```
using System.Collections;
public class LogglyTest : MonoBehaviour {
    void Start () {
        Debug.Log ("Hello world");
```

4. 注册 Unity3D LogHandler。

After completing the preceding steps, run the Unity program. Then, you can preview your sent logs in the Log Service console. For how to preview logs, see Preview logs.

The preceding example provides the methods for collecting logs such as *`Debug.Log`* , *`Debug.LogError`* , and *`Debug.LogException`* . 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.

3 Logtail collection

3.2 Install

3.2.1 Linux

Supported systems

Logtail supports the Linux x86-64 (64 bit) servers in the following releases:

- Aliyun Linux
- Ubuntu
- Debian
- Centos
- Opensuse

Install Logtail

Install Logtail in overwrite mode. If you have installed Logtail before, the installer uninstalls the Logtail and deletes the `/usr/local/ilogtail` directory before installing Logtail. By default, Logtail is started after the installation and at startup.

Download the installer based on the network environment of your machine and the region of Log Service. Select different parameters for installation.

Follow the steps in Installation method of this document to install Logtail. If failed, open a ticket.

Installation method

To install Logtail, download and run the installation script. You must select the **installation parameters** based on the regions and network types.

Installation parameters

Note:
To install Logtail in Docker or Kubernetes, the `${your_region_name}` is the parameter in the following table. Copy the corresponding installation statement directly.

The installation parameters for different regions and network types are as follows (we recommend that you copy the corresponding installation statement directly).

Region	Classic network	VPC	Internet (self-built IDCs)
China North 2 (Beijing)	cn-beijing	cn-beijing_vpc	cn-beijing_internet
China North 1 (Qingdao)	cn-qingdao	cn-qingdao_vpc	cn-qingdao_internet
China East 1 (Hangzhou)	cn-hangzhou	cn-hangzhou_vpc	cn-hangzhou_internet
China East 2 (Shanghai)	cn-shanghai	cn-shanghai_vpc	cn-shanghai_internet
China South 1 (Shenzhen)	cn-shenzhen	cn-shenzhen_vpc	cn-shenzhen_internet
China North 3 (Zhangjiakou)	cn-zhangjiakou	cn-zhangjiakou_vpc	cn-zhangjiakou_internet
China North 5 (Huhehaote)	cn-huhehaote	cn-huhehaote	cn-huhehaote_internet
China (Chengdu)	None	cn-chengdu	cn-chengdu_internet
Hong Kong (China)	cn-hongkong	cn-hongkong_vpc	cn-hongkong_internet
US West 1 (Silicon Valley)	us-west-1	us-west-1_vpc	us-west-1_internet
US East 1 (Virginia)	None	us-east-1	us-east-1_internet
Asia Pacific SE 1 (Singapore)	ap-southeast-1	ap-southeast-1_vpc	ap-southeast-1_internet
Asia Pacific SE 2 (Sydney)	ap-southeast-2	ap-southeast-2_vpc	ap-southeast-2_internet
Asia Pacific SE 3 (Kuala Lumpur)	ap-southeast-3	ap-southeast-3	ap-southeast-3_internet
Asia Pacific SE 5 (Jakarta)	None	ap-southeast-5	ap-southeast-5_internet
Asia Pacific SOU 1 (Mumbai)	None	ap-south-1	ap-south-1_internet
Asia Pacific NE 1 (Japan)	ap-northeast-1	ap-northeast-1_vpc	ap-northeast-1_internet
EU Central 1 (Frankfurt)	eu-central-1	eu-central-1_vpc	eu-central-1_internet
Middle East 1 (Dubai)	me-east-1	me-east-1_vpc	me-east-1_internet

Region	Classic network	VPC	Internet (self-built IDCs)
China East 1 (Hangzhou) (financial cloud)	None	CN-Hangzhou-finance	None
China East 2 (Shanghai) (financial cloud)	None	cn-shanghai-finance	None
China South 1 (Shenzhen) (financial cloud)	None	cn-shenzhen-finance	None

ECS instances of classic network

Data is written to Log Service by means of the Alibaba Cloud intranet without consuming Internet bandwidth, which is applicable to Alibaba Cloud Elastic Compute Service (ECS) instances.

- Note: By default, newly created ECS instances in new regions (such as overseas regions) are of Virtual Private Cloud (VPC). See ECS instances of VPC in this document.

- China North 2 (Beijing)

```
wget http://logtail-release-bj.oss-cn-beijing-internal.aliyuncs.com
/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-beijing
```

- China North 1 (Qingdao)

```
Wget maid logtail. Sh; chmod 755 logtail. Sh; SH logtail. sh install
 CN-Qingdao
```

- China EasEast China 1 (Hangzhou)

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

- China East 2 (Shanghai)

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

- China South 1 (Shenzhen)

```
wget http://logtail-release-sz.oss-cn-shenzhen-internal.aliyuncs.com
/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-shenzhen
```

- China North 3 (Zhangjianchou)

```
wget http://logtail-release-zjk.oss-cn-zhangjiakou-internal.aliyuncs
.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh
logtail.sh install cn-zhangjiakou
```

- China North 5 (Huhhote)

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

- Hong Kong (China)

```
wget http://logtail-release-hk.oss-cn-hongkong-internal.aliyuncs.com
/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-hongkong
```

- US West 1 (Silicon Valley)

```
wget http://logtail-release-us-west-1.oss-us-west-1-internal.
aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh
; sh logtail.sh install us-west-1
```

- Asia Pacific SE 1 (Singapore)

```
wget http://logtail-release-ap-southeast-1.oss-ap-southeast-1-
internal.aliyuncs.com/linux64/logtail.sh; chmod 755 logtail.sh; sh
logtail.sh install ap-southeast-1
```

- Asia Pacific SE 2 (Sydney)

- China North 2 (Beijing)

```
wget http://logtail-release-bj.vpc100-oss-cn-beijing.aliyuncs.com/
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-beijing_vpc
```

- China North 1 (Qingdao)

```
wget http://logtail-release-qd.vpc100-oss-cn-qingdao.aliyuncs.com/
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-qingdao_vpc
```

- China East 1 (Hangzhou)

```
wget http://logtail-release.vpc100-oss-cn-hangzhou.aliyuncs.com/
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-hangzhou_vpc
```

- China East 2. Shanghai)

```
wget http://logtail-release-sh.vpc100-oss-cn-shanghai.aliyuncs.com/
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-shanghai_vpc
```

- China South 1 1 (Shenzhen)

```
wget http://logtail-release-sz.vpc100-oss-cn-shenzhen.aliyuncs.com/
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-shenzhen_vpc
```

- China North 3 (Zhangjianchou)

```
wget http://logtail-release-zjk.oss-cn-zhangjiakou-internal.aliyuncs
.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh
logtail.sh install cn-zhangjiakou_vpc
```

- China North 5 (Huhhote)

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

- China (Chengdu)

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

- Hong Kong (China)

```
wget http://logtail-release-hk.vpc100-oss-cn-hongkong.aliyuncs.com/
linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.
sh install cn-hongkong_vpc
```

- US West 1 (Silicon Valley)

```
wget http://logtail-release-us-west-1.vpc100-oss-us-west-1.aliyuncs
.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh
logtail.sh install us-west-1_vpc
```

- US East 1 (Virginia)

Note:

Log Service cannot obtain the owner information of non-Alibaba Cloud machines. Therefore, you must manually configure the user identification after installing Logtail. ###*ECS*#######For more information, see Configure a user identity for non-Alibaba Cloud ECS instances. Otherwise, Logtail has abnormal heartbeat and cannot collect logs..

- China North 2 (Beijing)

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

- China North 1 (Qingdao)

```
Wget glaslogtail. Sh; chmod 755 logtail. Sh; SH logtail. sh install
```

- China East 1 1 (Hangzhou)

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

- China East 2. Shanghai)

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

- China South 1 1 (Shenzhen)

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

- China North 3 (Zhangjianchou)

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

- China North 5 (Huhhote)

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

- China (Chengdu)

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

- Hong Kong (China)

```
wget http://logtail-release-hk.oss-cn-hongkong.aliyuncs.com/linux64
/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.sh
install cn-hongkong_internet
```

- US West 1 (Silicon Valley)

```
wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64
/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.sh
install us-west-1_internet
```

- US East 1 (Virginia)

```
wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64
```

```
   "UUID" : "0DF18E97-0F2D-486F-B77F-*********",
   "hostname" : "david*******",
   "instance_id" : "F4FAFADA-F1D7-11E7-846C-00163E30349E_*********
_1515129548",
   "ip" : "**********",
   "logtail_version" : "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_time" : "2018-01-05 13:19:08"
```

Update Logtail

The procedure of updating Logtail is the same as that of installing Logtail. When you update the Logtail, the Logtail is automatically uninstalled first and then the latest version of Logtail is installed .

Manually start and stop Logtail

- Start Logtail

  Run the following command as an administrator:

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

- Stop Logtail

  Run the following command as an administrator:

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

Uninstall Logtail

Download the installer **logtail.sh**. For more information, see *Install Logtail* in this document. Run the following command as an administrator in shell mode:

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

3.2.2 Windows

Supported systems

Logtail supports the following systems:

- Windows 7 (Client) 32bit
- Windows 7 (Client) 64bit
- Windows Server 2003 32bit
- Windows Server 2003 64bit

- Windows Server 2008 32bit
- Windows Server 2008 64bit
- Windows Server 2012 64bit

Installing Logtail

1. Download the installation package

   You can download the installation package at http://logtail-release.oss-cn-hangzhou.aliyuncs.com/win/logtail_installer.zip.

2. Install Logtail based on the network environment of your machine and the region of Log Service.

   Extract `logtail.zip` to the current directory and use Windows PowerShell or cmd.exe to enter the `logtail_installer` directory.

Region of Log Service	Network environment of your machine	Installation command
China North 1 (Qingdao)	Elastic Compute Service (ECS) instances of classic network	.\logtail_installer.exe install cn_qingdao
	ECS instances of VPC	.\logtail_installer.exe install cn_qingdao_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install cn_qingdao_internet
North China 2 (Beijing)	ECS instances of classic network	.\logtail_installer.exe install cn_beijing
	ECS instances of VPC	.\logtail_installer.exe install cn_beijing_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install cn_beijing_internet
North China 3 (zhangjianchou)	ECS instances of classic network	.\logtail_installer.exe install cn -zhangjiakou
	ECS instances of VPC	.\logtail_installer.exe install cn -zhangjiakou_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install cn -zhangjiakou_internet

Region of Log Service	Network environment of your machine	Installation command
China North 5 (Huhehaote)	ECS instances of VPC	.\logtail_installer.exe install cn-huhehaote
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install Cn-huhehaote_internet
China East 1 (Hangzhou)	ECS instances of classic network	.\logtail_installer.exe install cn_hangzhou
	ECS instances of VPC	.\logtail_installer.exe install cn_hangzhou_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install cn_hangzhou_internet
	ECS instances of AntCloud	.\logtail_installer.exe install cn_hangzhou_finance
China East 2 (Shanghai)	ECS instances of classic network	.\logtail_installer.exe install cn_shanghai
	ECS instances of VPC	.\logtail_installer.exe install cn_shanghai_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install Cn_shanghai_internet
	ECS instances of AntCloud	.\logtail_installer.exe install cn-shanghai-finance
China South 1 (Shenzhen)	ECS instances of classic network	.\logtail_installer.exe install cn_shenzhen
	ECS instances of VPC	.\logtail_installer.exe install cn_shenzhen_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install cn_shenzhen_internet
	ECS instances of AntCloud	.\logtail_installer.exe install cn_shenzhen_finance
cn-hongkong	ECS instances of classic network	.\logtail_installer.exe install cn-hongkong
	ECS instances of VPC	.\logtail_installer.exe install cn-hongkong_vpc

Region of Log Service	Network environment of your machine	Installation command
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install cn-hongkong_internet
US West 1 (Silicon Valley)	ECS instances of classic network	.\logtail_installer.exe install us-west-1
	ECS instances of VPC	.\logtail_installer.exe install us-west-1_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install us-west-1_internet
US East 1 (Virginia)	ECS instances of VPC	.\logtail_installer.exe install us-east-1
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install us-east-1_internet
Asia Pacific SE 1 (Singapore)	ECS instances of classic network	.\logtail_installer.exe install ap-southeast-1
	ECS instances of VPC	.\logtail_installer.exe install ap-southeast-1_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install ap-southeast-1_internet
Asia Pacific SE 2 (Sydney)	ECS instances of classic network	.\logtail_installer.exe install ap-southeast-2
	ECS instances of VPC	.\logtail_installer.exe install ap-southeast-2_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install ap-southeast-2_internet
Asia Pacific SE 3 (Kuala Lumpur)	ECS instances of VPC	.\logtail_installer.exe install ap-southeast-3
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install ap-southeast-3_internet
Asia Pacific NE 1 (Japan)	ECS instances of classic network	.\logtail_installer.exe install ap-northeast-1
	ECS instances of VPC	.\logtail_installer.exe install ap-northeast-1_vpc

Region of Log Service	Network environment of your machine	Installation command
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install ap-northeast-1_internet
EU Central 1 (Frankfurt)	ECS instances of classic network	.\logtail_installer.exe install eu-central-1
	ECS instances of VPC	.\logtail_installer.exe install eu-central-1_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install eu-central-1_internet
Middle East 1 (Dubai)	ECS instances of classic network	.\logtail_installer.exe install me-east-1
	ECS instances of VPC	.\logtail_installer.exe install me-east-1_vpc
	Internet (self-built IDCs or other cloud hosts)	.\logtail_installer.exe install me-east-1_internet

Note:
Log Service cannot obtain the owner information of non-Alibaba Cloud machines. Therefore, you must manually configure the user identification after installing Logtail when Logtail is used by self-built IDCs or other cloud hosts. *###ECS########*For more information, see Configure user identification for non-Alibaba Cloud ECS instances. Otherwise, Logtail will have abnormal heartbeats and cannot collect logs.

Uninstall Logtail

Use Windows PowerShell or cmd.exe to enter the `logtail_installer` directory and run the following command:

```
.\logtail_installer.exe uninstall
```

3.2.5 Configure startup parameters

This document describes the Logtail startup configuration parameters. You can configure the startup parameters by following this document 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.
- The CPU usage is high because the volume of log data is large and the traffic sent to Log Service is heavy.
- Therefore, the memory usage might be high if a large number of log files are to be collected.
- Syslog/TCP data streams are to be collected.

Startup configuration

- File path

```
/usr/local/ilogtail/ilogtail_config.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_thread_count" : 1,
"send_request_concurrency" : 4,
"streamlog_open" : false,
"streamlog_pool_size_in_mb" : 50,
"streamlog_rcv_size_each_call" : 1024,
"streamlog_formats":[],
"streamlog_tcp_port" : 11111,
"buffer_file_num" : 25,
"buffer_file_size" : 20971520,
"buffer_file_path" : "",
```

Common configuration parameters

Parameter name	Value	Description
cpu_usage_limit	The CPU usage threshold . Double type. Calculated per core.	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. In most cases, the single-core processing capability is about 24 MB/s in simple mode and about 12 MB/s in full mode. .

Parameter name	Value	Description
mem_usage_limit	The usage threshold of resident memory. Int type. Measured in MBs.	For example, the value 100 indicates the memory usage of Logtail is limited to 100 MB. Logtail restarts automatically when the threshold is exceeded. To collect more than 1,000 distinct files, properly increase the threshold value.
max_bytes_per_sec	The traffic limit on the raw data sent by Logtail. Int type. Measured in bytes per second.	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.	The default value is 1, which generally supports a write speed of 24 MB/s in simple mode and 12 MB/s in full mode. Increase the threshold value only when necessary. You do not need to adjust this threshold by default, and you only raise the threshold if necessary.
send_request_concurrency	By default, Logtail sends data packets asynchronously.	You can set a larger asynchronous concurrency value if the write TPS is large. By default, four asynchronous concurrencies are available.  **Note:** You can calculate the concurrency quantity based on the condition that one concurrency supports 0.5–1 MB/s network throughout. The actual quantity varies with network delay.
streamlog_open	Whether or not to enable the syslog reception function. Bool type.	false indicates to disable the function. true indicates to enable the function. *Syslog-####*.
streamlog_pool_size_in_mb	Size of the cache storing received syslog data. Measured in MBs.	The size of the memory pool that syslog uses to receive logs. Logtail requests a specified size of memory at one time when started. Configure the size according to the memory size of your machine and your actual requirements

Parameter name	Value	Description
streamlog_rcv_size_each_call	Size of the buffer Logtail uses when calling the Linux socket rcv interface. Measured in bytes.	You can increase the value if the syslog traffic is heavy. The recommended value range is 1024–8192.
streamlog_formats	The method of parsing received syslogs.	*Syslog-####*.
streamlog_tcp_addr	The binding address that Logtail uses to receive syslogs. The default value is 0.0.0.0.	*Syslog-####*.
streamlog_tcp_port	The TCP port that Logtail uses to receive syslogs.	The default value is 11111.
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.	The default value is 25 for public cloud users.
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.	The default value is 20,971,520 (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 default value is null, indicating the cached files are stored in the Logtail installation directory (*/usr/local/ilogtail*).

Parameter name	Value	Description
	the old cache directory to the new cache directory so that Logtail can read the cached files and delete them after sending logs.	
bind_interface	The name of the network card that is bound to the local machine, such as eth1 (only Linux versions are supported).	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.
check_point_filename	The full path stored by the checkpoint file, which is used to customize the checkpoint storage location of Logtail.	The default value is /tmp/logtail_check_point . **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.

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. Unused configuration parameters (for example, parameters related to the collection of . syslog data streams) 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
```

3.3 Data Source

3.3.3 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 (Description)
%a	The abbreviation of a day in a week.	Example: Fri
%A	The name of a day in a week.	Example: Friday
%b	The abbreviation of a month.	Example: Jan
%B	The name of a month.	Example: January
%d	The day of the month in decimal format [01,31].	Example: 07, 31
%h	The abbreviation of a month. Same as %b.	Example: Jan
%H	The hour in 24-hour format.	Example: 22
%I	The hour in 12-hour format.	Example: 11
%m	The month in decimal format.	Example: 08
%M	The minute in decimal format [00,59].	Example: 59

Format	Meaning	Example (Description)
%n	A line break.	A line break
%p	AM or PM locally.	例如：AM/PM
%r	Time in 12-hour format, which is equivalent to %I:%M:%S %p.	Example: 11: 59: 59 AM
%R	Time expressed in hour and minute, which is equivalent to %H:%M.	Example: 23: 59
%S	The second in decimal format [00,59].	Example: 59
%t	Tab character.	Tab character
%y	The year without century in decimal format [00,99].	Example: 04,98
%Y	The year in decimal format.	For example: 2004,1998
%z	The time zone or its abbreviation.	Example:-0x7, + 0800
%C	The century in decimal format [00-99].	Example: 16
%e	The day of the month in decimal format [1,31]. A single digit is preceded by a space.	Example: 7, 31
%j	The day of the year in decimal format [001,366].	For example 365
%u	The day of the week in decimal format [1,7]. 1 represents Monday.	Example: 2
%U	The week number of the year (Sunday as the first day of the week) [00,53].	Example: 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	Example: 24

Format	Meaning	Example (Description)
	is considered as the first week of the year.	
%w	The day of the week in decimal format [0,6]. 0 represents Sunday.	Example: 5
%W	The week number of the year (Monday as the first day of the week) [00,53].	Example: 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.
%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 long date and short date, we recommend that you use the preceding supported formats for more precise expression.
%s	UNIX timestamp.	Example: 1476187251

3.3.5 Syslog

By using Logtail, you can configure TCP ports locally to receive syslog data forwarded by syslog agents by means of the TCP protocol. Logtail parses the received data and forwards it to LogHub.

Prerequisite

You must install Logtail before using it to collect logs. Logtail supports Windows and Linux operating systems. For the installation methods, see *Linux* and *Windows*.

Step 1. Create a Logtail syslog configuration in the Log Service console

1. Log on to the Log Service console, and click the target project to enter the **Logstore List**.
2. On the Logstore List page, click the **Data Import Wizard** icon at the right of the Logstore.
3. 3.Select the data source type.

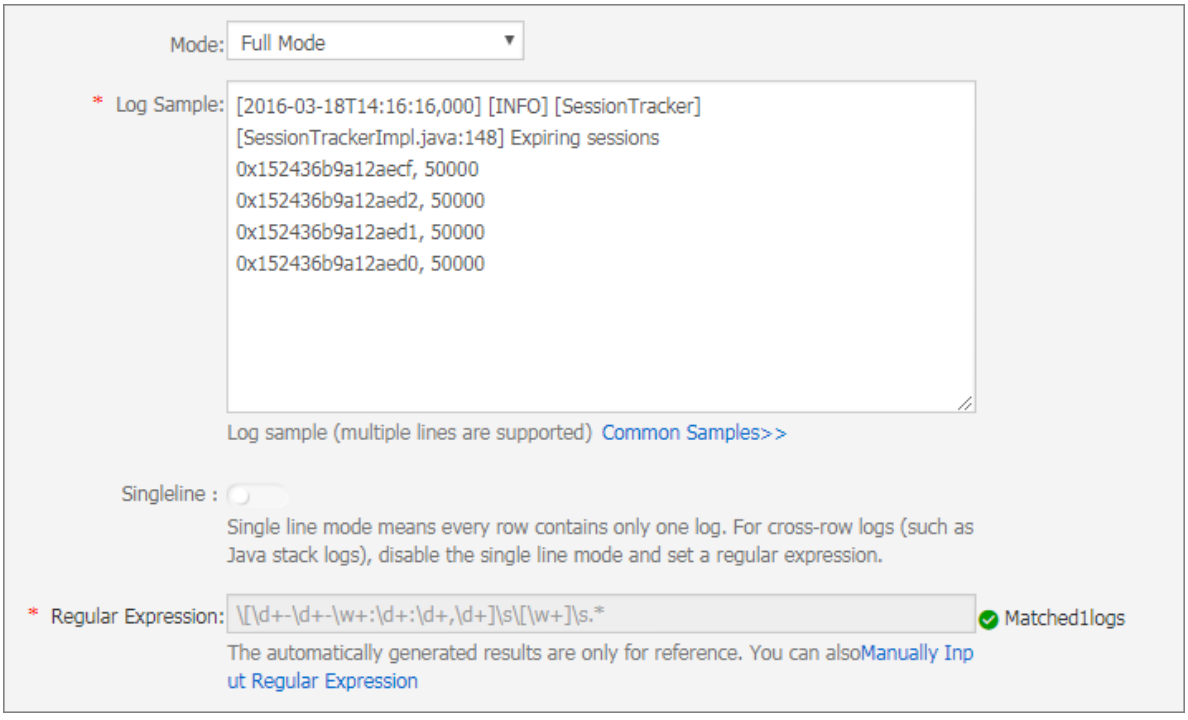
   Select **syslog** in **Other Sources** and click **Next**.
4. Specify the **`Configuration Name`**.

   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 **Tag Settings**.

   For more information, see *Syslog-####*.

   Figure 3-1: Set tag



6. Set **Advanced Options** as needed.

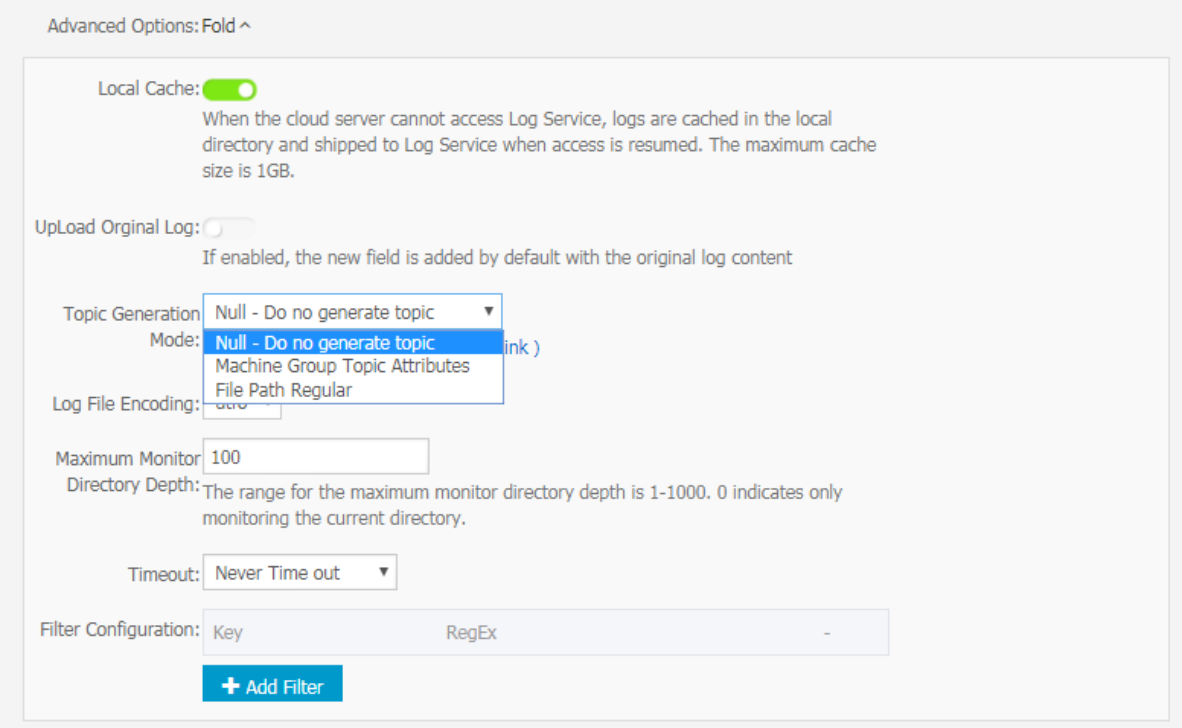
   Select whether to enable **Local Cache**. When the Log Service is unavailable, logs can be cached in the local directory of the machine, and continue to be sent to Log Service after the service recovery. By default, this function is enabled, and at most 1 GB logs can be cached.

7. Select the machine group and click

   Apply to Machine Group to apply the configuration to the selected machine group.

   If you have not created a machine group, you must create one first. For how to create a machine group, see #####.

Figure 3-2: Applied to the Machine group



Step 2. Configure Logtail to bring the protocol into effect

Find the `ilogtail_config.json` file in the Logtail installation directory on the machine. Generally, it is in the `/usr/local/ilogtail/` directory. Modify the syslog configuration as needed.

1. Confirm that the syslog function is enabled.

true indicates the function is enabled. false indicates the function is disabled.

```
"streamlog_open" : true
```

2. Specify the size of the memory pool that syslog uses to receive logs Logtail requests a specified size of memory at one time when started. Configure the size (in MB) according to the memory size of your machine and your actual requirements.

```
"streamlog_pool_size_in_mb" : 50
```

3. Specify the buffer size (in bytes). You must specify the size of the buffer that Logtail uses when calling the socket io rcv interface.

```
"streamlog_rcv_size_each_call" : 1024
```

4. Specify the syslog log format.

```
"streamlog_formats":[]
```

5. 5.Specify the TCP binding address and port. You must specify the TCP binding address and port that Logtail uses to receive syslog data. By default, the binding address is 0.0.0.0 and the binding port is 11111.

```
"streamlog_tcp_addr" : "0.0.0.0",
"streamlog_tcp_port" : 11111
```

6. After the configuration is complete, restart Logtail. To restart Logtail, run the following commands to stop the Logtail client and then start it again.

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

Step 3. Install rsyslog and modify its configurations

Skip this step if you have installed rsyslog on the machine.

1. Install rsyslog.

   For the installation method, see:

   - *Ubuntu installation method*
   - *Debian installation method*
   - *RHEL/CENTOS installation method*

2. Modify configurations.

   In `/etc/rsyslog.conf` , modify the configurations as needed, for example:

```
$WorkDirectory /var/spool/rsyslog # where to place spool files
```

```
$ActionQueueFileName fwdRule1 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g # 1gb space limit (use as much as
possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList # run asynchronously
$ActionResumeRetryCount -1 # infinite retries if host is down
# Defines the fields of log data
$template ALI_LOG_FMT,"0.1 sys_tag %timegenerated:::date-unixtimest
amp% %fromhost-ip% %hostname% %pri-text% %protocol-version% %app-
name% %procid% %msgid% %msg:::drop-last-lf%\n"
* @@10.101.166.173:11111;ALI_LOG_FMT
```

Note:

In the template `ALI_LOG_FMT` , the value of the second field is `sys_tag` . This value must be the same as the one entered in step 1. This configuration indicates that all the (*.*) syslog data received`\*.\*` by this machine is formatted according to the `ALI_LOG_FMT` template, and forwarded to 10.101.166.173:11111 by using the TCP protocol. The machine 10.101.166.173 must be in the machine group selected in step 1 and configured according to step 2.

3. Start rsyslog.

```
sudo /etc/init.d/rsyslog restart
```

Before starting rsyslog, check whether another syslog agent is installed on the machine, such as syslogd, sysklogd, or syslog-ng. If yes, stop that syslog agent.

After completing the preceding three steps, you can collect syslog data on the machine to Log Service.

Further information

For more information about syslog collection, and how to format syslog data, see *Syslog-####*.

3.3.8 Container 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.

Function features

- Supports collecting stdout and stderr.
- Supports using labels to specify containers to be collected.
- Supports using labels to exclude specific containers.
- Supports environments to to specify containers to be collected.

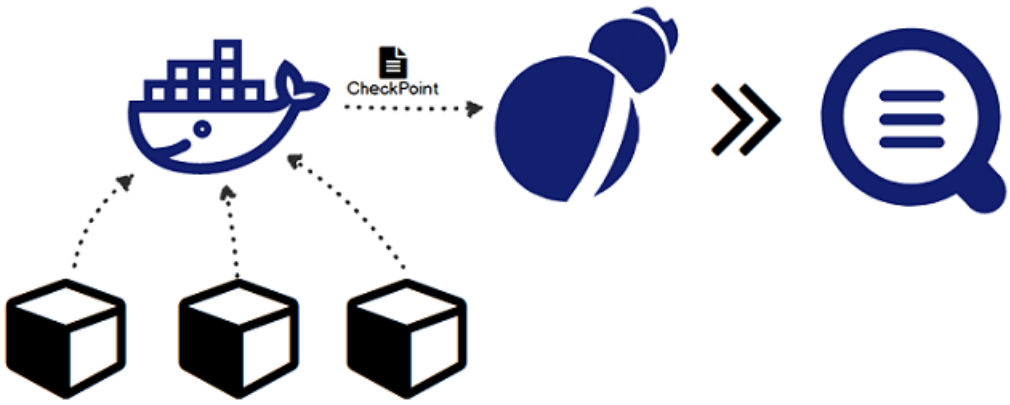
- Supports environments to exclude specific containers.
- Supports multiline logs (such as Java stack logs).
- Supports automatic tagging for container data.
- Supports automatic tagging for Kubernetes containers.

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.

Logtail periodically saves the collected point information to the checkpoint file when collecting the standard output of the container. If Logtail is restarted after being stopped, logs are collected from the last saved point.

Figure 3-3: 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 *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**. TTo make sure that a log made up of multiple lines is not split up due to output delay, the last collected multiline log is 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.

- **Stop collection policy**. 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**. By default, a collection configuration is in the same context. To set different contexts for each type of container, create a collection configuration for each type.
- **Data processing**. The default field of collected data is `content`, which supports common processing configurations. To configure one or more collection methods, see #####.
- **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. Deploy and configure the Logtail container

- **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. Set the collection configuration in Log Service

1. On the **Logstore List** page, click the **Data Import Wizard** icon 1 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.For more information on the descriptions of configuration items, see *Configuration item description* in this document.

```
"inputs": [
```

```
        "type": "service_docker_stdout",
        "detail": {
            "Stdout": true,
            "Stderr": true,
            "IncludeLabel": {
                "io.kubernetes.container.name": "nginx"

            "ExcludeLabel": {
                "io.kubernetes.container.name": "nginx-ingress-
controller"

            "IncludeEnv": {
                "NGINX_SERVICE_PORT": "80"

            "ExcludeLabel": {
                "POD_NAMESPACE": "kube-system"
            }
```

4. 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.

Configuration item description

The input source type is `service_docker_stdout`

> **Note:**
> Logtail supports processing and then uploading the collected data. For more information on the processing methods, see ##-######.

Configuration item	Type	Required	Description
IncludeLabel	The mapping type, where key and value are both strings.	Yes	Empty by default. If empty, all containers are collected. If the key is not empty but the value is empty, all the containers whose label includes this key are collected. **Note:** 1. Key-value pairs have an OR

Configuration item	Type	Required	Description
			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.
ExcludeLabel	The mapping type, where key and value are both strings.	No	Empty by default. If empty, no containers are excluded. If the key is not empty but the value is empty, all the containers whose label 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 label includes any of the key-value pairs. 2. Here the label is the label information in Docker inspect.
IncludeEnv	The mapping type, where key and value are both strings.	No	Empty by default. If empty, all containers are collected. If the key is not empty but

Configuration item	Type	Required	Description
			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. Here the environment is the environmen t information configured in the container startup.
ExcludeEnv	The mapping type, where key and value are both strings.	No	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

Configuration item	Type	Required	Description
			environment includes any of the key-value pairs. 2. Here the environment is the environment information configured in the container startup.
Stdout	bool	No	True by default. When false, stdout data is not collected.
Stderr	bool	No	True by default. When false, stderr data is not collected.
BeginLineRegex	string	No	Empty by default. If not empty, it is the regular expression to match with the beginning of a row. If this regular expression matches with a row, the row is treated as a new log. Otherwise, this row of data is connected to the previous log.
BeginLineTimeoutMs	int	No	Timeout (in milliseconds) for matching with the beginning of a row. The default value is 3,000. If no new logs appear within three seconds, the last log is output.
BeginLineCheckLength	int	No	The length (in bytes) of the beginning of a row used to match with the regular

Configuration item	Type	Required	Description
			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	No	The maximum length (in bytes) of a log. The default value is 512 *1024. If the length of a log exceeds the configured value, the log is uploaded directly without searching for the matched beginning of a row.

Default field

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_name_`	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 Environment configuration**

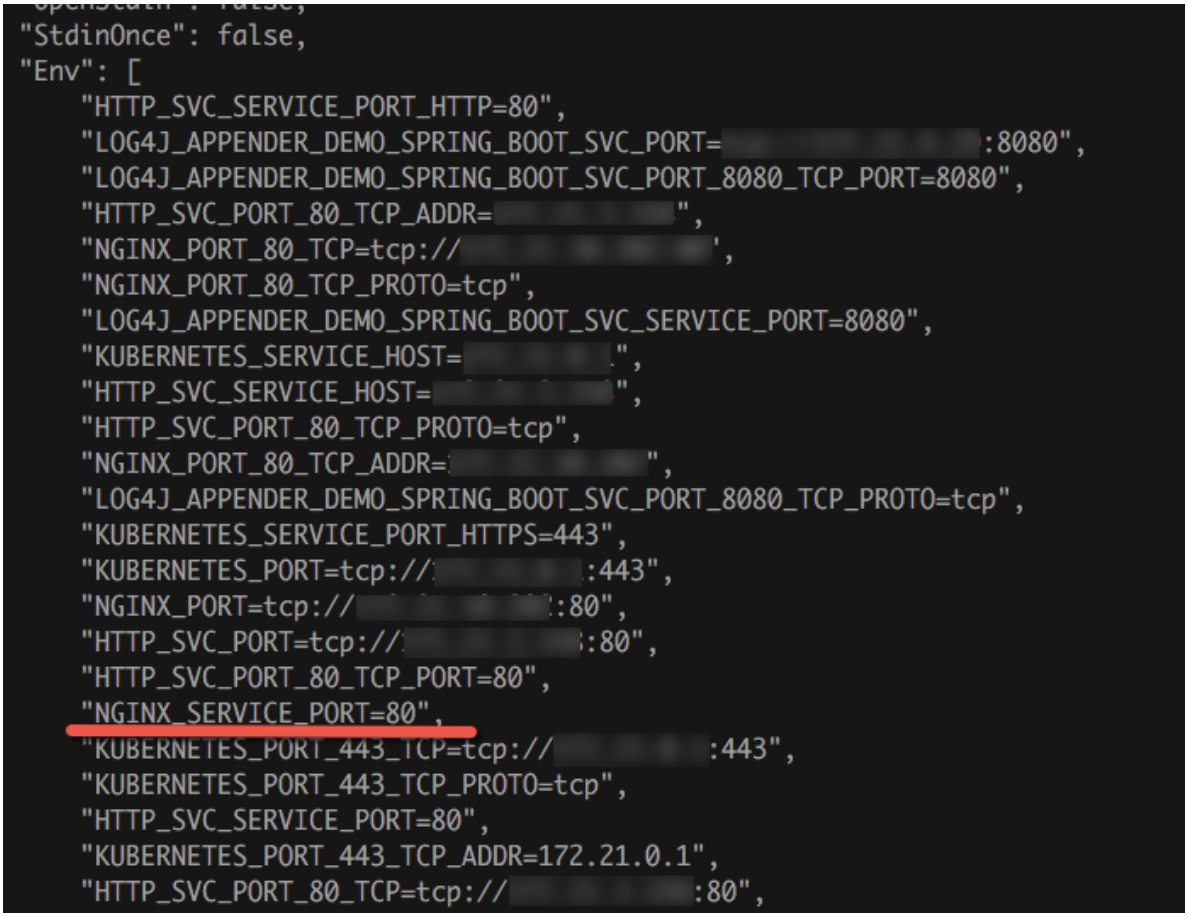
  Collect the stdout logs and stderr logs of containers whose environment is `NGINX_PORT_80_TCP_PORT=80`, and is not `POD_NAMESPACE=kube-system`:

  Note:

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

Figure 3-4: Example of Environment Configuration



Collection configuration:

```
    "inputs": [

            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeEnv": {
                    "NGINX_PORT_80_TCP_PORT": "80"

                "ExcludeEnv": {
                    "POD_NAMESPACE": "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:

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

Figure 3-5: Label configuration example


```
    "inputs": [

            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeLabel": {
                    "io.kubernetes.container.name": "nginx"

                "Excluded Abel ":{
                    "type": "pre"
```

Collection configuration of multiline logs

Multiline log collection is particularly important for the collection of Java exception stack output. Here we introduce a standard collection configuration of Java standard output logs.

- **Log example:**

```
2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4
] c.g.s.web.controller.DemoController : service start
2018-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-
exec-4] c.g.s.web.controller.DemoController : java.lang.NullPointe
rException
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(
ApplicationFilterChain.java:193)
```

```
at org.apache.catalina.core.ApplicationFilterChain.doFilter(
ApplicationFilterChain.java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWr
apperValve.java:199)
at org.apache.catalina.core.StandardContextValve.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.DemoController : 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": {
      "BeginLineCheckLength": 10,
      "BeginLineRegex": "\\d+-\\d+-\\d+.
      "IncludeLabel": {
        "app": "monitor"


    "type": "service_docker_stdout"
```

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": {
      "BeginLineCheckLength": 10,
      "BeginLineRegex": "\\d+-\\d+-\\d+.
      "IncludeLabel": {
        "app": "monitor"


    "type": "service_docker_stdout"
```

```
"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,
            "NoMatchError": 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.DemoController : service start done` is as follows:

```
__tag__:__hostname__:logtail-dfgef
_container_name_:monitor
_image_name_:registry.cn-hangzhou.aliyuncs.xxxxxxxxxxxxxxx
_namespace_:default
_pod_name_:monitor-6f54bd5d74-rtzc7
_pod_uid_:7f012b72-04c7-11e8-84aa-00163f00c369
_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. demcontroller
message:service start done
```

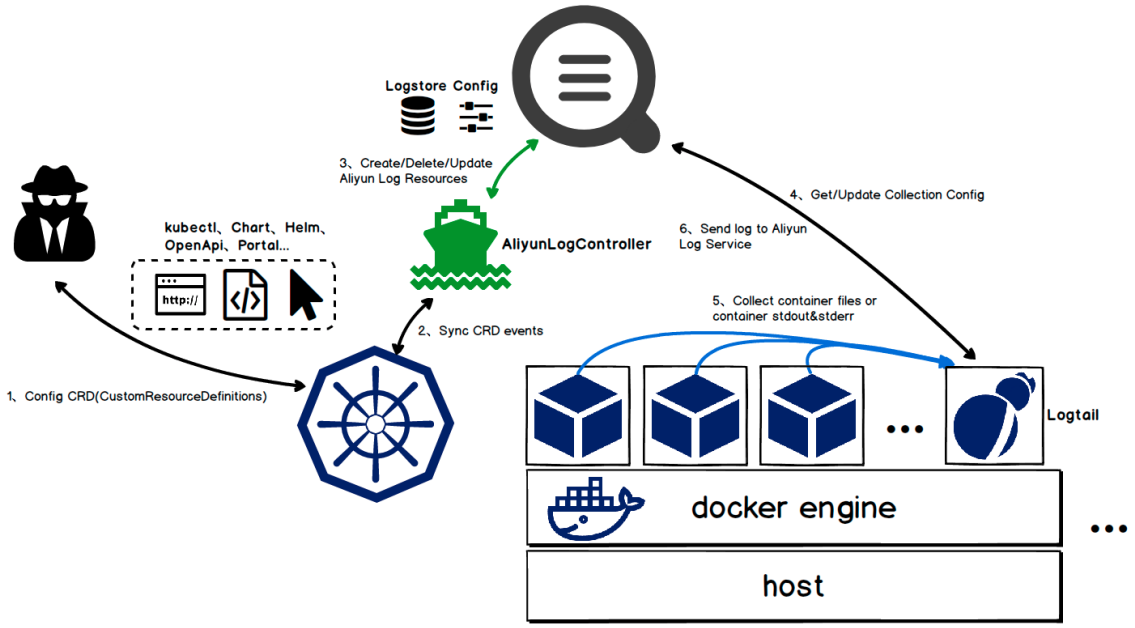
3.3.9 Configure Kubernetes log collection on CRD

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 3-6: 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.alibabacloud.com/v1alpha1 ## Default value, no need
for change
kind: AliyunLogConfig ## 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, automatically 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.
  logtailConfig: ## Detailed configuration
    inputType: plugin ## Input type of collection. Generally, the
value is file or plugin.
    configName: simple-stdout-example ## Collection configuration name
. The value must the same as the resource name (metadata.name).
    inputDetail: ## Detailed configuration information, 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
logtailConfig:
  configName: simple-file-example
  inputDetail:
    dockerFile: true
    dockerIncludeEnv:
      ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
    filePattern: simple.LOG
    logPath: /usr/local/ilogtail
    logType: common_reg_log
  inputType: file
machineGroups: 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 `inputDetail`. For more information on the configuration fields, see *Containers-standard output*.

- **Simple collection mode**

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

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
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
  logtailConfig:
    # docker stdout's input type is 'plugin'
    inputType: plugin
    # logtail config name, should be same with [metadata.name]
    configName: simple-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # collect stdout and stderr
              Stdout: true
              Stderr: true
              # collect all container's stdout except containers
with "COLLECT_STDOUT_FLAG:false" in docker env config
              ExcludeEnv:
                COLLECT_STDOUT_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 3-7: 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_addr=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.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
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
  logtailConfig:
    # docker stdouts input type is plugin
    inputType: plugin
    # logtail config name, should be same with [metadata.name]
    configName: regex-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            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 environment variable configuration 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_addr=(\S+) time_ms=(\d+) size
=(\d+) referer=(\S*). *'
              # Extracted keys
              Keys: ['time', 'level', 'message', 'logger', 'userId
', 'orgId', 'uname', 'method', 'path', 'status', 'remote_addr', '
time_ms', 'size', 'referer']
              # Retain the original fields
              KeepSource: true
              NoKeyError: true
```

```
                NoMatchError: true
```

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

Figure 3-8: Collected log data

Container file

- **Simple file**

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

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
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
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    # logtail config name, must same with [metadata.name]
    configName: simple-file-example
    inputDetail:
      # 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
      filePattern: simple.LOG
      # Collect files from the container. dockerFile flag is set to
true
      dockerFile: true
      # Only collect container with "ALIYUN_LOGTAIL_USER_DEFINED_ID
" in docker env config
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

- **Complete regular expression files**

The following is an example of a Java program log:

```
[2018-05-11T20:10:16,000] [INFO] [SessionTracker] [SessionTra
ckerImpl.java:148] Expiring sessions
java.sql.SQLException: Incorrect string value: '\xF0\x9F\x8E\x8F
",...' for column 'data' at row 1
at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTr
anslator.translate(AbstractFallbackSQLExceptionTranslator.java:84)
at org.springframework.jdbc.support.AbstractFallbackSQLException
```

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.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in your k8s cluster
  name: regex-file-example
spec:
  # logstore name to upload log
  logstore: k8s-file
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    # logtail config name, should be same with [metadata.name]
    configName: regex-file-example
    inputDetail:
      # 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
      filePattern: error.LOG
      # Regular expression for first line
      logBeginRegex: '\[\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_LOGTAIL_USER_DEFINED_ID
" in docker env config
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

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

Figure 3-9: Collected log data

- **Delimiter pattern file**

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

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in your k8s cluster
  name: delimiter-file-example
spec:
  # logstore name to upload log
  logstore: k8s-file
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    configName: delimiter-file-example
    # logtail config name, should be same with [metadata.name]
    inputDetail:
      # 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
      filePattern: 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_LOGTAIL_USER_DEFINED_ID
" in docker env config
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_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.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in you k8s cluster
  name: json-file-example
spec:
  # logstore name to upload log
  logstore: k8s-file
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    # logtail config name, should be same with [metadata.name]
    configName: json-file-example
    inputDetail:
      # 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
      filePattern: 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_LOGTAIL_USER_DEFINED_ID
" in docker env config
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

3.4 Machine Group

3.4.1 Manage a collection 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 *Linux* and *Install Logtail on Windows*. You can create and modify a Logtail configuration of a Logstore on the Logstore List page.

Create a Logtail configuration

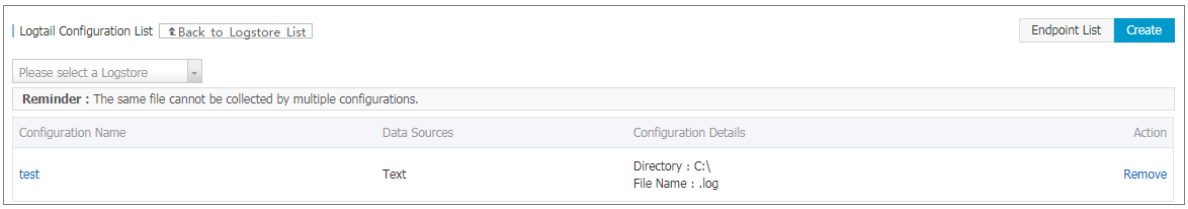
For how to create a Logtail configuration in the Log Service console, see *EN-US_TP_13062.dita* and *Syslog*.

View Logtail configuration list

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.

   All the configurations of this Logstore are displayed on this 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 3-10: 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, or click **manage** on the right.
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 groups that apply this modified configuration. 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, or click **manage** on the right.
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.

3.4.5 Collect logs from non-Alibaba Cloud ECS instances or ECS instances not in your account

To use Logtail to collect logs from non-Alibaba Cloud Elastic Compute Service (ECS) instances or ECS instances that are not created by your account, install Logtail on the server and configure the user identity (account ID) to verify that the server can be accessed by your account. Otherwise, the heartbeat status is set to FAIL and Logtail cannot collect data to Log Service. Follow these steps to configure the user identity (account ID).

Procedure

1. Install Logtail

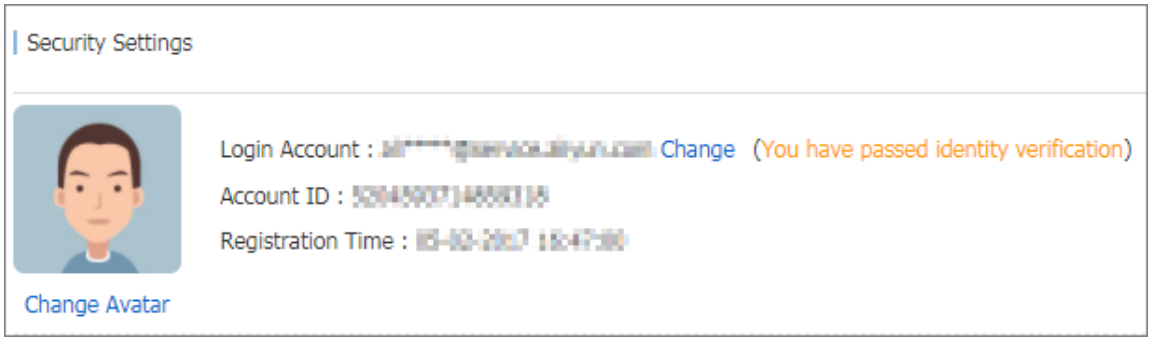
   To install the Logtail client on the server where you want to collect logs, see *Linux* if your operating system is Windows, and *Install Logtail on Windows* if your operating system is Linux.

2. Configure a user identity

   a) View your Alibaba Cloud account ID

   Log on to the Alibaba Cloud Account Management page to view the account ID of your Log Service project.

   Figure 3-11: View account ID



   b) Configure account ID identification file on the server

Create a file named after the account ID in the /etc/ilogtail/users directory. If the directory does not exist, create it manually. You can configure multiple account IDs on a single machine, for example:

```
touch /etc/ilogtail/users/1559122535028493
touch /etc/ilogtail/users/1329232535020452
```

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

```
rm /etc/ilogtail/users/1559122535028493
```

- **Linux System**

  Create a file named after the account ID in the */etc/ilogtail/users directory.* If the directory does not exist, create it manually. You can configure multiple account IDs on a single machine, for example:

  ```
  touch /etc/ilogtail/users/1559122535028493
  touch /etc/ilogtail/users/1329232535020452
  ```

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

  ```
  rm /etc/ilogtail/users/1559122535028493
  ```

- **Windows system**

  Create a file named after the account ID in the *C:\LogtailData\users* directory to configure the user identity. To delete the user identity, delete this file directly.

  For example, *C:\LogtailData\users\1559122535028493*.

  Note:

  - After the user identity (account ID) is configured on a machine, the cloud account has the permission to collect logs from the machine by using Logtail. Clear unnecessary account identification files from machines in time.
  - After adding and removing user identities, You can take effect within 1 minute.

3.5 Troubleshoot

3.5.3 Troubleshoot collection errors

If the log collection fails or the collection status is abnormal when you use Logtail, follow these steps to troubleshoot the errors.

Procedure

1. Check whether the Logtail heartbeat in the machine group is normal

   Log on to the Log Service console and click Machine Status to view the status of the machine group. For more information, see Manage a machine group. If the heartbeat status is normal, move to the next step.

   If the heartbeat status is fail, see *Logtail heartbeat error for troubleshooting*.

2. Check whether the collection configuration is created and applied to the machine group

   After you confirm that the Logtail client status is normal, check the following configurations.

   a) Check whether Logtail configuration is created

      For more information, see Logtail configuration. 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.

   b) Check whether Logtail configuration is applied to the machine group

      See *Manage configurations* in **Manage a machine group**. Check if the target configuration is applied to the machine group.

3. Check for collection errors

   If Logtail is properly configured, check whether new data is generated in real time in the log file. Logtail collects incremental data only, it does not read inventory files if the files are not updated. If the log file is updated but the updates cannot be queried in Log Service, diagnose the problem in the following ways:

   - **Diagnose collection errors**

     See *Query diagnosed errors* to handle the errors according to the error type reported by Logtail.

   - **View Logtail logs**

     Client logs include key INFO logs and all the WARNING and ERROR logs. To see complete and real-time errors, view the client logs in the following paths:

- Linux: `/usr/local/ilogtail/ilogtail.LOG`
- Linux: /usr/local/ilogtail/logtail_plugin.LOG (logs of input sources such as HTTP, MySQL binlog, and MySQL query results)
- Windows x64: `C:\Program Files Program Files (x86)\Alibaba\Logtail\logtail_*.log`
- Windows x32: `C:\Program Files\Alibaba\Logtail\logtail_*.log`

- **Usage exceeds the limit**
  - To collect large volumes of logs, files, or data, you can modify the Logtail startup parameters for higher log collection throughput. For more information, see *Configure startup parameters*.

If the problem persists, open a ticket to contact Log Service engineers and attach the key information collected during troubleshooting to the ticket.

3.6 Limits

Table 3-1: 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 3-2: 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.
Checkpoint storage policy	Regular save every 15 minutes, automatically saved when the program exits.

Item	Capabilities and limits
Checkpoint save path	The default save path is `/tmp/logtail_checkpoint` , you can modify the parameters according to *Configure startup parameters*.

Table 3-3: 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 3-4: 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.
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.

Item	Capabilities and limits
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 3-5: 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 3-6: 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.
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

4.2 DDoS log collection

4.2.1 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 (*Github suffers from 1.35Tbps DDoS attacks*), and the number of 100 GB attacks has doubled. However, application layer attacks are also increasing significantly (*Imperva 2017Q4 DDoS Risk Report*).

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 ####.

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_access_log
  ```

  Query results

Figure 4-1: DDoS access 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_access_log and cc_blocks > 0| SELECT ip_to_country
(if(real_client_ip='-', remote_addr, real_client_ip)) as country,
count(1) as "number of attacks" group by country
```

  The analysis results are displayed in a dashboard:

  Figure 4-2: CC attack source

- For example, view the PV access by the following statement:

```
__topic__: ddos_access_log | select count(1) as PV
```

  The analysis results are displayed in a dashboard:

  Figure 4-3: PV access

- **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_access_log | select ip_to_provider(if(real_clien
t_ip='-', remote_addr, real_client_ip)) as provider, round(sum(
request_length)/1024.0/1024.0, 3) as mb_in group by provider having
```

```
ip_to_provider(if(real_client_ip='-', remote_addr, real_client_ip))
<> '' order by mb_in desc limit 10
```

Analysis results are displayed in the dashboard:

Figure 4-4: Access client distribution

4.2.2 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. Activate Log Service.

Context

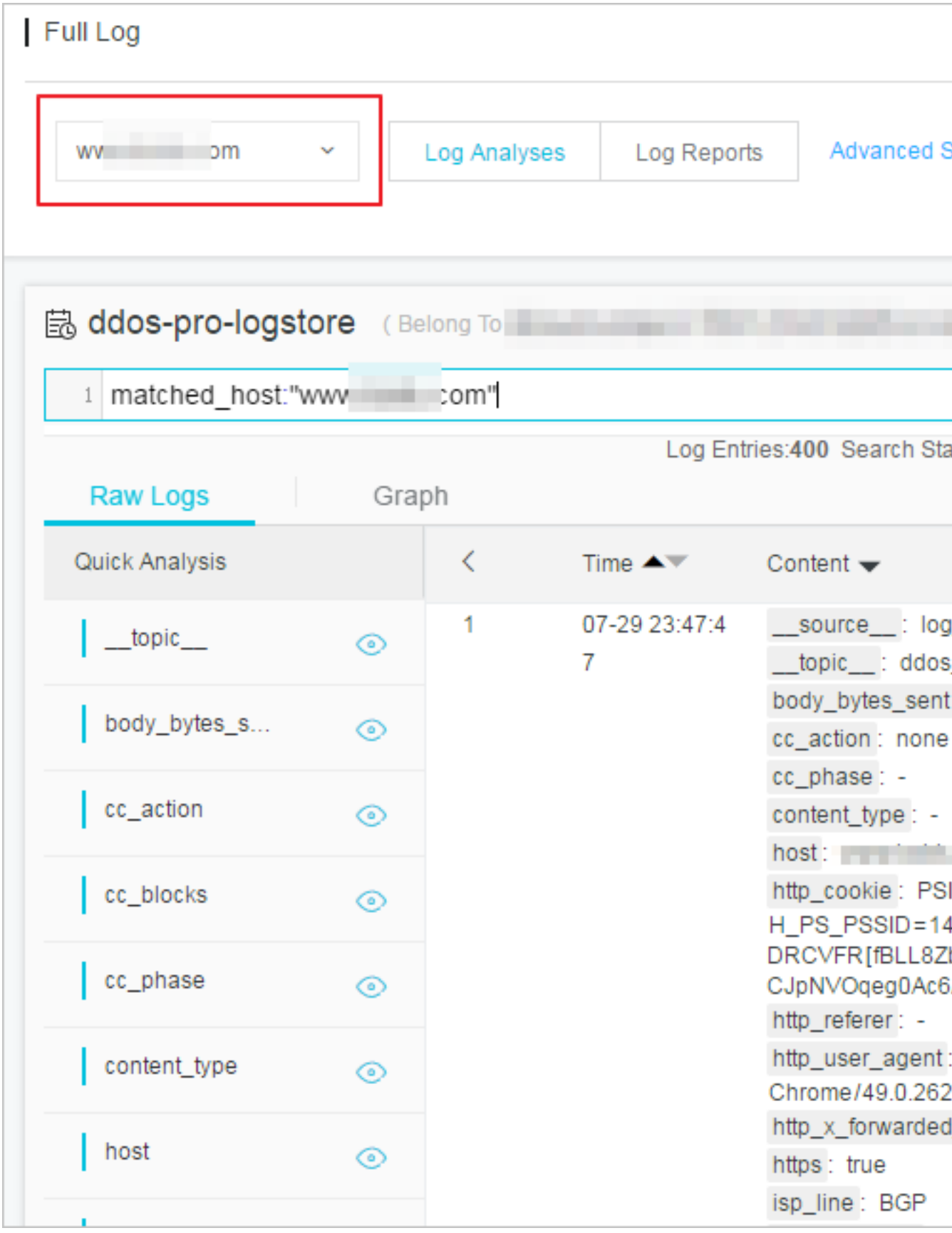
Log Service supports real-time collection of **Alibaba Cloud Anti-DDoS Pro** website access logs, CC attack logs, and supports real-time query and analysis of collected log data. The results of the query are displayed in the form of dashboards, and logs are used to analyze the access and attack behavior 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*.
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 current**Full 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.	-
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

Field	Description	Example
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
cc_action	CC protection policy, such as none, challenge, pass, close, captcha, wait, login, 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

Field	Description	Example
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
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.

4.2.4 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_host: www.aliyun.com` to view log reports based on selected website.

Figure 4-6: 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 4-7: 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 4-8: 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_host: 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 4-9: 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 ####.

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.
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	95%

Chart	Type	Default time range	Description	Example
			total number of all requests.	
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
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

Chart	Type	Default time range	Description	Example
Attack data packets	Single value	1 hour (relative)	The sum of the number of requests for the CC attack , measured in units.	100
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.	-
Access status distribution trend	Flow chart	1 hour (entire point)	Trend chart of various request processing	-

Chart	Type	Default time range	Description	Example
			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 telecommun ications, Unicom , and BGP.	-
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
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	-

Chart	Type	Default time range	Description	Example
			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.	-
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	-

Chart	Type	Default time range	Description	Example
			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.	-
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.	-

4.2.5 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 #####.

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.
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.

Billing item	Description
	• 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
Internet read traffic	0
Read traffic consumption	0

Billing item	FreeTier quota
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 FAQs

- **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**.

     Figure 4-10: Modify log storage time

- **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_access_log | select count(1) as PV
       ```

    4. Click **Query** and select **Statistics Chart** with the chart type **Table** .

Figure 4-11: View log volume

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.

     Figure 4-12: Alarm rule

     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.

4.3 ActionTrail access logs

4.3.1 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 increasingly 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*.

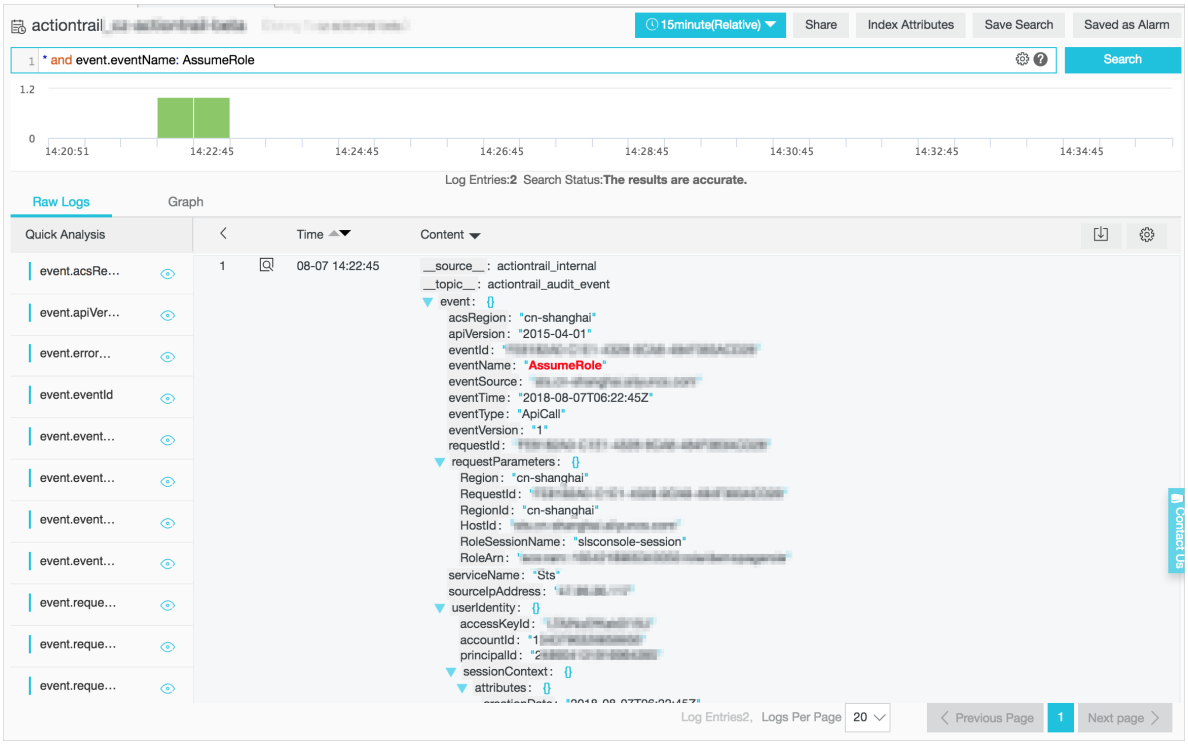
Application scenarios

- **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 4-13: View the ECS release operation log

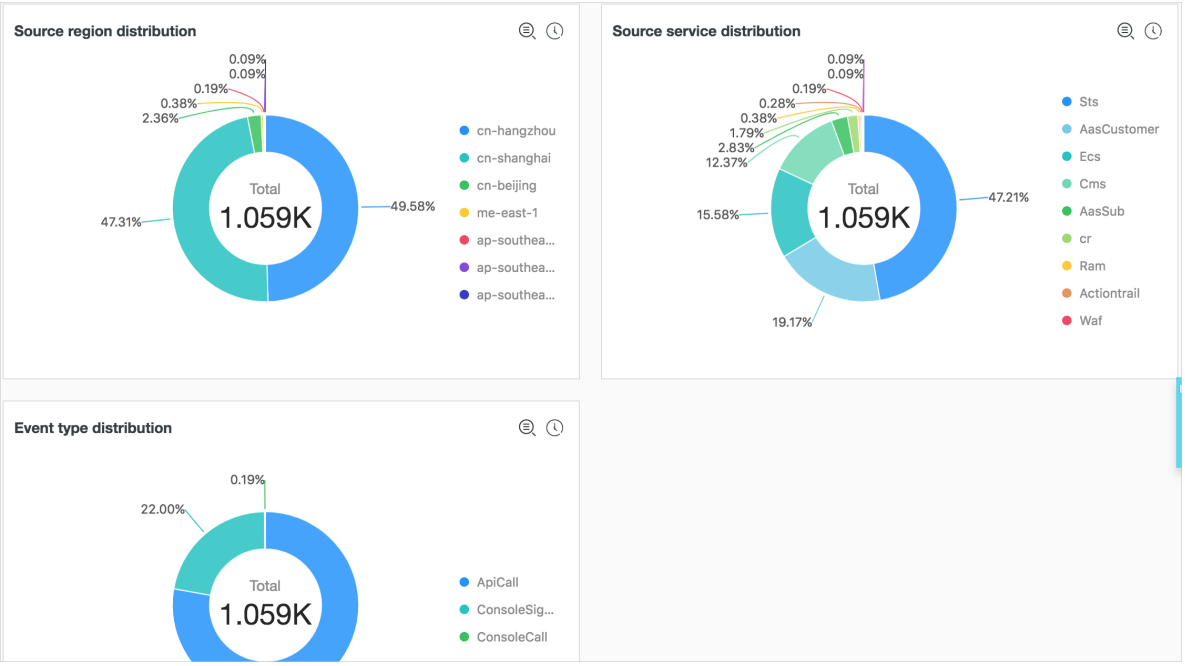


- **Distribution and source tracking of important resource operations**

  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 4-14: View the distribution of RDS deletion

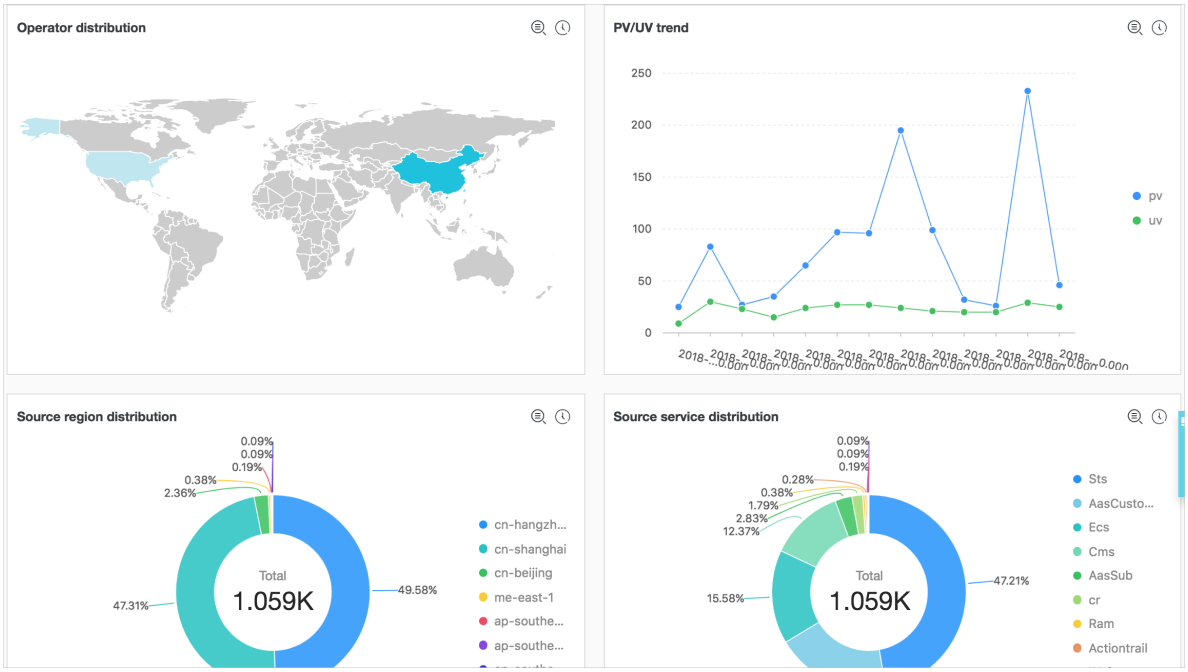


- **Resource operation distribution view**

  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 4-15: Trends of failed operations

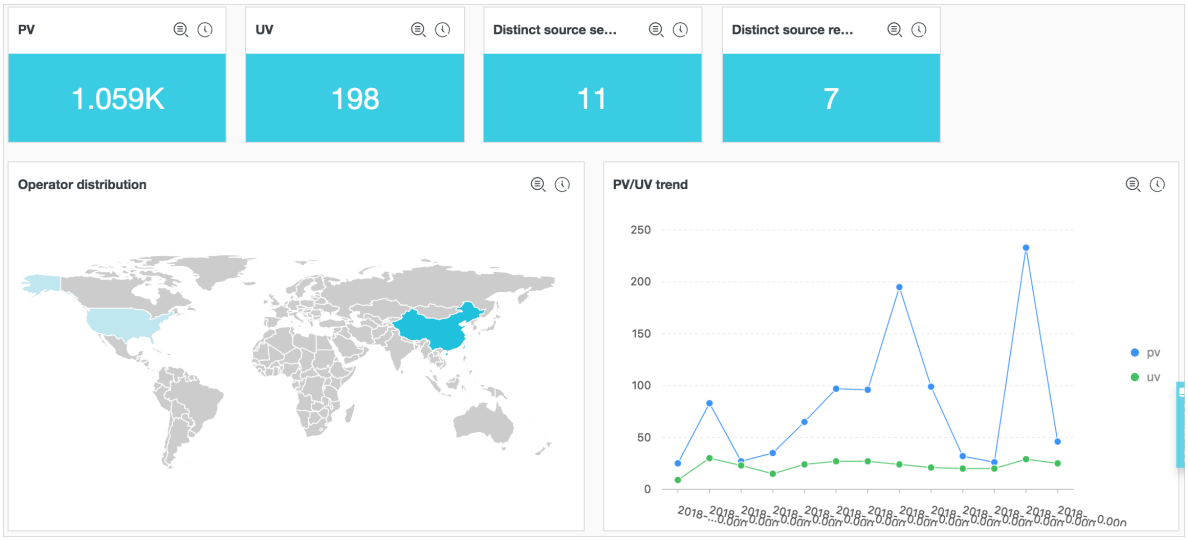


- **Real-time analysis of operation data**

  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 4-16: Frequency distribution of operators from network operators



4.3.2 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

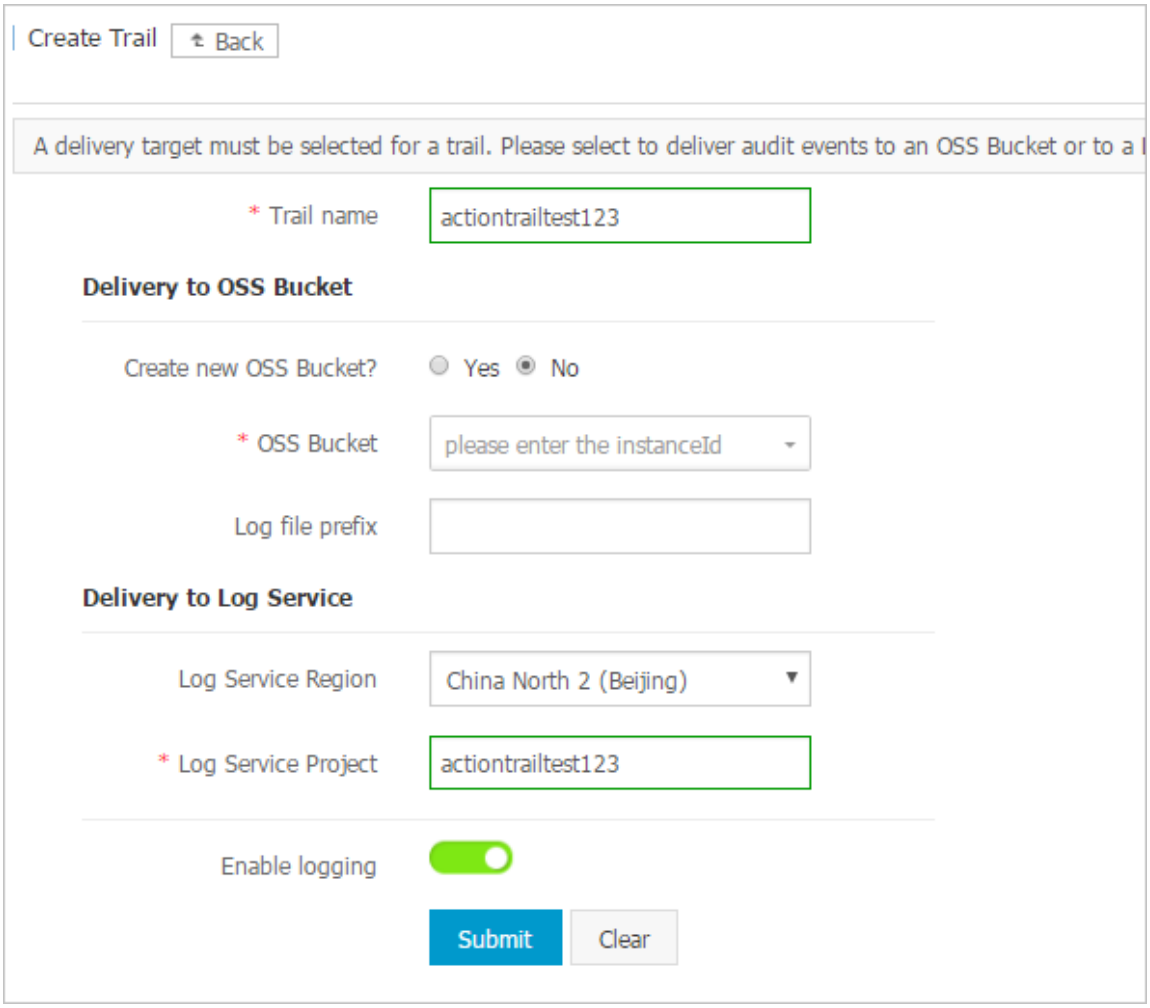
1. Log on to the ActionTrail console.
2. Click **Trail list** in the left-side navigation pane to go to the **Trail list** page.
3. Click **Create Trail** in the upper-right corner to go to the **Create Trail** page.
4. Configure trail parameters.
    1. Enter **Trail name**.
    2. Deliver audit events to an OSS Bucket (optional).

       For more information, see *Create trail*.
    3. Select an region in Log Service Region.
    4. 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.
    5. **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 4-17: Configure trail parameters.

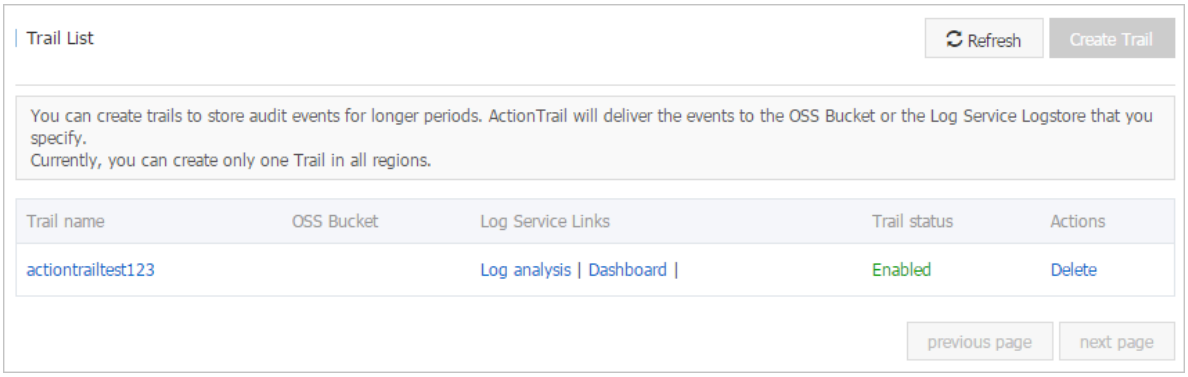


5. Click **Submit** to complete the configuration.

   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 4-18: 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 *Set alarms*.

- **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 *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 4-2: 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.

Default configuration item	Configuration content
	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
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

Field name	Name	Example
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
event.userIdentity.userName	The name of account requested.	root

event example

```
{
  "acsRegion": "cn-hangzhou",
  "additionalEventData": {
    "isMFAChecked": "false",
    "loginAccount": "test1234@aliyun.com"
  },
  "eventId": "7be1e173-1234-44a1-b135-1234",
  "eventName": "ConsoleSignin",
  "eventSource": "http://account.aliyun.com:443/login/login_aliyun.htm
",
  "eventTime": "2018-07-12T06:14:50Z",
  "eventType": "ConsoleSignin",
  "eventVersion": "1",
  "requestId": "7be1e173-1234-44a1-b135-1234",
  "serviceName": "AasCustomer",
  "sourceIpAddress": "42.120.75.137",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.
36",
  "userIdentity": {
    "accessKeyId": "25***************",
```

```
    "accountId": "1234",
    "principalId": "1234",
    "type": "root-account",
    "userName": "root"
  }
}
```

5 Index and query

5.1 Overview

Log Service provides the LogSearch/Analytics function to query and analyze large amounts of logs in real time. You can use this function by enabling the index and field statistics.

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.
- Ecologic: 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.

Basic concepts

Without enabling the LogSearch/Analytics (index) function, raw data is consumed according to the sequence in the shard, which is similar to Kafka. With the LogSearch/Analytics (index) function enabled, besides the consumption in sequence, you can also count and query the logs. For the difference between log consumption and log query, see Differences between log consumption and log query.

Enable an index

1. Log on to the Log Service console. On the Project List page, click the project name.
2. Select the Logstore, and click **Search**. Then, click **Enable Index** in the upper-right corner. If you have enabled the index before, click **Index Attributes** > **Modify.**.
   - After enabling the query and statistics, data is indexed in the backend. Traffic and storage space for the index are required.
   - ◦If this function is not required, click **Disable** to disable it.

3. Enter the Settings menu to complete configuration.

Data types

You can configure the type of each key in a log (full text index is a special key, whose value is the log). Currently, Log Service supports the following data types.

Category	Type	Description	Query example
Basic	*TEXT*	The text type that supports keyword and fuzzy match.	`uri:"login*" method:"post"`
Basic	*Long*	The value type that supports interval query.	`status>200, status in [200, 500]`
Basic	*Double*	The value type with a float.	`price>28.95, t in [20.0, 37]`
Combination	*JSON*	The content is a JSON field, which is of the text type by default and supports the nested model. You can configure indexes of text, long, and double type for element b under a by using the path format such as a.b . The field type after the configuration is subject to the configuration.	`level0.key>29.95 level0.key2:"action"`
Combination	*Full text*	Use a log as the text for query.	`error and "login fail"`

Query and analysis syntax

Real-time query and analysis is composed of Search and Analytics, which are separated with a vertical line (|):

`$Search |$Analytics`

- •Search: The query condition, which is generated by using keywords, fuzzy match conditions, values, ranges, and combination conditions. If Search is empty or an asterisk (*), all data is queried.
- Analytics: Calculate and count the query results or the full 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.

Note:

For more information, see *Query syntax*, *Syntax description*.

Query examples

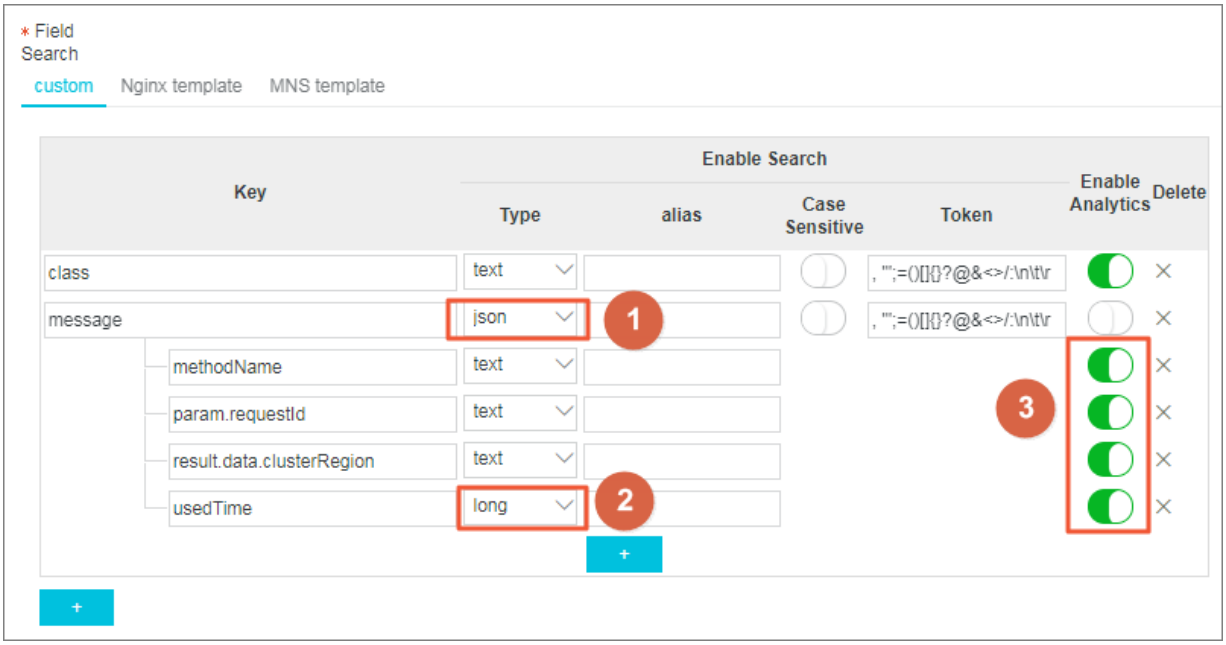
Besides time, the following log also contains four key values.

Sequence number	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": "getProjectInfo",
      "success": true,
      "remoteAddress": "1.1.1.1:11111",
      "usedTime": 48,
      "param": {
              "projectName": "ali-log-test-project",
              "requestId": "d3f0c96a-51b0-4166-a850-f4175dde7323"

      "result": {
          "message": "successful",
          "code": "200",
          "data": {
              "clusterRegion": "ap-southeast-1",
              "ProjectName": "ali-log-test-project",
              "CreateTime": "2017-06-08 20:22:41"

          "success": true
      }
```

Configuration is as follows:

Figure 5-1: Index settings



Where:

- ① indicates that all the data of the string type and bool type in the JSON field can be queried.
- ② indicates that data of the long type can be queried.
- ③ indicates that you can analyze the configured field by using SQL statements.

Example 1: Query string, bool type

```
class : cental*
message.traceInfo.requestId : 92.137_1518139699935_5599
message.param.projectName : ali-log-test-project
message.success : true
```

Note:

- No configurations in the JSON field are needed.
- JSON Map and Array are auto scaling and support multi-level nesting. Each layer is separated with a period (.).

Example 2: Query double、long type

```
latency>40
message.usedTime > 40
```

Note:

You configure JSON fields independently. The fields must not be in array.

Example 3: Combined query

```
class : cental* and message.usedTime > 40 not message.param.projectNam
e:ali-log-test-project
```

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.

5.2 Text type

Similar to search engines, text data is queried based on terms. Therefore, you must configure the Case Sensitive and Token.

Instructions

Case sensitive

Determine whether or not to support case sensitive 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,an error occurred in this field
          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" .

5.3 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*.

5.4 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"` and `jsonkey.key2:true`.

```
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.

Example:

```
"json_string":

     "key_1" : "value_1",
     "key_map" :

             "key_2" : "value_2",
             "key_3" : "valu
```

Data after key_3 is truncated and lost. The field `json_string.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 ####.

6 Real-time analysis

6.2 Analysis grammar

6.2.2 Map map function

The Log Service Query analysis function supports log analysis by mapping functions, with detailed statements and implications as follows:

Statements	Meaning	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` .	`latency > 10 | latency > 10 | 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.	-
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 multidimensional array into a map.	`SELECT map_from_entries (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_entries(MAP (ARRAY[1, 2], ARRAY[‘x`

Statements	Meaning	Example
		', '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.	-
map_zip_with(map<K, V1>, map<K, V2>, function<K, V1, V2, V3>) → map<K,V3>	Refer to power functions in Lambda.	-

6.2.3 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.

Statement	Meaning	Example
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])

Statement	Meaning	Example
`numeric_histogram( buckets, Value)`	Makes statistics on the value column in different buckets. Divides the value column into buckets number of buckets and returns the key and count of each bucket, which is equivalent to `select count group by` for values.	For post requests, divide the delay into 10 barrels, returns the size of each bucket: method: `method:POST | select numeric_histogram(10,latency)`

6.2.4 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_size)`
`covar_pop(y, x)`	Calculates the population covariance.	`latency>100| select covar_pop(request_size, latency)`
`covar_samp(y, x)`	Calculates the sample covariance.	`Latency> 100 | select covar_samp(request_size,latency)`
`regr_intercept(y, x)`	Returns the linear regression intercept of input values. y is the dependent value. x is the independent value.	`latency>100| select regr_intercept( request_size,latency)`
`regr_slope(y,x)`	Returns the linear regression slope of input values. y is the dependent value. x is the independent value.	`latency>100| select regr_slope(request_size,latency)`
`stddev(x)` or `stddev_samp (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)`

Statements	Meaning	Example
`var_pop(x)`	Calculates the population variance of column x.	`latency>100| select variance(latency)`

6.2.5 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_similarity(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.

Function name	Meaning
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.
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_infinity(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.

6.2.6 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
`length(x)`	Returns the length of a field.
`levenshtein_distance(string1, string2)`	Returns the minimum edit distance between two strings.
`lower(string)`	Converts the string to lowercase characters.
`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 No. offset string. The generated result is an array with subscripts starting at 1.
`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 of two strings.

Note:
Enclose strings with single quotation marks and column names with double quotation marks. For example, a='abc' means column a = string abc, and a = "abc" means column a = column abc.

6.2.8 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_extract_fragment(url)`	Extracts the fragment from a URL and the result is of varchar type.	`*| select url_extract_fragment(url)`
`url_extract_host(url)`	Extracts the host from a URL and the result is of varchar type.	`*|select url_extract_host(url)`
`url_extract_parameter(url, name)`	Extracts the value of the name parameter in the query from a URL and the result is of varchar type.	`*|select url_extract_parameter(url)`
`url_extract_path(url)`	Extracts the path from a URL and the result is of varchar type.	`*|select url_extract_path(url)`
`url_extract_port(url)`	Extracts the port from a URL and the result is of bigint type.	`*|select url_extract_port(url)`
`url_extract_protocol(url)`	Extracts the protocol from a URL and the result is of varchar type.	`*|select url_extract_protocol(url)`
`url_extract_query(url)`	Extracts the query from a URL and the result is of varchar type.	`*|select url_extract_query(url)`
`url_encode(value)`	Encodes a URL.	`*|select url_encode(url)`
`url_decode(value)`	Decodes a URL.	`*|select url_decode(url)`

6.2.10 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.

Log Service supports the following common JSON functions.

Function name	Meaning	Example
`json_parse(string)`	Converts a string to the JSON type.	`SELECT json_parse('[1, 2, 3]')` returns a JSON array
`json_format(json)`	Converts the JSON type to a string.	`SELECT json_format(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_extract(json, json_path)`	Extracts the value from a JSON object. The JSON path syntax is similar to `$.store.book[0].title`. The returned result is a JSON object.	`SELECT json_extract(json, '$.store.book');`
`json_extract_scalar(json, json_path)`	Similar to `json_extract`, 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`

6.2.11 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
```

6.2.12 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_do_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. If the IP resides outside of China, the country name is returned.	`SELECT ip_to_province(ip)`
`ip_to_city(ip)`	Determines the city in which the IP resides. If the IP resides outside of China, the country name is returned.	`SELECT ip_to_city(ip)`
`ip_to_geo(ip)`	Judging 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_provider(ip)`	Obtains the network operator of the IP.	`SELECT ip_to_provider(ip)`
`ip_to_country(ip,'en')`	Obtains the network operator of the IP.	`SELECT ip_to_country(ip,'en')`

Function Name	Meaning	Example
`ip_to_country_code(ip)`	Obtains the network operator of the IP.	`SELECT ip_to_country_code(ip)`
`ip_to_province(ip,'en')`	Judging the province where IP is located, return to the English province name or Chinese alphabet.	`SELECT ip_to_province(ip,'en')`
`ip_to_city(ip,'en')`	Judging the city where IP is located, return to 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

```
* | selectcount(1)whereip_to_domain(ip)! ='intranet'
```

- View the top 10 access provinces

```
* | SELECT count(1) as pv, ip_to_province(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 ArabEmirates (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_province(ip) as province WHERE
ip_to_domain(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_country(ip) as country group by country limit 100
```

- View average latency for different network operators

```
* | SELECT AVG(latency) , ip_to_provider(ip) as provider group by
provider limit 100
```

- Check the average latency of different network operators

```
* | 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

6.2.13 GROUP BY syntax

GROUP BY supports multiple columns and indicating the corresponding KEY by using the SELECT column alias.

Example:

```
method:PostLogstoreLogs |select avg(latency),projectName,date_trunc('
hour',__time__) as hour group by projectName,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:PostLogstoreLogs |select avg(latency) group by cube(projectName
,logstore)
method:PostLogstoreLogs |select avg(latency)   group by GROUPING SETS
 ( ( projectName,logstore), (projectName,method))
method:PostLogstoreLogs |select avg(latency) group by rollup(
projectName,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) gropu by a
```

To achieve the preceding aim, use the arbitrary function to output b:

```
*|select a, arbitrary(b), count(c) gropu by a
```

6.2.14 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.
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_rank order by
department,sallary_rank
```

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_percentage
```

Response results:

department	persionId	sallary	sallary_percentage
dev	john	9000	0.3
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_distinct(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

6.2.15 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 :PostLogstoreLogs |select avg(latency),projectName group by
projectName 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 teporature > 10groupby province
having avg(rain) > 100
```

6.2.16 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 :PostLogstoreLogs |select avg(latency) as avg_latency,
projectName group by projectName
HAVING avg(latency) > 5700000
order by avg_latency desc
```

6.2.17 LIMIT syntax

LIMIT is followed by a number to indicate the maximum number of rows in the output results. If no LIMIT statement is added, only 10 rows are output by default.

Note:

Limit offset and lines syntaxes are not supported.

Example:

```
*| select avg(latency) as avg_latency , methodgroupbymethodorderbyavg_
latencydesclimit100
```

6.2.18 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_percentage
```

- 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_slot,
 count(1) as pv
 group by latency_slot
```

IF syntax

The if syntax is logically equivalent to the CASE WHEN syntax.

```
Case
     WHEN condition THEN true_value
     [ ELSE false_value ]
 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)
```

6.2.19 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
```

6.2.20 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.	-
array_max(x) → x	Obtain the maximum value in array x.	
array_min(x) → x	Obtain the minimum value in array x.	-
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.	-

Statement	Meaning	Example
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.	-
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')]`

Statement	Meaning	Example
zip_with(array1, array2, function) → array	See zip_with() in Lambda.	-

6.2.21 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 ||.
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.

6.2.22 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.	-

6.2.23 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

Operator	Meaning
<>	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 和 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

a	b	a = b	a <> b	a DISTINCT b	a NOT DISTINCT b
NULL	NULL	NULL	NULL	FALSE	TRUE

GREATEST 和 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
```

6.2.25 Logical functions

Logical operators

Table 6-1: 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 6-2: 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 6-3: Truth Table 2

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

6.2.26 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 6-4: Alias Example:

Original column name	Alias
User-Agent	ua
User.Agent	ua
123	col
abceefghijklmnopqrstuvw	a

6.2.27 Geospatial functions

Geospatial concept

Geospatial functions support the geometries in the Well-Known Text (WKT) format.

Table 6-5: Geometry format

Geometry	Well-maid text (WKT) Format
Point	`POINT (0 0)`
Line string	`LINESTRING (0 0, 1 1, 1 2)`

Geometry	Well-maid text (WKT) Format
Polygon	`Polygon`
Multi-point	`MULTIPOINT (0 0, 1 2)`
Multi-line string	`MULTILINESTRING ((0 0, 1 1, 1 2), (2 3, 3 2, 5 4))`
Multi-polygon	`MULTIPOLYGON (((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	`GEOMETRYCOLLECTION (POINT(2 3), LINESTRING (2 3, 3 4))`

Constructors

Table 6-6: 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.

Function	Description
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.
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.

Function	Description
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 mathematic al 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.
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.

Function	Description
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.

6.2.28 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 6-7: Geo functions

Function	Description	Example
geohash(string)	Returns the geohash value of the specified geographical 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 geographical coordinate. The geographical coordinate is represented by two separate parameters that indicate the latitude and longitude.	`select geohash(34.1,120.6)= 'wwjcbrdnzs'`

6.2.29 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. logexception import logexception
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
From aliyun. log. getlogsrequest import getlogsrequest
from aliyun.log.getprojectlogsrequest import GetProjectLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listtopicsrequest import ListTopicsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest
from aliyun.log.index_config import *
from aliyun.log.logtail_config_detail import *
from aliyun.log.machine_group_detail import *
from aliyun.log.acl_config import *
if __name__=='__main__':
    token = None
    endpoint = "http://cn-hangzhou.log.aliyuncs.com"
    accessKeyId = 'LTAIvKy7U'
    accessKey='6gXLNTLyCfdsfwrewrfhdskfdsfuiwu'
    client = LogClient(endpoint, accessKeyId, accessKey,token)
    logstore = "meta"
    # In the query statement, specify two Logstores. For each Logstore
 specify its time range and the key
    req = GetProjectLogsRequest(project,"select count(1) from
sls_operation_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)
```

6.3 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_unixtime(__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_distinct(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
```

6.4 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_count,window_time)/1.0/sum
(error_count) as error_ratio, sum(error_count) as total_error from (
select __topic__, count(*) as error_count , __time__ - __time__ % 300
as window_time from log group by __topic__, window_time

group by __topic__ having max_by(error_count,window_time)/1.0/sum(
error_count) > 0.4 and sum(error_count) > 500 order by total_error
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_num, date_format(date_trunc('minute',
__time__), '%m-%d %H:%i') as time group by time order by time limit
100
```

6.5 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.

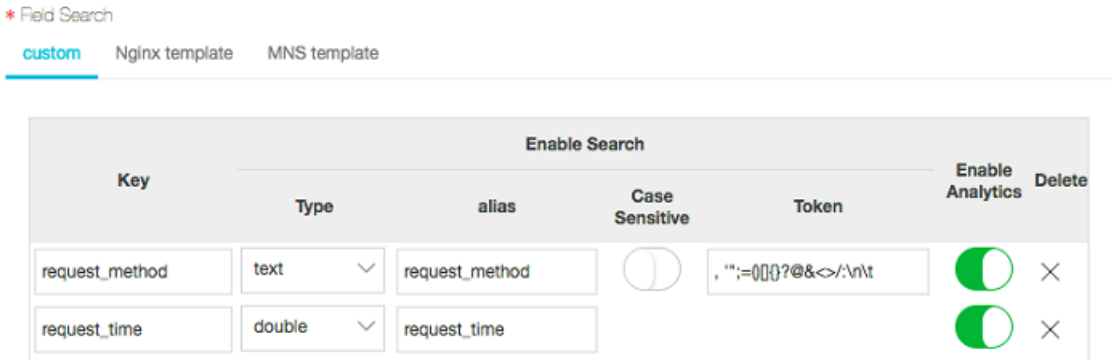
Prerequisite

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_method` and `request_time`, you can configure the following settings.

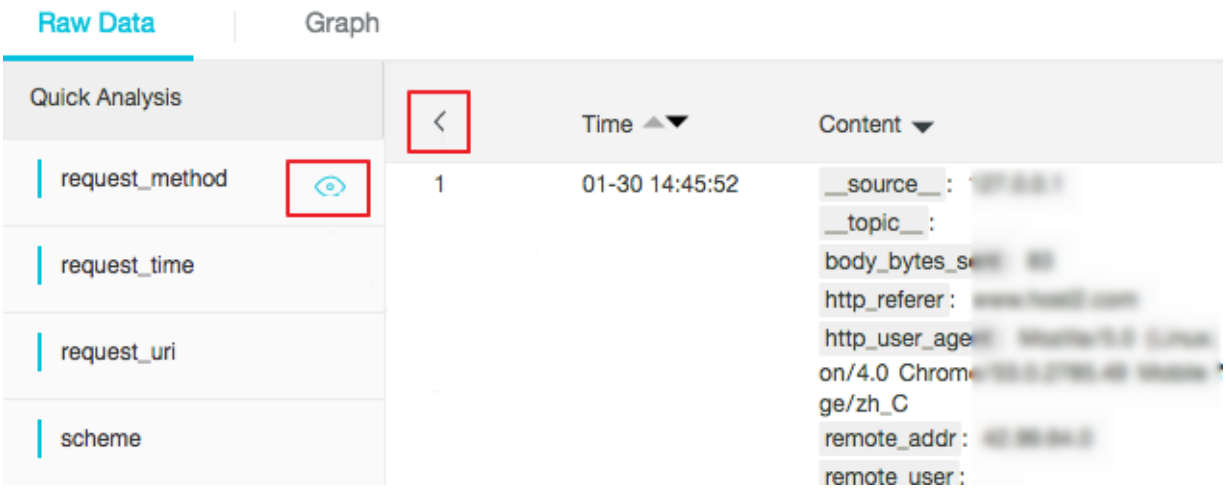
Figure 6-1: Prerequisites



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 6-2: Original log



Text

- **Grouping statistics for Text fields**

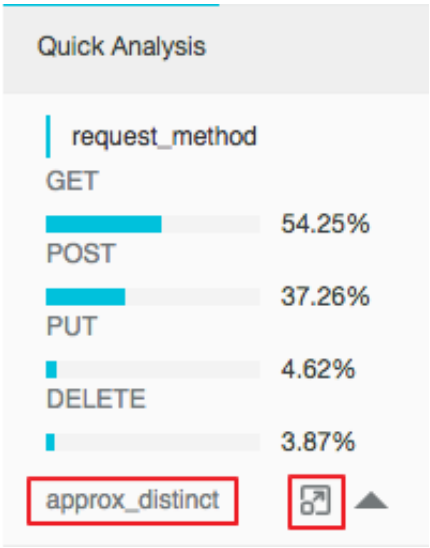
  Click the **eye** button at the right of the filed to quickly group the first 100,000 pieces of data of this **Text** field and return the ratio of the first 10 pieces.

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_method` returns the following result based on the grouping statistics, where GET requests are in the majority.

Figure 6-3: Group statistics



- **Check the number of unique entries of the field**

  Under the target fields in **Quick Analysis**, click**approx_distinct** to check the number of unique entries for `${keyName}`.

  `request_method` 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

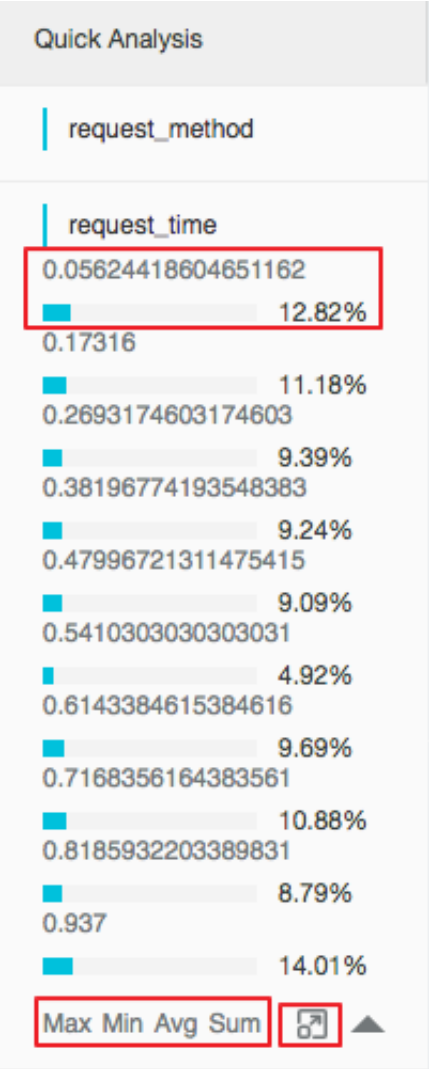
- **Histogram statistics for the approximate distribution**

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 6-4: Request Distribution



- **Quick analysis of the`MaxMinAvgSum` 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.

6.6 Use JDBC to query and analyze logs

In addition to ##, you can use JDBC and standard SQL 92 for log query and analysis.

Connection parameters

Connection parameter	Example	Description
host	regionid.example.com	####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 -ubq2sjzesjmo86kq -
p4fdO1fTDDuZP -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.ResultSetMetaData;
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 = DriverManager.getConnection("jdbc:mysql://cn-
shanghai-intranet.log.aliyuncs.com:10005/sample-project","accessid","
accesskey");
            System.out.println("Connected database successfully...")
            //STEP 4: Execute a query
            System. Out. println ("creating statement ...");
            stmt = conn.createStatement();
            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='Postlogstorelogs' or method='GetLogtailConfig') 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 = 'getlogstorelogS
' or method='Get**' and method <> 'GetCursorOrData' )";
            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.executeQuery(sql);
            //STEP 5: Extract data from result set
            while(rs.next()){
                //Retrieve by column name
                ResultSetMetaData data = rs.getMetaData();
                System.out.println(data.getColumnCount());
                for(int i = 0;i < data.getColumnCount();++i) {
                    String name = data.getColumnName(i+1);
                    System.out.print(name+":");
                    System.out.print(rs.getObject(name));

                System.out.println();

            Rs. Close ();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (Exception e) {
            E. printstacktrace ();
        } Finally {
            if (stmt ! = null) {
                try {
                    Stmt. Close ();
```

```
                } catch (SQLException e) {
                    e.printStackTrace();


            if (conn ! = null) {
                try {
                    conn.close();
                } catch (SQLException e) {
                    e.printStackTrace();
```

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 ###### carefully.

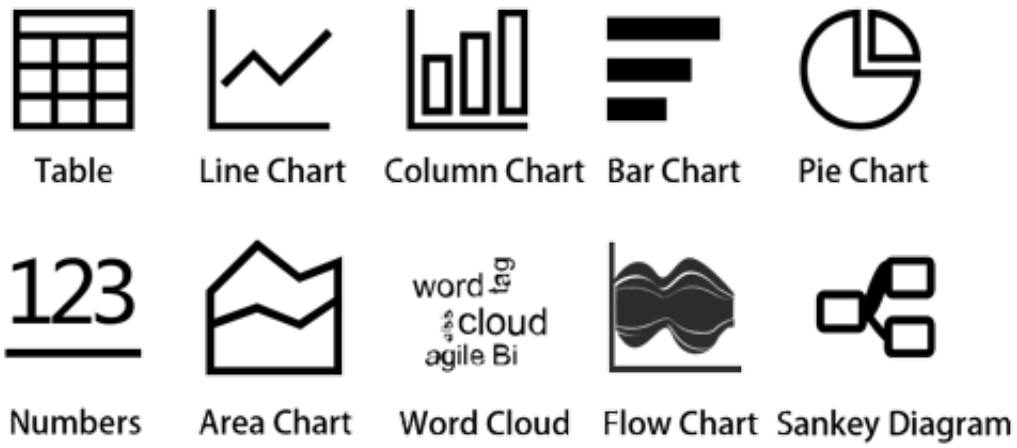
Prerequisite

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:

- ##
- ###
- ###

- ###
- ##
- ###
- ###
- ##
- ##
- ###
- ##

7.1.2 Dashboard

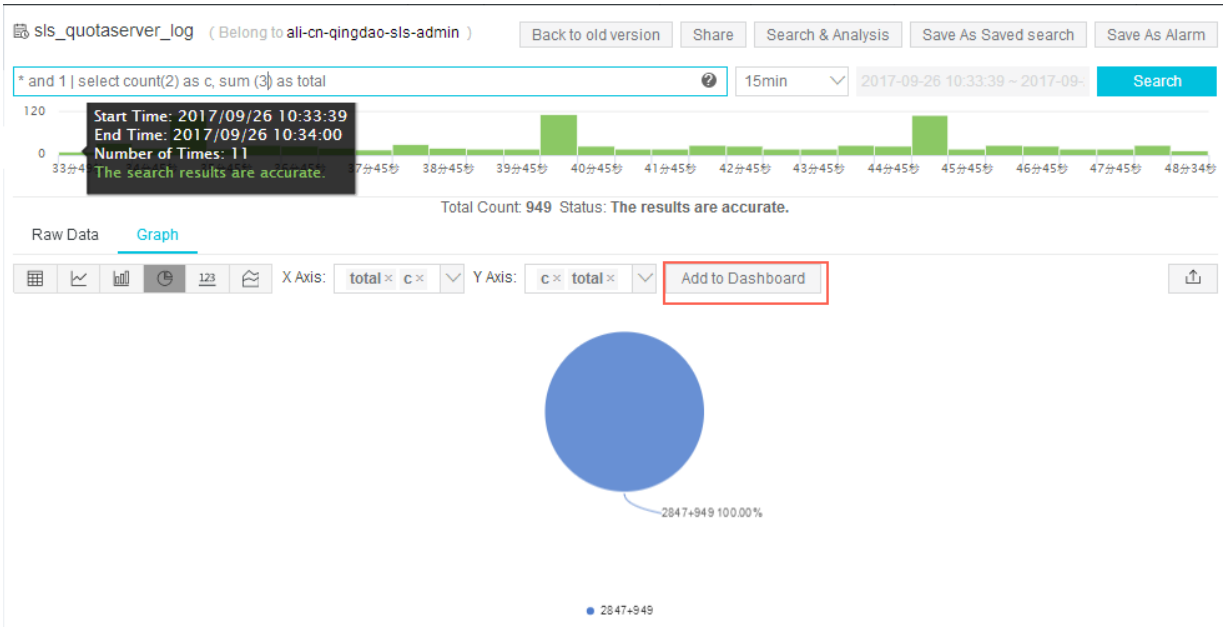
After you enable the LogSearch/Analytics function, in addition to entering a query condition in the search box, you can save frequently used queries to the following locations:

- Dashboard
- Saved Search
- Saved Search

Procedure

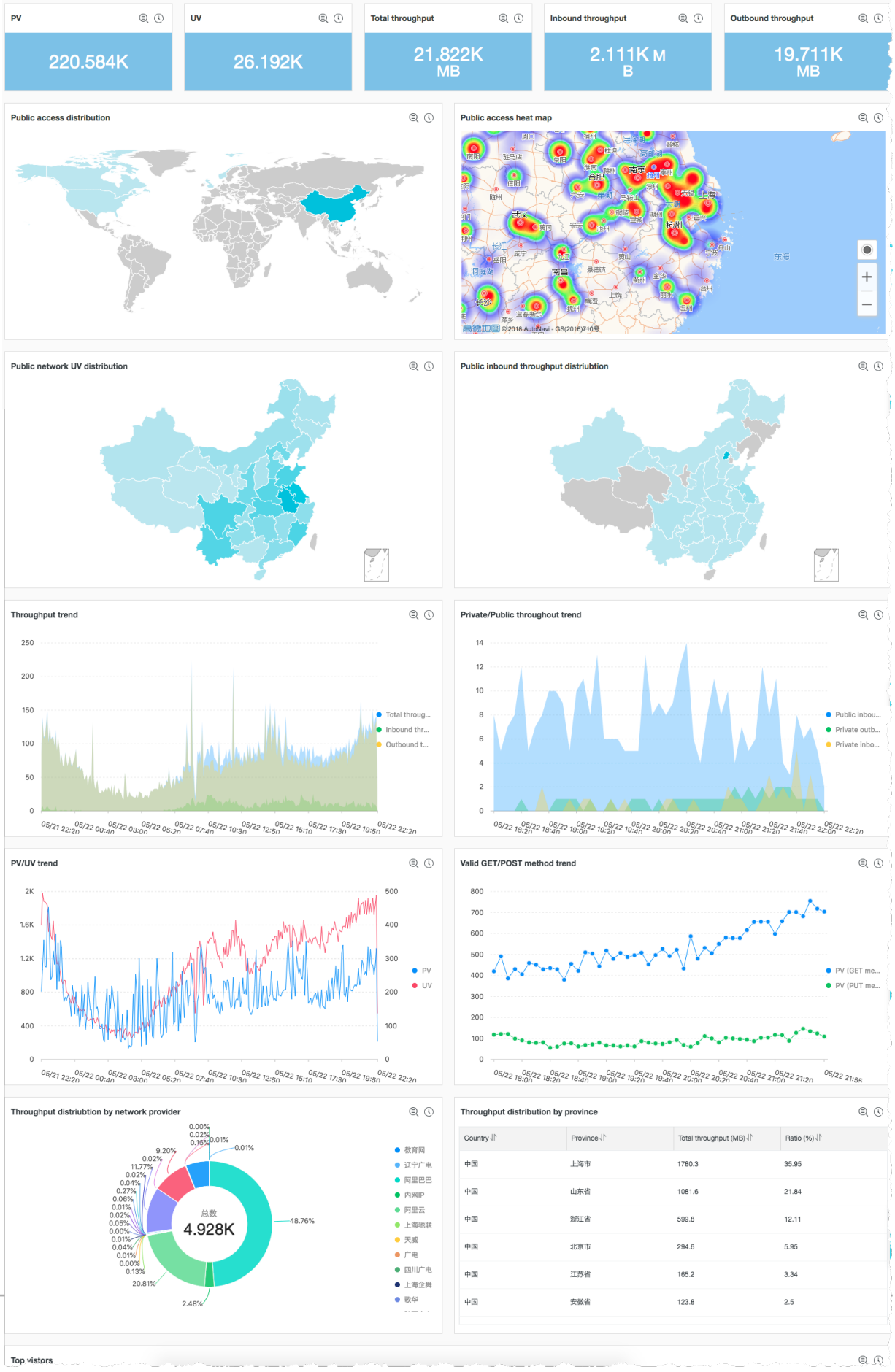
1. 1.Enter the query and analysis statement in the search box and then click **Search**.
2. Add a dashboard for the query and analysis view. Select a view on the query and analysis page, and click **Add to Dashboard**.
3. In the displayed dialog box, click **New** or **Add to Existing Dashboard**. Enter the dashboard name and the table name.

Figure 7-2: Add to Existing Dashboard



Dashboard sample:

Figure 7-3: Dashboard



Other Operations

- **View**: View an existing **Dashboard** in the following ways.
  - Click **LogSearch/Analytics** > **Dashboard** in the left-side navigation pane to view the dashboard.
  - On the query and analysis page, click **New Tab** in the left-side navigation pane and click the **dashboard tab** on the displayed page to view the dashboard.
- **Edit**: Click Edit on the dashboard page to adjust the icon attributes, size, and location. You can also click Full Screen or Refresh for better effect.

Limits

- Each project can create up to five dashboards. Each dashboard can create at most 10 analytic queries for simultaneous display.
- Display by line charts, bar charts, pie charts, numeric values, and area charts is supported.
- You can customize the positions and adjust the sizes of individual charts.

7.1.3 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

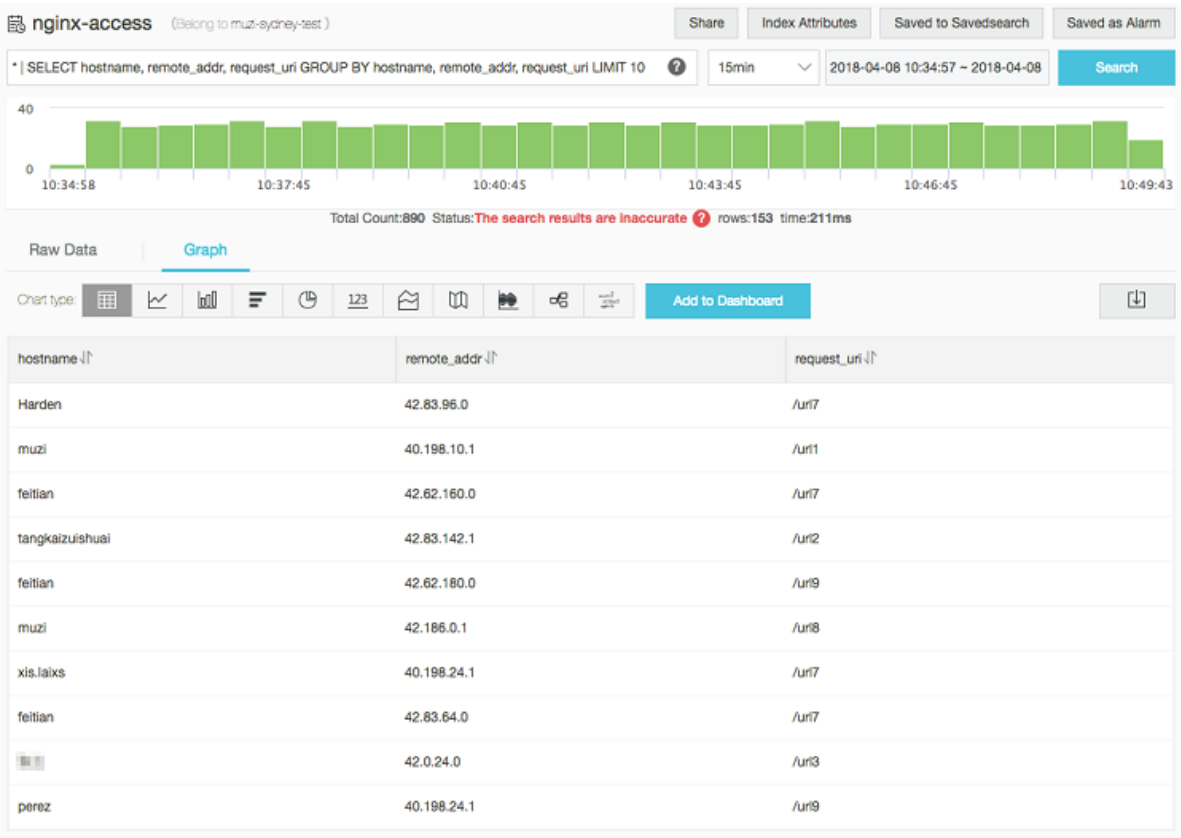
The raw log is as follows.

Figure 7-4: 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_addr, request_uri GROUP BY hostname,
remote_addr, request_uri LIMIT 10
```

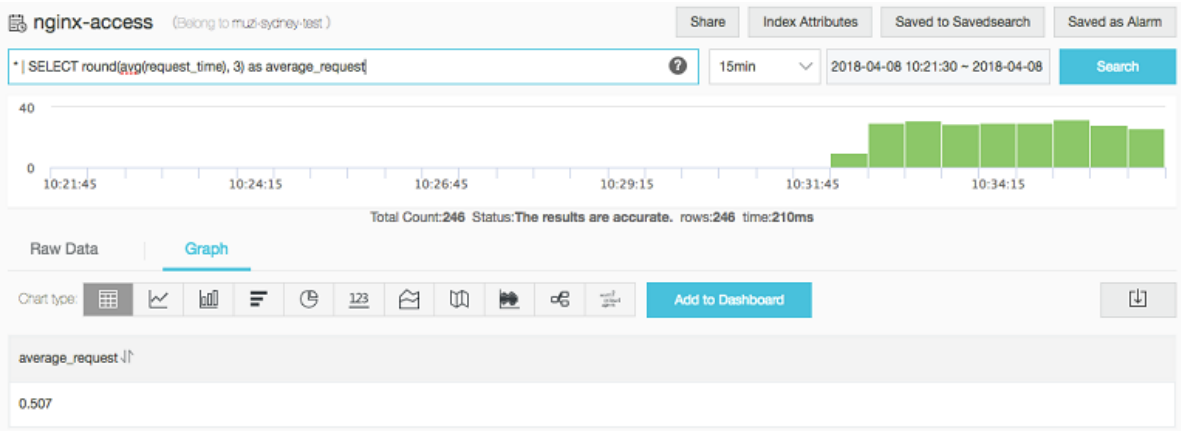
Figure 7-5: case 1



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_time), 3) as average_request
```

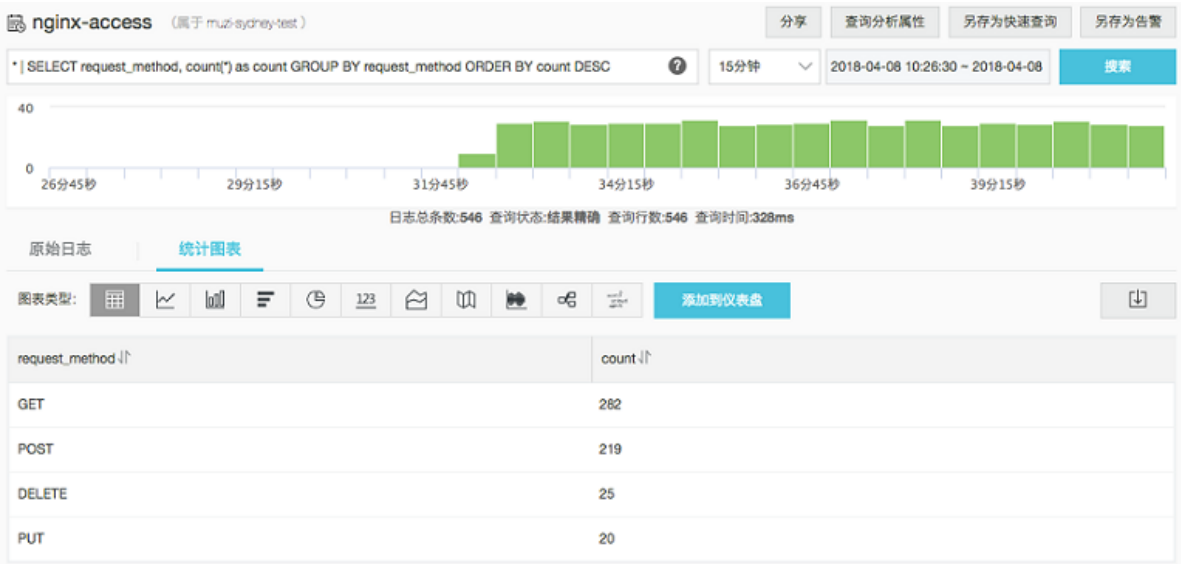
Figure 7-6: case 2



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_method, count(*) as count GROUP BY request_method
 ORDER BY count DESC
```

Figure 7-7: case 3



7.1.5 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_referer` in the current time interval, the statement is as follows:

```
* | select http_referer, count(1) as count group by http_referer
```

Select `http_referer` 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_referer` in the current time interval, the statement is as follows:

```
* | select http_referer, count(1) as count, avg(body_bytes_sent) as
avg group by http_referer
```

Select `http_referer` as the X Axis, and select `count` and `avg` as the Y Axis.

Figure 7-9: Grouped column chart

7.1.6 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.

Description	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.

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:**

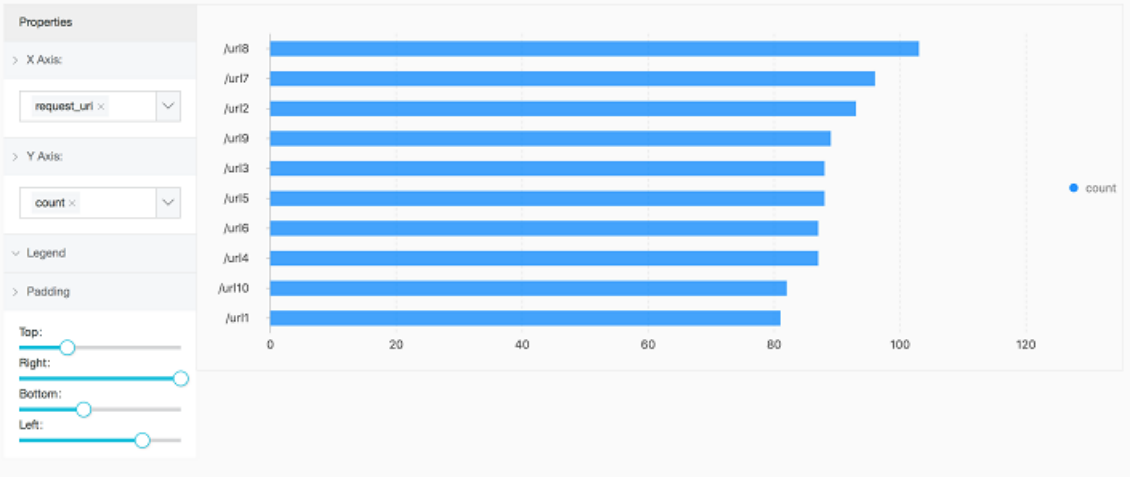
- Use the bar chart if the number of data types is no more than 20. We recommend that you use `LIMIT` to control the number of data types in case that the vertical height is so large that the analytical comparison is not intuitive, and use the`ORDER BY` syntax when analyzing the top scenario. We also 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, count(1) as count group by request_uri order
by count desc limit 10
```

Figure 7-10: Simple bar chart



7.1.7 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. 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. Configure the graph properties.

> **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.

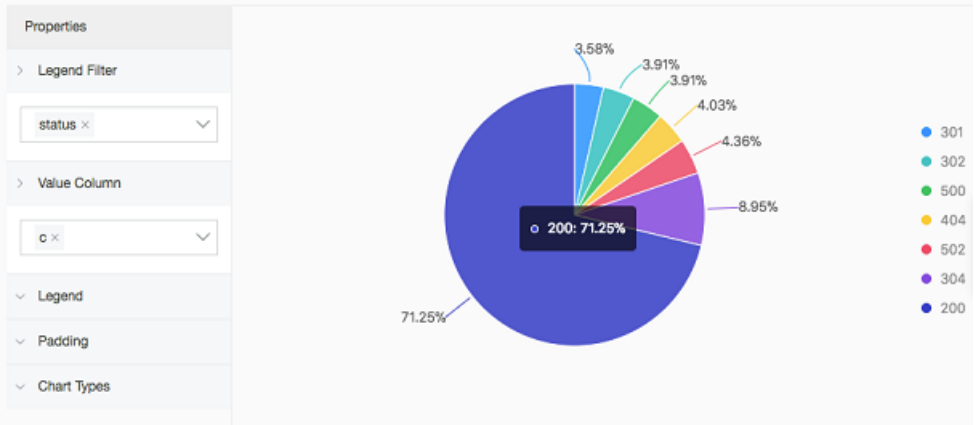
Example

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

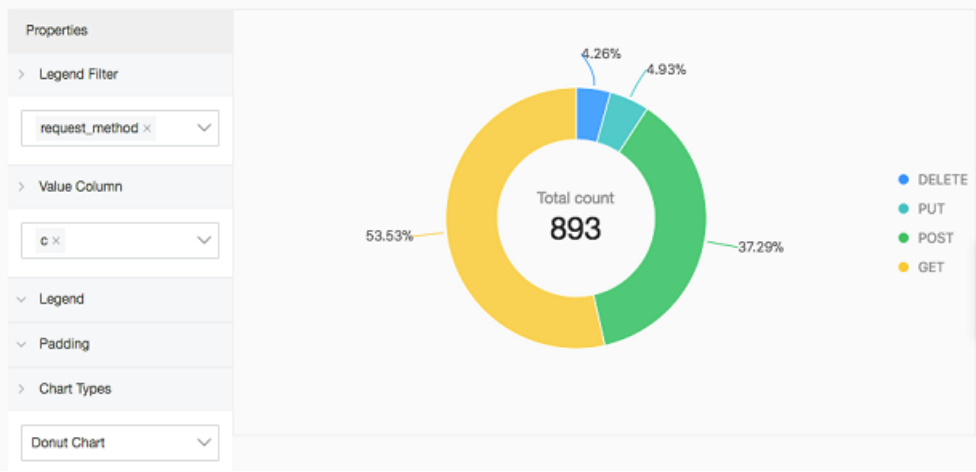


Cycle graph

Analyze the ratio of the access `request_method` :

```
* | select request_method, count(1) as c group by request_method order
 by c limit 10
```

Figure 7-12: Cycle graph

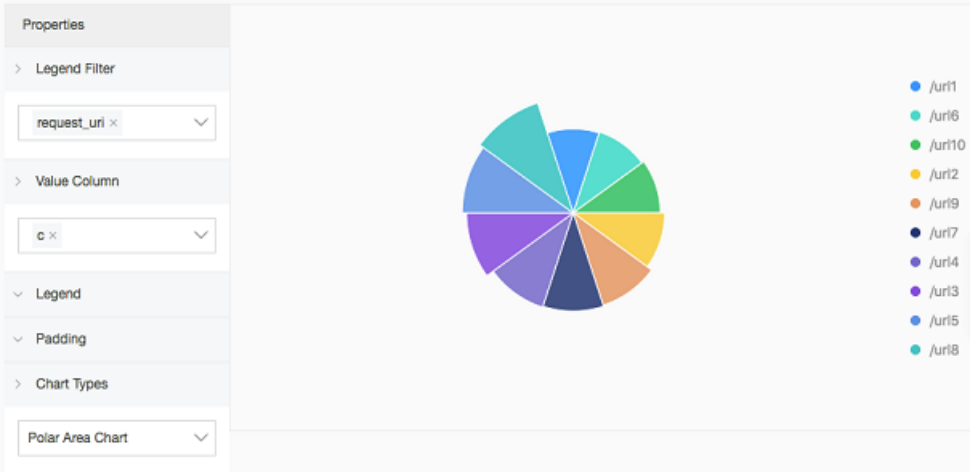


Nightingale rose diagram

Analyze the ratio of the access request_method :

```
* | select request_uri, count(1) as c group by request_uri order by c
```

Figure 7-13: Nightingale rose diagram



7.1.8 Number chart

The number chart, as the easiest and most intuitive display type of data, shows the data on a point clearly and intuitively, and is generally used to indicate the key information on a time point. Log Service number chart automatically normalizes the numeric values. For example, `230000` is processed as `230K`. To customize the numeric format, you must do it in the real-time analysis phase (for more information, see######).

Basic components

- Main text
- Unit (optional)
- Description (optional)

Configuration items

Configuration item	Description
Value column	By default, the first line of data in this column is displayed.
Color	The color in the number chart, including: • Font color • Background Color
Text	The attribute configurations related to the text, including: • Font size (12–100 px) • Unit • Unit font size (12–100 px) • Description • Description font size (12–100 px)

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 number chart ().
3. Configure the graph properties.

Note:

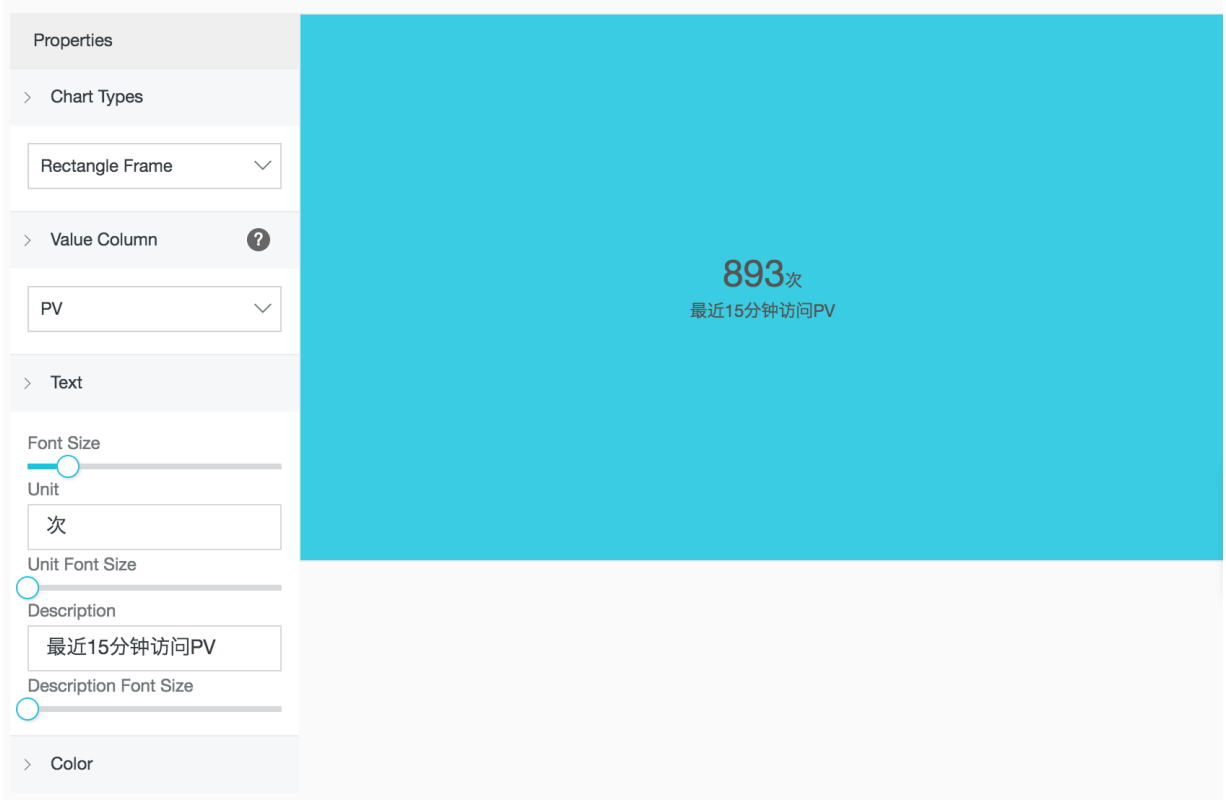
Log Service number chart automatically normalizes the numeric values. For example, `230000` is processed as `230K`. To customize the numeric format, you must do it in the real-time analysis phase (for more information, see######).

Example

Execute the following query analysis statement to view the number of visits.

```
* | select
            count(1)
            as
            C.
```

Figure 7-14: Number chart



7.1.9 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

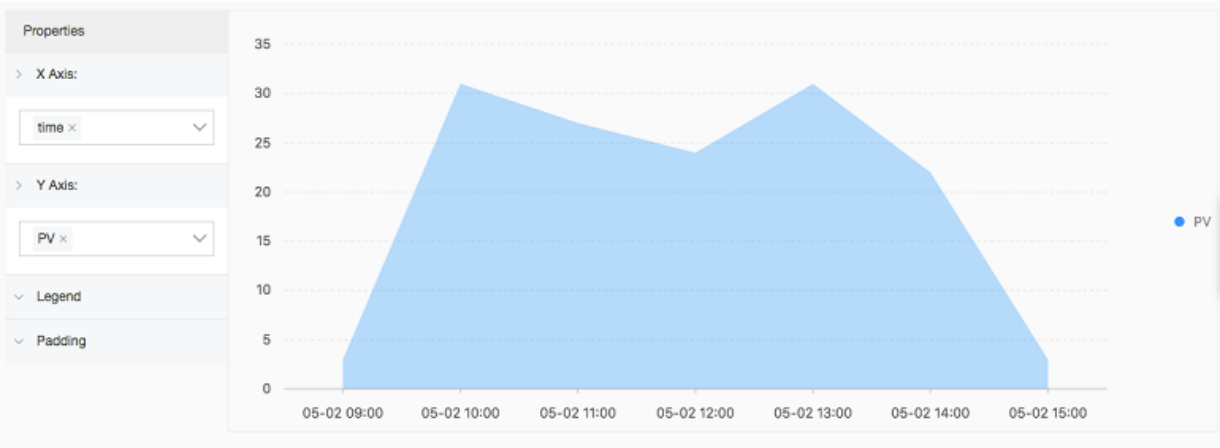
Simple area chart

The PV of IP `42.0.192.0` within the last day:

```
remote_addr: 42.0.192.0 | select date_format(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 7-15: Simple area chart

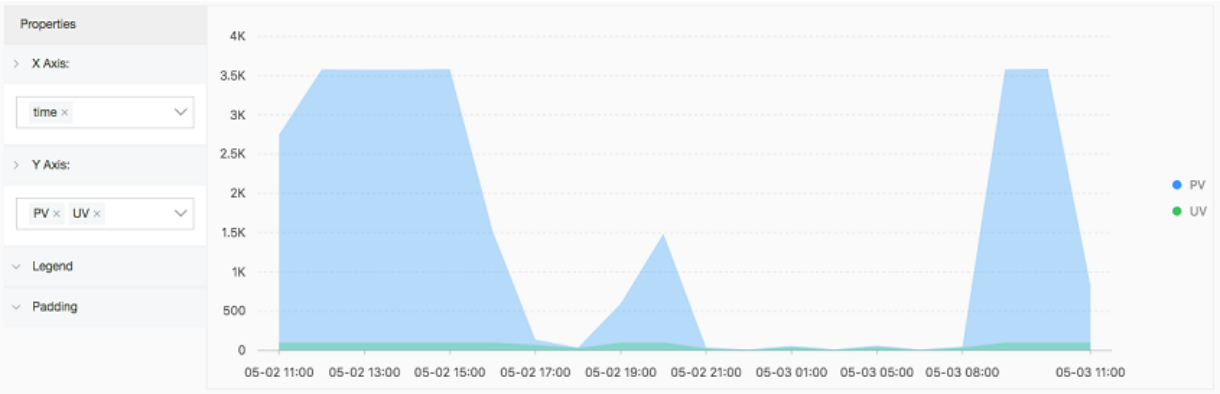


Stacked area chart

```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i')
as time, count(1) as PV, approx_distinct(remote_addr) 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 7-16: Stacked area chart



7.1.10 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.

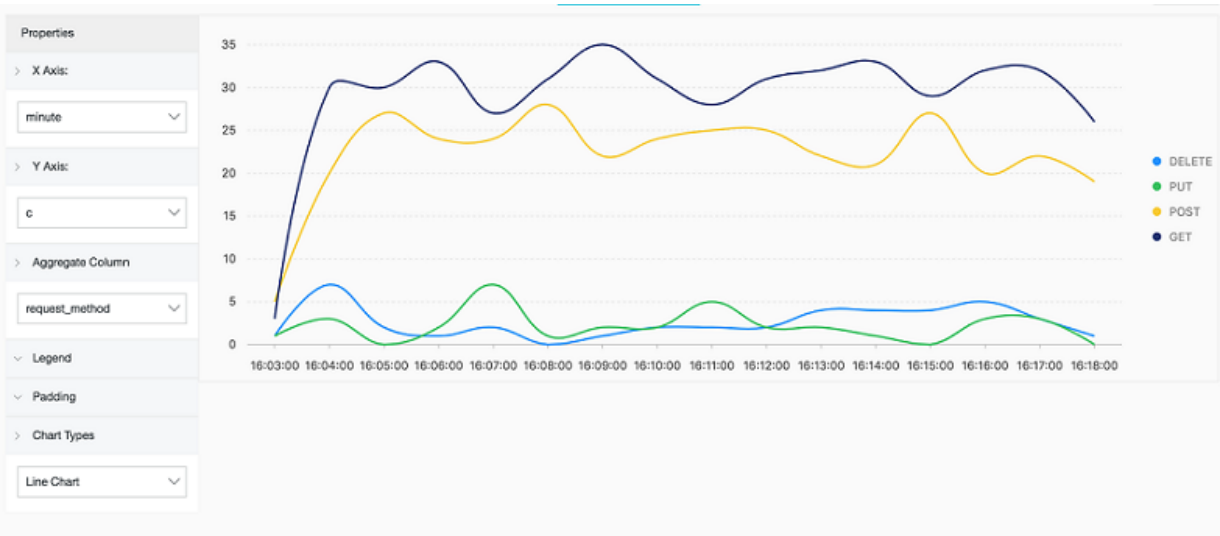
Example

The flow chart is applicable to displaying the three-dimensional relationship (time-type-value).

```
* | select date_format(from_unixtime(__time__ - __time__% 60), '%H
:%i:%S') as minute, count(1) as c, request_method group by minute,
request_method order by minute asc limit 100000
```

Select `minute` as the X Axis, `c` as the Y Axis, and `request_method` as the Aggregate Column.

Figure 7-17: Flow chart



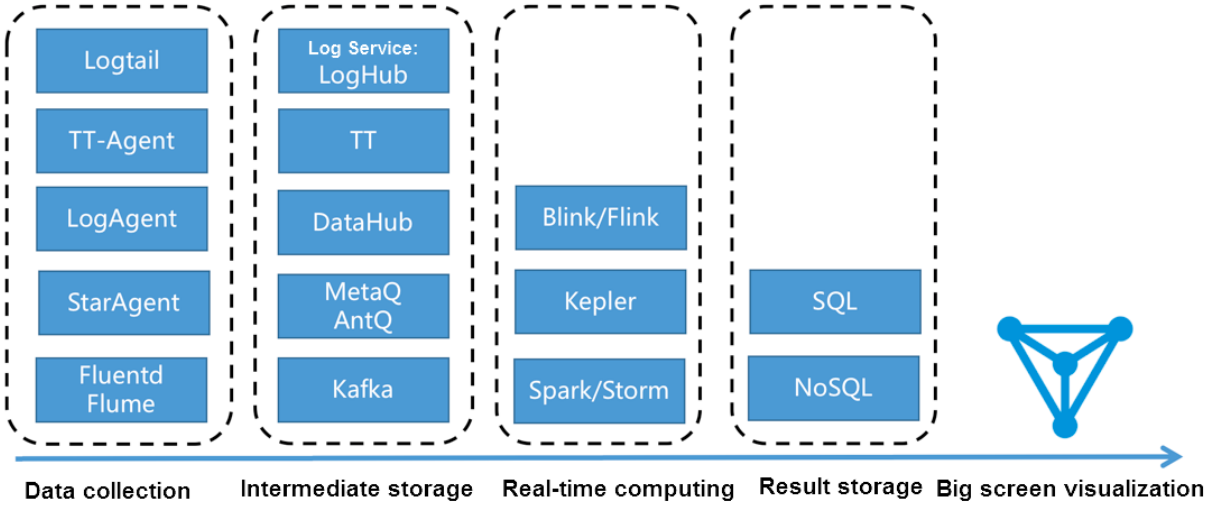
7.3 Interconnect with DataV big screen

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 7-18: 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 7-19: 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.

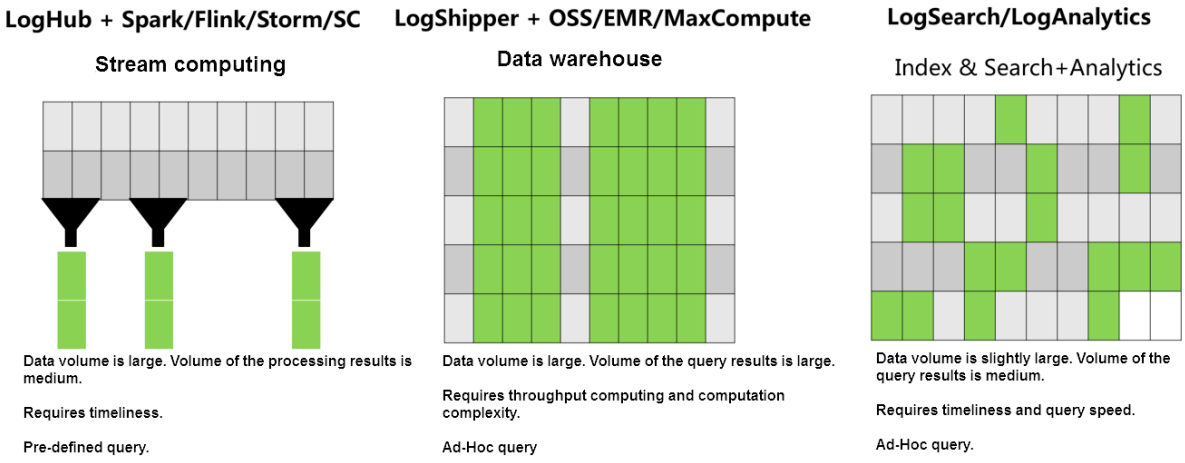
Log Service provides two interconnection methods to collect data in real time. In addition, in log analysis scenarios that has timeliness needs, LogHub data can be indexed in real time. Then, you can use LogSearch/Analytics to directly query and analyze data. This method has the following advantages:

- 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 7-20: 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.

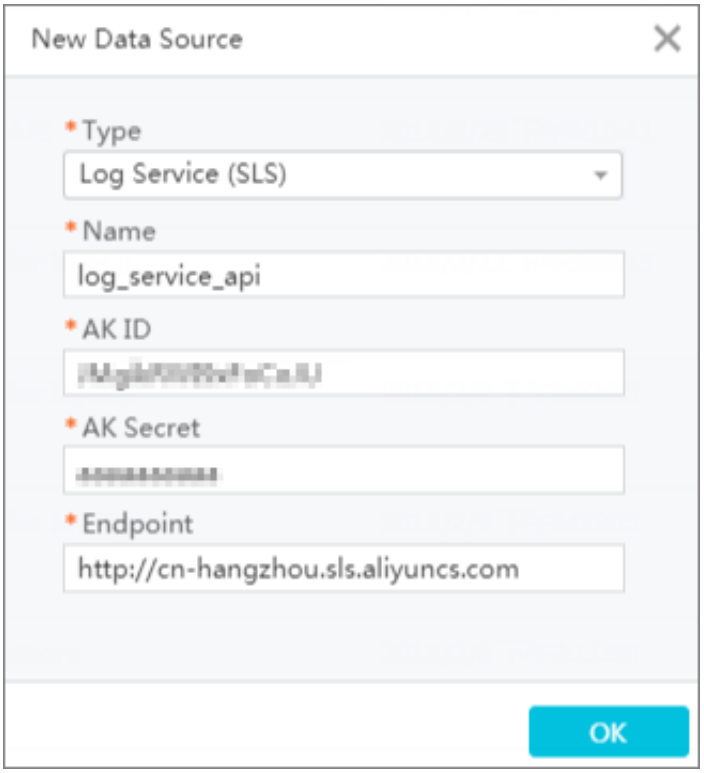
After completing steps 1 and 2, you can view the raw logs on the search page. This document mainly describes how to perform step 3.

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 7-21: New data



Configuration item	Description
Type	Select **Log Service (SLS)**.
Name	Configure a name for the data source.

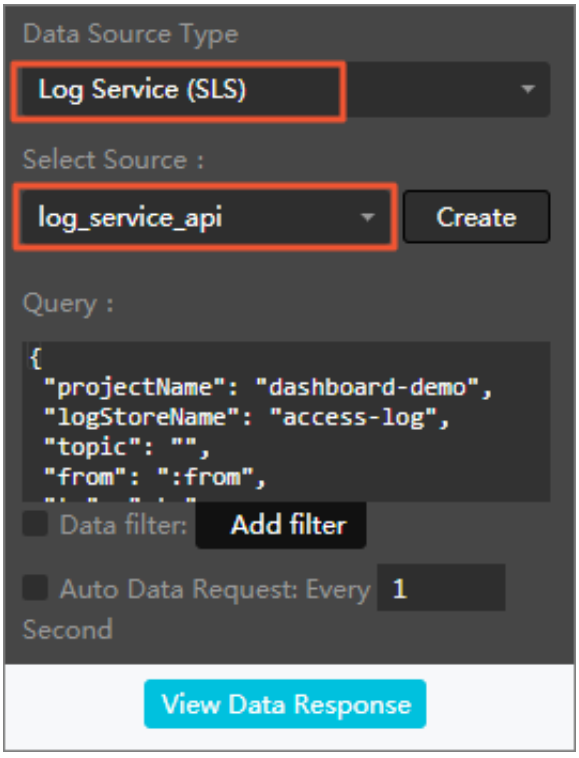
Configuration item	Description
AK ID	The AccessKey ID of the main account, or the AccessKey ID of the sub-account that has the permission to read Log Service.
AK Secret	The AccessKey Secret of the main account, or the AccessKey Secret of the sub-account that has the permission to read Log Service.
Endpoint	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 (SLS)**, select the data source **log_service_api** created in the previous step, and enter the parameters in the **Query** text box.

   Figure 7-22: Data source



An example of the query parameters is as follows and the following table describes the parameters.

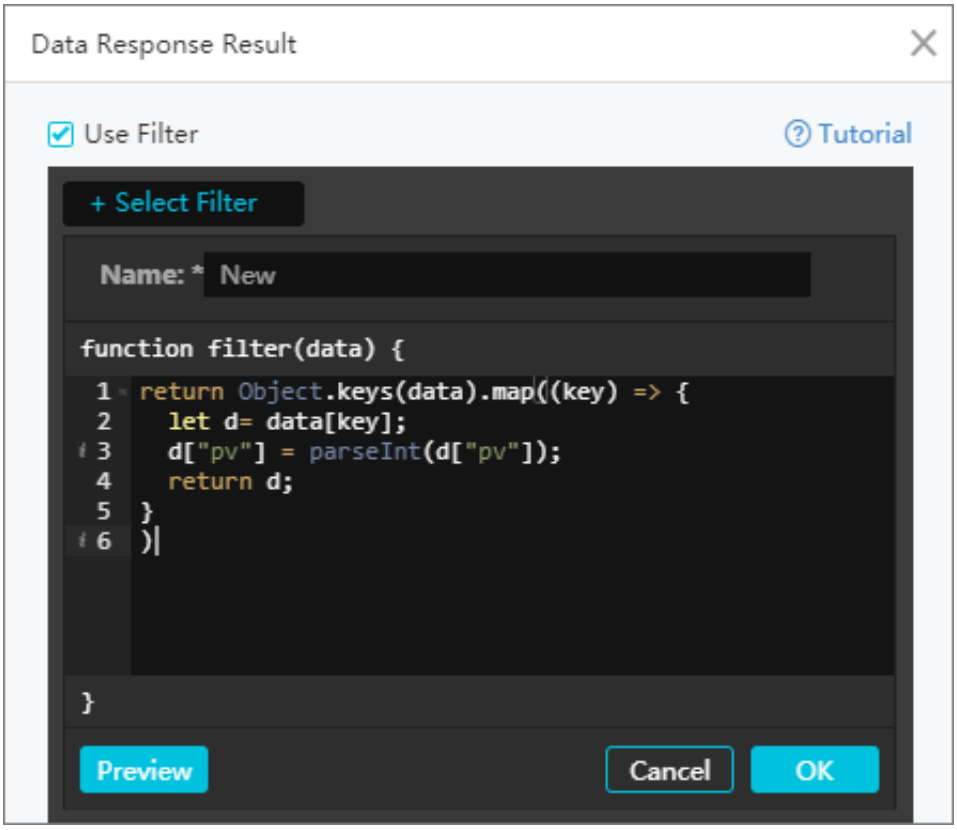
```
{
 "projectName": "dashboard-demo",
 "logStoreName": "access-log",
 "topic": "",
 "from": ":from",
 "to": ":to",
 "query": "*| select approx_distinct(remote_addr) as uv ,count(1) as
 pv , date_format(from_unixtime(date_trunc('hour',__time__) ) ,'%Y/%
m/%d %H:%i:%s') as time group by time order by 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.
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*.

Configuration item	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',__time__)) ,'%m/%d %H:%i:%s') to align the time on the hour, and then convert it to the target format. `date_format(from_unixtime( date_trunc('hour',__time__ ) ) ,'%Y/%m/%d %H:%i:%s')`
line	Enter the default value 100.
offset	Enter the default value 0.

After the configurations, click **View Data Response**.

Figure 7-23: 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;
}
)
```

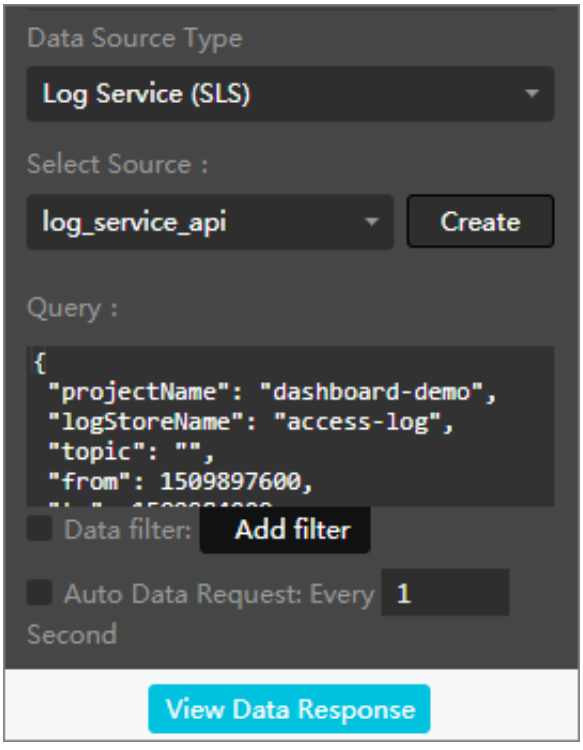
In the filter, convert the result used by y-axis to the int type. In the preceding example, the y-axis indicates the pv. Therefore, the pv column must be converted.

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 7-24: Query text box



Enter the following contents in the Query text box:

```
{
 "projectName": "dashboard-demo",
 "logStoreName": "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;
}
)
```

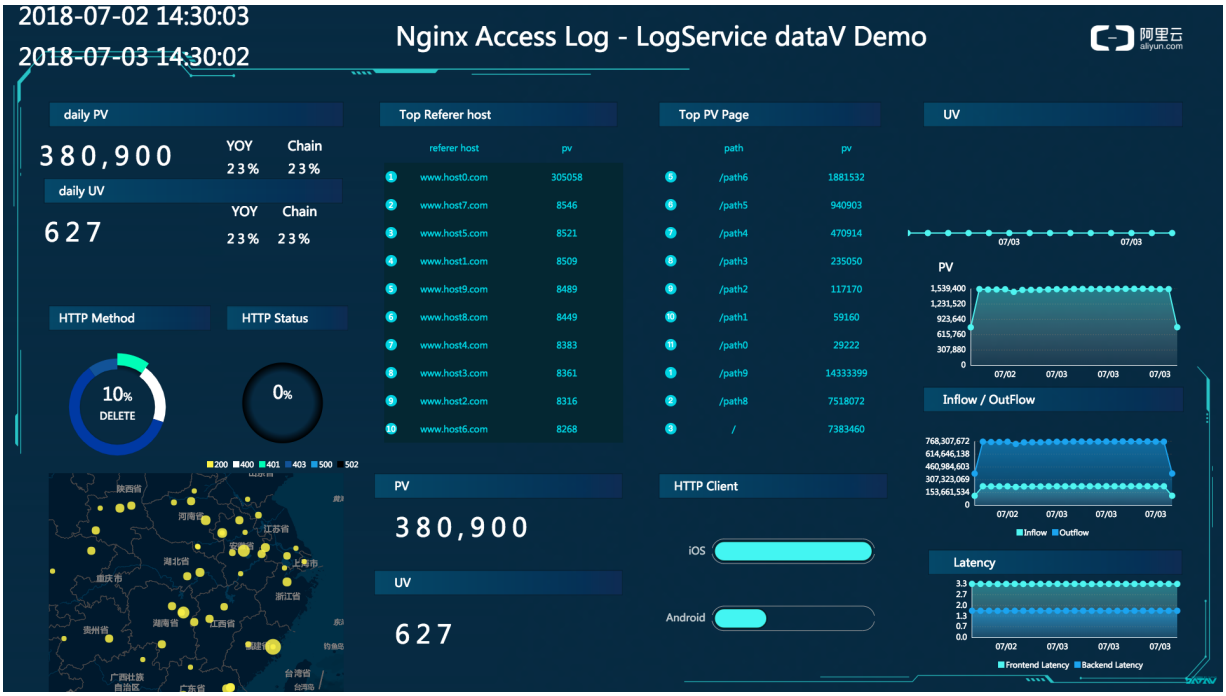
Enter method in the type text box and pv in the value text box for the pie chart.

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 7-25: 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_distinct(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_distinct(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_distinct(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_distinct(
split_part(forward,',',1)) as uv
```

7.5 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 by using 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 your age, gender, battery usage, vehicle ID, operation latency, latitude, lock type, longitude, operation type, operation result, and unlocking method. Data is stored in `Logstore:ebike` of project:`project:trip_demo` . The region where the project resides is cn-hangzhou.

The log sample 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
```

Prerequisite

To use the index and analysis functions of logs, enable the functions for each column of Logstore in the console or by using APIs.

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[]){
           //Modify to your configuration here.
          final String endpoint = "cn-hangzhou-intranet.sls.aliyuncs
.com";//The domain name of Log Service intranet or Virtual Private
Cloud (VPC).
          final String port = "10005"; //The MySQL protocol port of
Log Service.
     final String project = "trip-demo";
     Final string logstore = "ebike ";
     final String accessKeyId = "";
     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 = DriverManager.getConnection("jdbc:mysql://"+endpoint
+":"+port+"/"+project,accessKeyId,accessKey);
                  //Step 3: Create a statement.
         stmt = conn.createStatement();
                  //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_latency from "+logstore+" " +
                 "where __date__ >= '2017-10-11 00:00:00' " +
                 " and __date__ < '2017-10-12 00:00:00'" +
                 " and op ='unlock'";
                  //Step 5: Run query conditions.
         ResultSet rs = stmt.executeQuery(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_latency:");
             // Get avg_latency in results
             System.out.println(rs.getDouble("avg_latency"));
             System.out.println();

         rs.close();
     } catch (ClassNotFoundException e) {
         e.printStackTrace();
     } catch (SQLException e) {
         e.printStackTrace();
     } catch (Exception e) {
         e.printStackTrace();
     } finally {
         if (stmt ! = null) {
             try {
                 stmt.close();
             } catch (SQLException e) {
```

```
                e.printStackTrace();


        if (conn ! = null) {
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
```

Use DavaV to access and display data

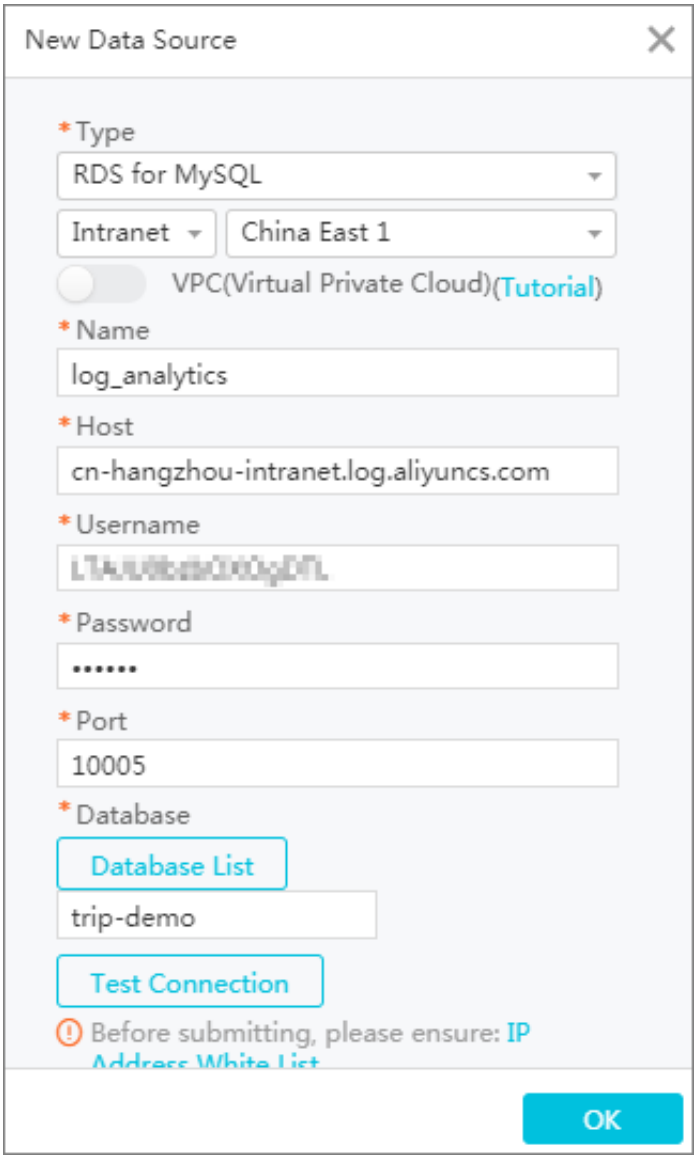
Visualized big screen DataV displays data and connects to Log Service to read log data or display log computation results.

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. Enter 10005 in the Port field and enter the project name in the Database field.

Figure 7-26: Editing data

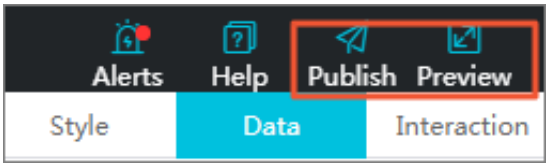


2. Creates a view.

   Select a business template for the view and click any view on the big screen. Modify the data and the data source of the view on the right.

   As shown in the figure, set the data source type to Database, select the data source log_analytics created in the previous step, and enter the SQL statement for query in the SQL field. Enter the mapping between query results and view fields under Field Mapping.

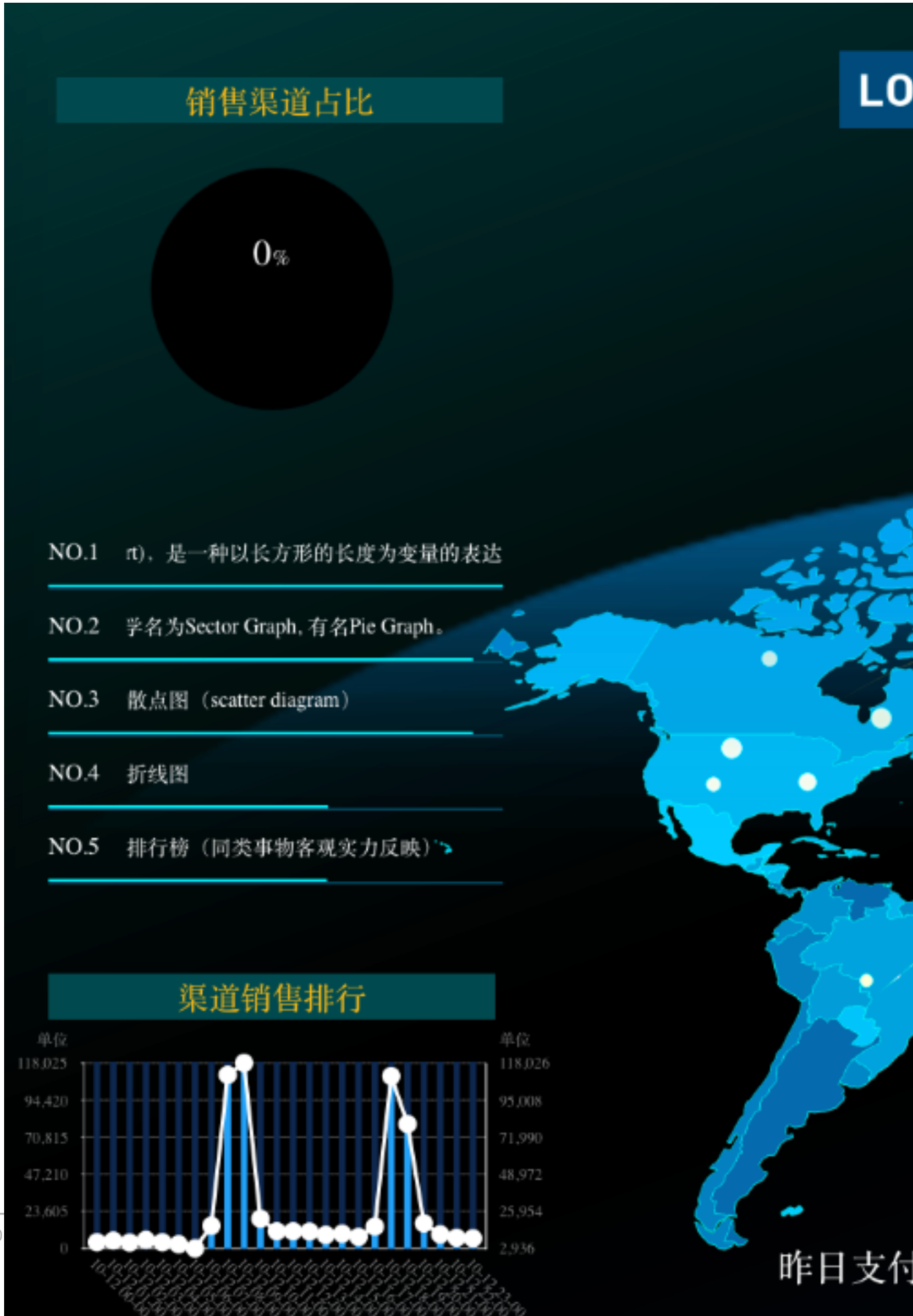
Figure 7-27: Select a database



3. Preview the view and publish.

Click Preview to view the preview effect.

Figure 7-28: Preview



7.6 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 Access Key pair and security token from STS, call the logon service interface to obtain the logon token.

   Request example:

   ```
   http://signin.aliyun.com/federation?Action=GetSigninToken
                       &AccessKeyId=<Temporary Access Key pair returned
    by the STS>
                       &AccessKeySecret=<Temporary secret returned by
   the STS>
                       &SecurityToken=<Security token returned by the
   STS>
   ```

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;>
                            &Destination=<Log Service page to be
accessed. Pages for query and dashboard are supported.>
                            &SigninToken=<Logon token obtained>
```

b) Embedded page.

- A complete page for query and analysis (multiple tags are allowed):

```
https://sls.console.aliyun.com/next/project/<Project name>/
logsearch/<Logstore name>?hideTopbar=true&hideSidebar=true
```

- Query page:

```
https://sls.console.aliyun.com/next/project/<Project name
>/logsearch/<Logstore name>?isShare=true&hideTopbar=true&
hideSidebar=true
```

- Dashboard page:

```
https://sls.console.aliyun.com/next/project/<Project name
>/dashboard/<Dashboard name>?isShare=true&hideTopbar=true&
hideSidebar=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.httpcomponents</
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*

8 Alarm and notification

9 Real-time subscription and consumption

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 Consumer group - Usage

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.

Functions

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.openservices</groupId>
  <artifactId>aliyun-log</artifactId>
  <version>0.6.11</version>
</dependency>
<dependency>
<groupId>com.aliyun.openservices</groupId>
<artifactId>loghub-client-lib</artifactId>
<version>0.6.15</version>
</dependency>
```

main 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_operation_log";
    // Enter the consumer group name according to your actual
situation.
  private static String sConsumerGroup = "consumerGroupX";
    // Enter the AccessKey of data consumption according to your
actual situation.
  private static String sAccessKeyId = "";
  private static String sAccessKey = "";
  public static void main(String []args) throws LogHubClientWorkerEx
ception, InterruptedException

              // 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 group 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].
      LogHubConfig config = new LogHubConfig(sConsumerGroup, "
consumer_1", sEndpoint, sProject, sLogstore, sAccessKeyId, sAccessKey
, LogHubConfig.ConsumePosition.BEGIN_CURSOR);
      ClientWorker worker = new ClientWorker(new SampleLogHubProcesso
rFactory(), config);
        Thread thread = new Thread(worker);
                //The ClientWorker automatically 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
consumption instance. The associated thread is automatically stopped.
       worker.shutdown();
              //Multiple asynchronous tasks are generated when the
ClientWorker 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 SampleLogHubProcessor implements ILogHubProcessor

  private int mShardId;
  // Record the last persistent checkpoint time.
  private long mLastCheckTime = 0;
  public void initialize(int shardId)

      mShardId = shardId;

  // The main logic of data consumption. Catch all the exceptions but
the caught exceptions cannot be thrown.
  public String process(List<LogGroupData> logGroups,
          ILogHubCheckPointTracker checkPointTracker)
```

```
          // 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(LogGroupData logGroup: logGroups){
          FastLogGroup flg = logGroup.GetFastLogGroup();
          System.out.println(String.format("\tcategory\t:\t%s\n\
tsource\t:\t%s\n\ttopic\t:\t%s\n\tmachineUUID\t:\t%s",
                  flg.getCategory(), flg.getSource(), flg.getTopic(),
flg.getMachineUUID()));
          System.out.println("Tags");
          for (int tagIdx = 0; tagIdx < flg.getLogTagsCount(); ++
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.getLogsCount(); ++lIdx) {
              FastLog log = flg.getLogs(lIdx);
              System.out.println("--------\nLog: " + lIdx + ", time: "
 + log.getTime() + ", GetContentCount: " + log.getContentsCount());
              for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx
) {
                  FastLogContent content = log.getContents(cIdx);
                  System.out.println(content.getKey() + "\t:\t" +
content.getValue());



      long curTime = System.currentTimeMillis();
          // 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 - mLastCheckTime > 30 * 1000)

          try

                          //The parameter true indicates to update the
 checkpoint to Log Service immediately. 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.
              checkPointTracker.saveCheckPoint(true);

          catch (LogHubCheckPointException e)

              e.printStackTrace();

          mLastCheckTime = curTime;

      return null;

  // The worker calls this function upon exit. You can perform cleanup
 here.
  public void shutdown(ILogHubCheckPointTracker checkPointTracker)

          //Save the consumption breakpoint to Log Service.
      try {
          checkPointTracker.saveCheckPoint(true);
      } catch (LogHubCheckPointException e) {
          e.printStackTrace();
```

```
class SampleLogHubProcessorFactory implements ILogHubProcessorFactory

  public ILogHubProcessor generatorProcessor()

         // Generate a consumption instance.
      return new SampleLogHubProcessor();
```

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 `ConsumerGroupQuotaExceed` 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.openservices.loghub.
client.LogHubConsumer.sampleLogError(LogHubConsumer.java:159)
com.aliyun.openservices.log.exception.LogException: Invalid loggroup
count, (0,1000]
```

See the following log4j.properties configuration for reference:

```
log4j.rootLogger = info,stdout
log4j.appender.stdout = org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target = System.out
log4j.appender.stdout.layout = org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern = [%-5p] %d{yyyy-MM-dd
HH:mm:ss,SSS} method:%l%n%m%n
```

9.4 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

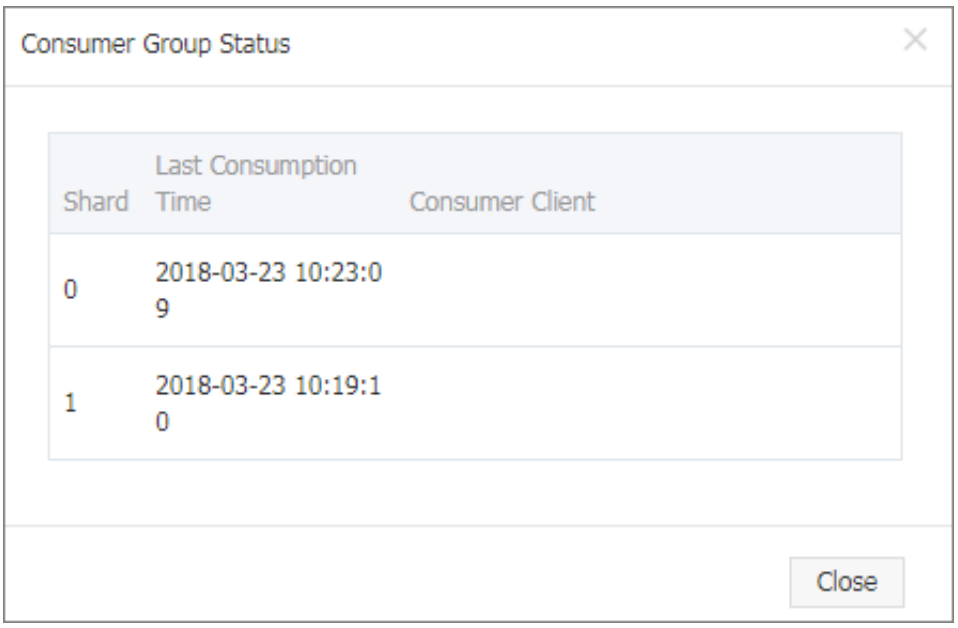
1. Log on to the Log Service console.
2. On the Project List page, click the project name.
3. Click **LogHub - Consume** > **Consumer**in the left-side navigation pane.

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.openservices.log.Client;
import com.aliyun.openservices.log.common.Consts.CursorMode;
import com.aliyun.openservices.log.common.ConsumerGroup;
import com.aliyun.openservices.log.common.ConsumerGroupShardCheckPoint
;
import com.aliyun.openservices.log.exception.LogException;
public class ConsumerGroupTest {
    static String endpoint = "";
    static String project = "";
    static String logstore = "";
    static String accesskeyId = "";
    static String accesskey = "";
    public static void main(String[] args) throws LogException {
        Client client = new Client(endpoint, accesskeyId, accesskey);
                //Retrieve all consumer groups in this Logstore. If no
 consumer group exists, the consumerGroups length is 0.
        ArrayList<ConsumerGroup> consumerGroups;
        try{
            consumerGroups = client.ListConsumerGroup(project,
logstore). GetConsumerGroups();

        catch(LogException e){
            if(e.GetErrorCode() == "LogStoreNotExist")
                System.out.println("this logstore does not have any
consumer group");
            else{
                //internal server error branch

            return;

        for(ConsumerGroup c: consumerGroups){
                        //Print consumer group properties, including
 names, heartbeat timeout, and whether or not the consumption is in
order.
                       System.out.println("Name:" + c.getConsume
rGroupName());
                        System.out.println("Heartbeat timeout:" + c.
getTimeout());
                        System.out.println("Consumption in order" + c.
isInOrder());
            for(ConsumerGroupShardCheckPoint cp: client.GetCheckPoint(
project, logstore, c.getConsumerGroupName()). GetCheckPoints()){
                System.out.println("shard: " + cp.getShard());
                // Please format, this time returns the exact time to
milliseconds, the length of the integer
                                //Format the returned time to be
precise to milliseconds in the long integer.
                                System.out.println("Last data
consumption time:" + cp.getUpdateTime());
                String consumerPrg = "";
                if(cp.getCheckPoint().isEmpty())
                                    consumerPrg = "Consumption not
 started";
```

```
                else{
                                          //UNIX timestamp. Measured in
seconds. Format the value upon output.
                    try{
                        int prg = client.GetPrevCursorTime(project,
logstore, cp.getShard(), cp.getCheckPoint()). GetCursorTime();
                        consumerPrg = "" + prg;

                    catch(LogException e){
                        if(e.GetErrorCode() == "InvalidCursor")
                                                              consumerPr
g = "Invalid. The previous consumption time has exceeded the data
lifecycle in the Logstore.";
                        else{
                            //internal server error
                            throw e;



                                System.out.println("Consumption
progress:" + consumerPrg);
                String endCursor = client.GetCursor(project, logstore
, cp.getShard(), CursorMode.END). GetCursor();
                int endPrg = 0;
                try{
                    endPrg = client.GetPrevCursorTime(project,
logstore, cp.getShard(), endCursor). GetCursorTime();

                catch(LogException 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);
```

9.6 Use Fuction Compute to cosume LogHub Logs

9.6.1 Development guide for ETL function

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 introduces 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 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. ####
projectName	The project name.
logstoreName	The Logstore name.
Shardid	A specific shard in the Logstore.
beginCursor	The shard location where to start consuming data.
endCursor	The shard location where to finish consuming data.

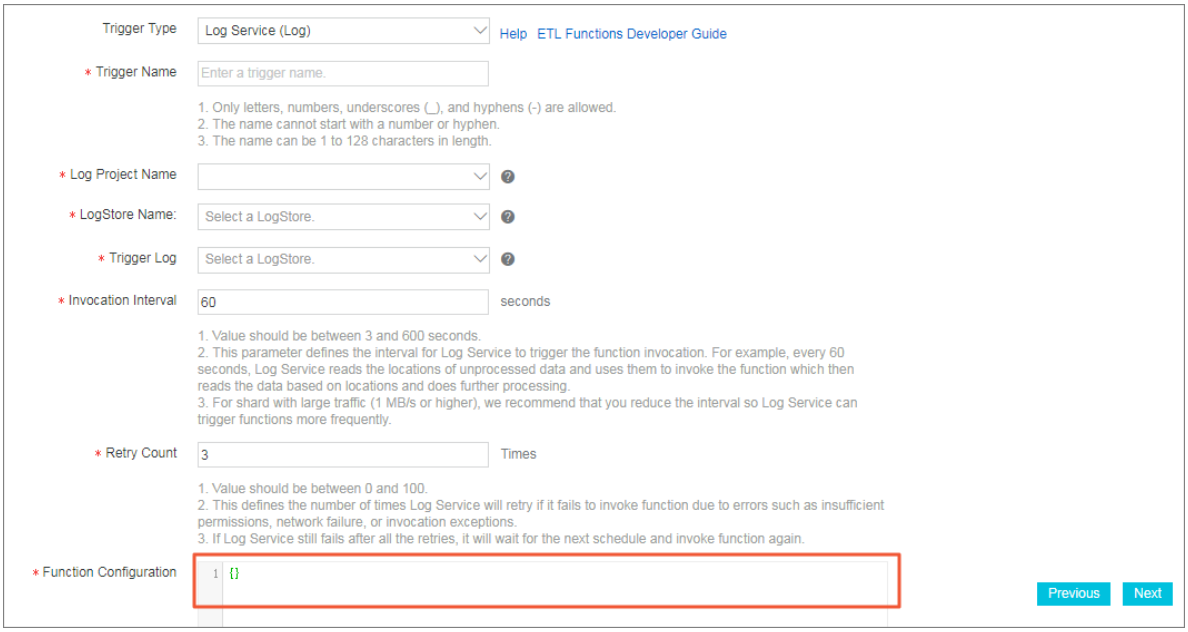
  Note:
  The [beginCursor, endCursor) of a shard is a left-closed and right-opened interval.

- **parameter field**

  This JSON Object field is set when you create the ETL job (Log Service trigger of Function Compute). When the user-defined function is running, this field is parsed to obtain the operating 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 9-4: Function configuration



Example of function event

```
"source": {
    "endpoint": "http://cn-shanghai-intranet.log.aliyuncs.com",
    "projectName": "fc-1584293594287572",
    "logstoreName": "demo",
    "shardId": 0,
    "beginCursor": "MTUwNTM5MDI3NTY1ODcwNzU2Ng==",
    "endCursor": "MTUwNTM5MDI3NTY1ODcwNzU2OA=="

"parameter": {


"jobName": "fedad35f51a2a97b466da57fd71f315f539d2234",
"TaskId": "9bc06c96-e364-4f41-85eb-b6e579214ae4 ",
"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 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 `UserDefinedFunctionParameter.java` to parse the parameter field in the function event.
- Use the processData API of `UserDefinedFunction.java` to customize the data business logic in the function.
- Replace `UserDefinedFunction` with a name that properly describes your function.

processData method implementation

In processData, you must consume, process, and ship a batch of data as per your 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.

Precautions:

- 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, Function Compute service generates the ETL error log and sends an alarm email or SMS to the system administrator. When creating a trigger, set the trigger log Logstore and enable the query and index functions 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**

  ETL process logs record the key points and errors for each step in the ETL execution process , including the start time, end time, initialization completion, and module error information in a step. You can use the ETL process logs to detect the ETL operation situation all the time and troubleshoot the error in time.

  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 .

9.7 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.

It consists of two parts: consumer and producer.The consumer reads data from Log Service, 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.openservices</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.openservices</groupId>
            <artifactId>aliyun-log</artifactId>
            <version>0.6.10</version>
 </dependency>
<dependency>
            <groupId>com.aliyun.openservices</groupId>
            <artifactId>log-loghub-producer</artifactId>
            <version>0.1.8</version>
</dependency>
```

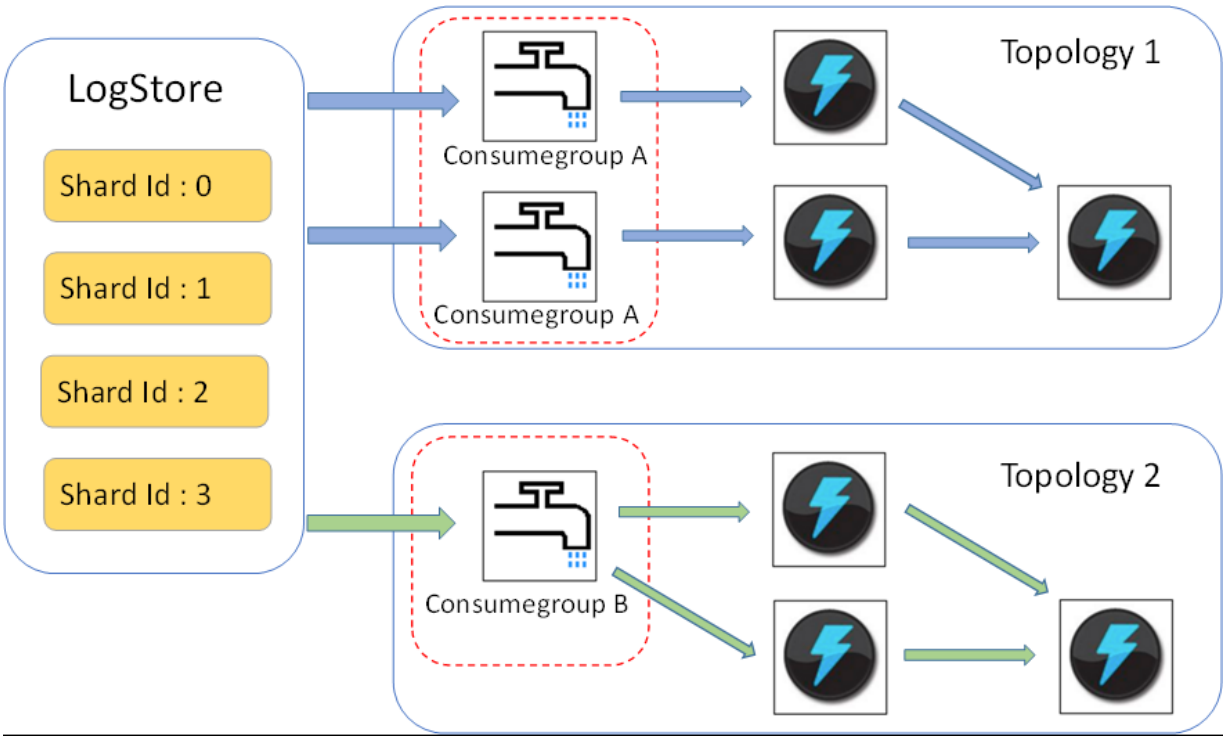
9.8 Use Storm to consume LogHub logs

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 9-5: 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 do not affect each other.
- Each topology is identified by a unique LogHub consumer group name. Spouts in the same topology use the *LogHub client lib* 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_group_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 underscores (_), 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 = ""; // The domain name used to
access Log Service.
                String access_id = ""; // Your AccessKey.
                String access_key = "";
                // Configurations required for building a LogHub
Storm spout.
        Loghubspoutconfig Config = new loghubspoutconfig (conumser_g
roup_name,
                endpoint, project, logstore, access_id,
                access_key, LogHubCursorPosition.END_CURSOR);
        Topologybuilder builder = new topologybuilder ();
        // 构建 loghub storm spout
        Loghubspout 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()).shuffleGrouping
("spout");
        Config conf = new Config();
        conf.setDebug(false);
        Conf. setmaxspoutpending (1 );
                // The serialization method LogGroupDataSerializ
Serializer of LogGroupData must be configured explicitly when Kryo
is used for data serialization and deserialization.
        Config.registerSerialization(conf, LogGroupData.class,
LogGroupDataSerializSerializer.class);
        if (mode.equals("Local")) {
            logger.info("Local mode...")
            Localcluster cluster = new localcluster ();
            cluster.submitTopology("test-jstorm-spout", conf,
builder.createTopology());
            Try {
                Thread.sleep(6000 * 1000); //waiting for several
minutes
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();

            cluster.killTopology("test-jstorm-spout");
            Cluster. Shutdown ();
        } else if (mode.equals("Remote")) {
            logger.info("Remote mode...");
            conf.setNumWorkers(2);
```

```
            Try {
                StormSubmitter.submitTopology("stt-jstorm-spout-4",
conf, builder.createTopology());
            } catch (AlreadyAliveException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            } catch (InvalidTopologyException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();

        } 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 BaseRichBolt {
    private static final long serialVersionUID = 4752656887
774402264L;
    private static final Logger logger = Logger.getLogger(BaseBasicB
olt.class);
    private OutputCollector mCollector;
    @Override
    public void prepare(@SuppressWarnings("rawtypes") Map stormConf
, TopologyContext context,
            OutputCollector collector) {
        mCollector = collector;

    @Override
    public void execute(Tuple tuple) {
        String shardId = (String) tuple
                .getValueByField(LogHubSpout.FIELD_SHARD_ID);
        @SuppressWarnings("unchecked")
        List<LogGroupData> logGroupDatas = (ArrayList<LogGroupData
>) tuple.getValueByField(LogHubSpout.FIELD_LOGGROUPS);
        for (LogGroupData groupData : logGroupDatas) {
                        // Each LogGroup consists of one or more
logs.
            LogGroup logGroup = groupData.GetLogGroup();
            for (Log log : logGroup.getLogsList()) {
                StringBuilder sb = new StringBuilder();
                // Each log has a time field and multiple key: value
 pairs,
                                    // Each log has a time field and
multiple key-value pairs.
                sb.append("LogTime:").append(log_time);
                for (Content content : log.getContentsList()) {
                    sb.append("\t").append(content.getKey()).append
(":")
                                .append(content.getValue());

                logger.info(sb.toString());


                // The 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.
        //Therefore, bolts must call ACK for confirmation.
        mCollector.ack(tuple);
```

```
    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
        //do nothing
```

Maven

Use the following code for versions earlier than Storm 1.0 (for example, 0.9.6):

```
<dependency>
  <groupId>com.aliyun.openservices</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.openservices</groupId>
  <artifactId>loghub-storm-1.0-spout</artifactId>
  <version>0.1.2</version>
</dependency>
```

9.9 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*.

9.10 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.

Scenarios

See Connect to 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)
  - *#####OSS*
  - Formats in OSS can be processed by using Hive. E-MapReduce is recommended.
- Table Store (NoSQL data storage service)
  - *Procedure*
- Maxcompute (large data computing services):
  - Delivery via dataworks Data Integration-operation steps

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

efore 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*.
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/aliyunlogdefaultrole`.

Configuration item	Description	Value range
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.
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

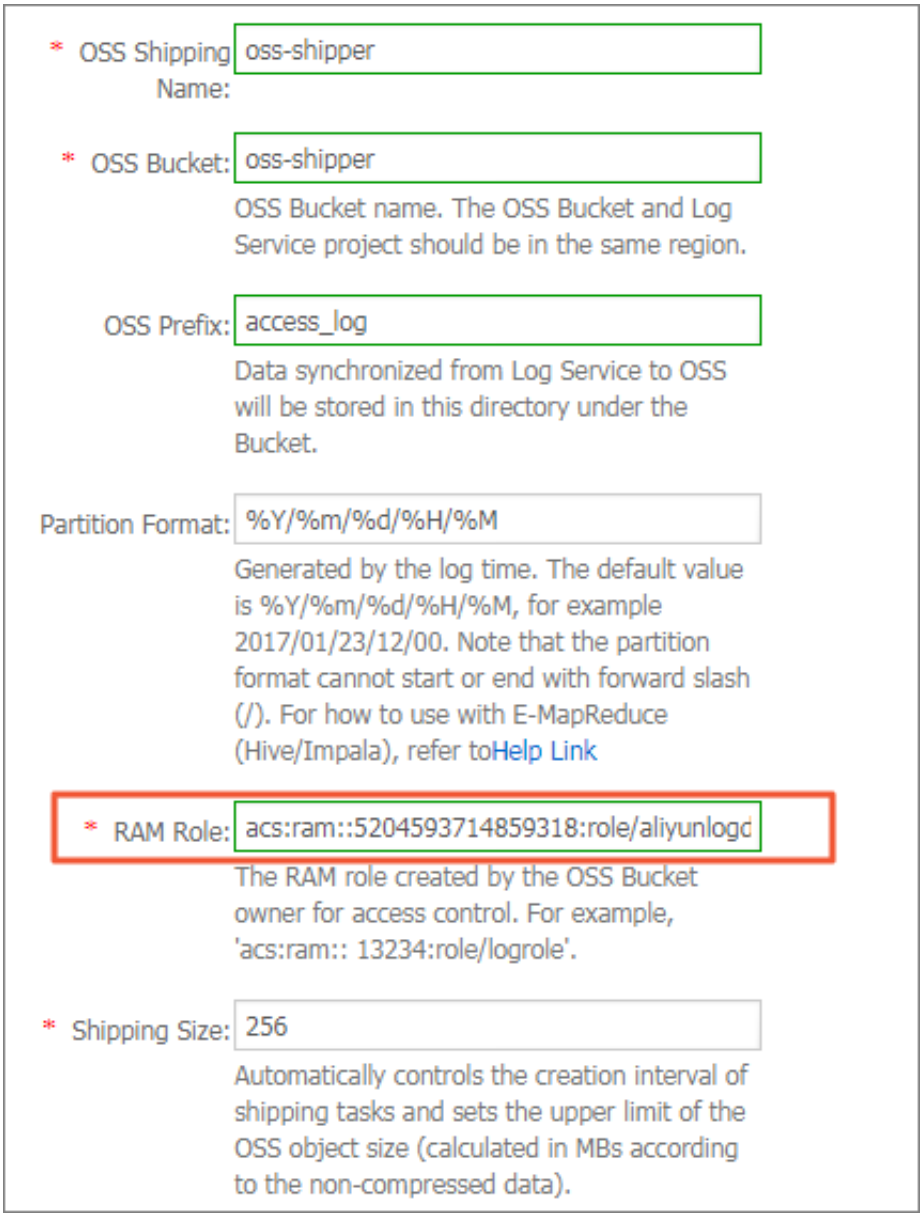
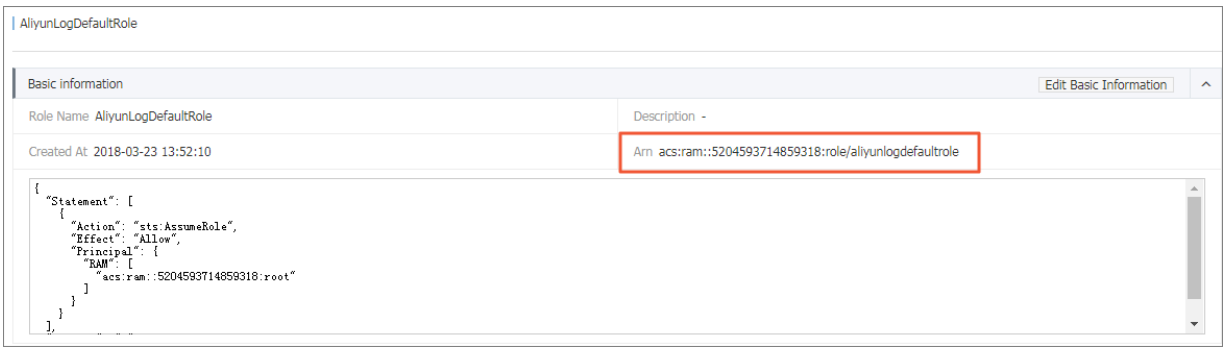


Figure 10-2: Role arn



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_RANDOM-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`
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`

OSS Bucket	OSS Prefix	Partition format	OSS file path
			3351525351_2850008
test-bucket	log_ship_oss_example	%Y%m%d%H	oss://test-bucket/log_ship_oss_example/2017012019_1484913043351525351_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_RANDOM-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 *#####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__":"10.170.148.237","
ip":"10.200.98.220","time":"26/Jan/2016:19:54:02 +0800","url":"POST
             /PutData? Category=YunOsAccountOpLog&AccessKeyId=
U0UjpekFQOVJW45A&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&
Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D
             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.

Decompress the file by using C++ Lib

Download Lib from *Snappy official website* and use the Snappy.Uncompress method to decompress the .snappy file.

Decompress the file by using Java Lib

Download Lib from*xerial snappy-java* Use Snappy.Uncompress or Snappy.SnappyInputStream to decompress the .snappy file. SnappyFramedInputStream is 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 fixed in version 1.1.2.6.

Snappy.Uncompress

```
String fileName = "C:\\Downloads\\36_1474212963188600684_4451886.
snappy";
RandomAccessFile randomFile = new RandomAccessFile(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[] uncompressed = Snappy.uncompress(bytes);
String result = new String(uncompressed, "UTF-8");
System.out.println(result);
```

Snappy.SnappyInputStream

```
String fileName = "C:\\Downloads\\36_1474212963188600684_4451886.
snappy";
Snappyinputstream SIS = new snappyinputstream (New FileInputStream (
fileName ));
byte[] buffer = new byte[4096];
int len = 0;
while ((len = sis.read(buffer)) ! = -1) {
    System.out.println(new String(buffer, 0, len));
```

Linux Decompression tool in Linux environment

For Linux environment, a tool used to decompress .snappy file is provided. Click to download the *snappy_tool*.

```
./snappy_tool 03_1453457006548078722_44148.snappy 03_14534570065480787
22_44148
compressed.size: 2217186
snappy::Uncompress return: 1
uncompressed.size: 25223660
```

10.2.3 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 *# ####OSS*.

Configure Parquet storage fields

Data types

Parquet supports storage in six types: string, boolean, int32, int64, float, and double.

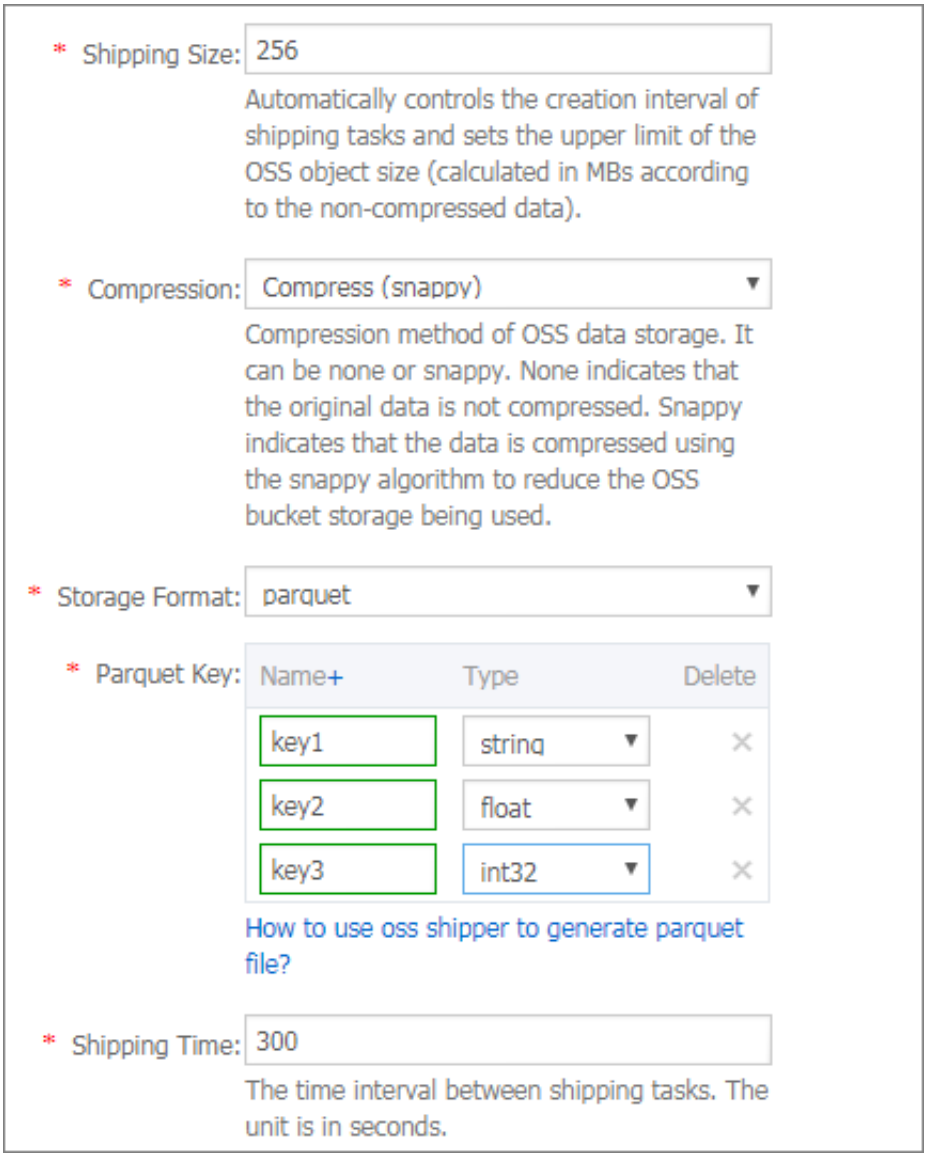
Log Service data is converted from strings to the target Parquet type during log shipping. If data fails to be converted to a non-string type, the corresponding column is set to null.

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 10-3: Field Configuration



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 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
532136470439_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 maid head-N 200_1490803532136470439_124353.snappy.parquet
__time__ = 1490803230
ip = 10.200.98.220
__source__ = 11.164.232.106
method = POST
__topic__ =
seq = 1667821.0
status = 200
time = 30/Mar/2017:00:00:30 +0800
url = /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0UjpekFQO
VJW45A&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&
Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1
__time__ = 1490803230
ip = 10.200.98.220
__source__ = 11.164.232.106
method = POST
__topic__ =
seq = 1667822.0
status = 200
time = 30/Mar/2017:00:00:30 +0800
url = /PutData? Category=YunOsAccountOpLog&AccessKeyId=U0UjpekFQO
VJW45A&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&
Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1
```

For more operation instructions, run the java -jar parquet-tools-1.6.0rc3-SNAPSHOT.jar -h command.

10.2.4 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*.

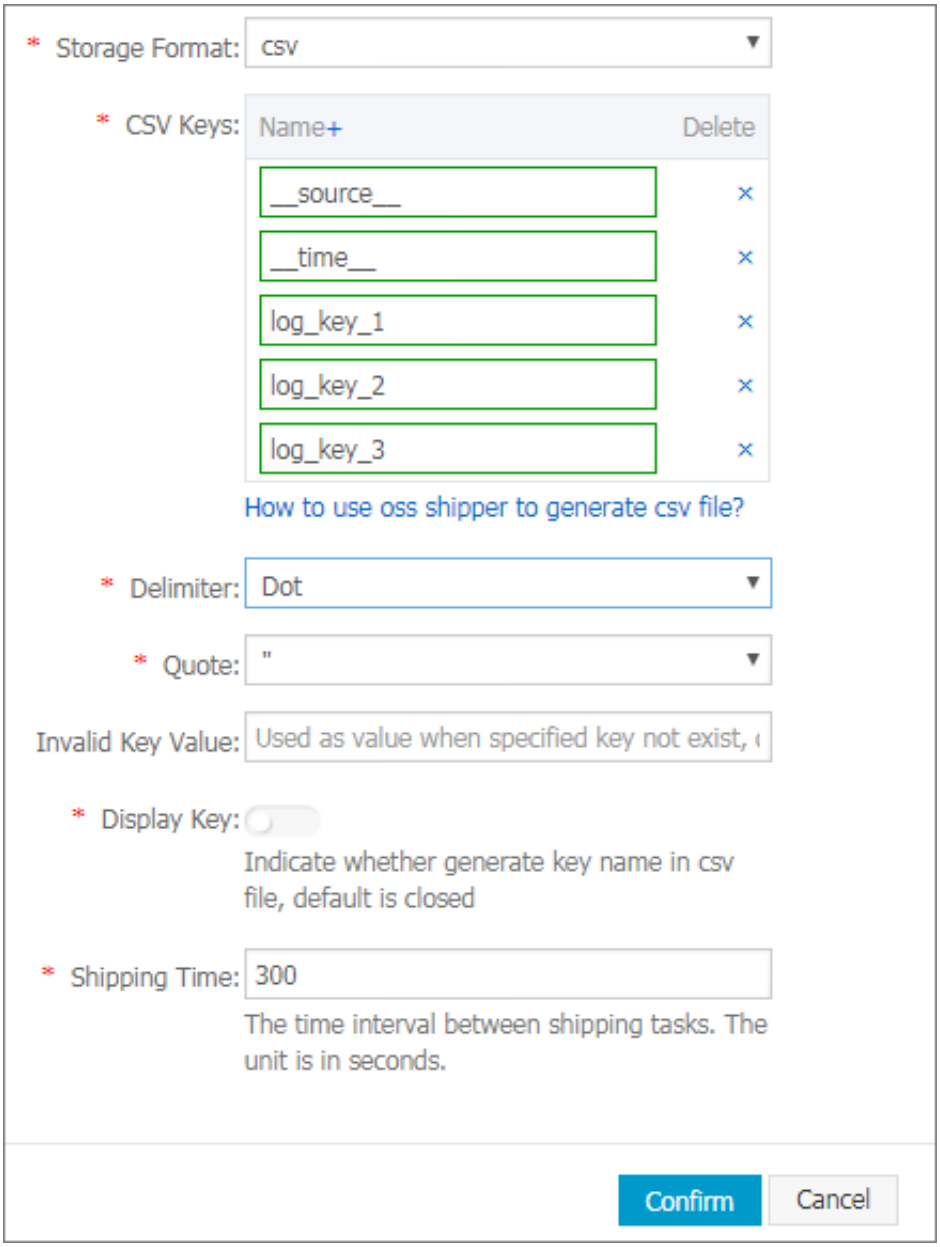
Configure CSV storage fields

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 10-4: Configuration item



Configuration item

Configuration item	Value	Note
Delimiter	character	A one-character string used to separate different fields.
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*.

10.2.5 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-account 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_ID@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 1654218965343050, and modified description is as follows:

```
"Statement": [
   "Action": "sts:AssumeRole",
```

```
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "1654218965343050@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 ######.

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. In the RAM console, the main account A grants AliyunRAMFullAccess permission to the sub-account a_1.
   b. On the User Management page, click Authorization on the right side of the sub-account a_1.

      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/aliyunlogdefaultrole"


"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 bucket owner, that is, the RAM role AliyunLogDefaultRole created by account B.

10.4 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 correspond ing 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 pop-up 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 *Set alarms*.

Procedure

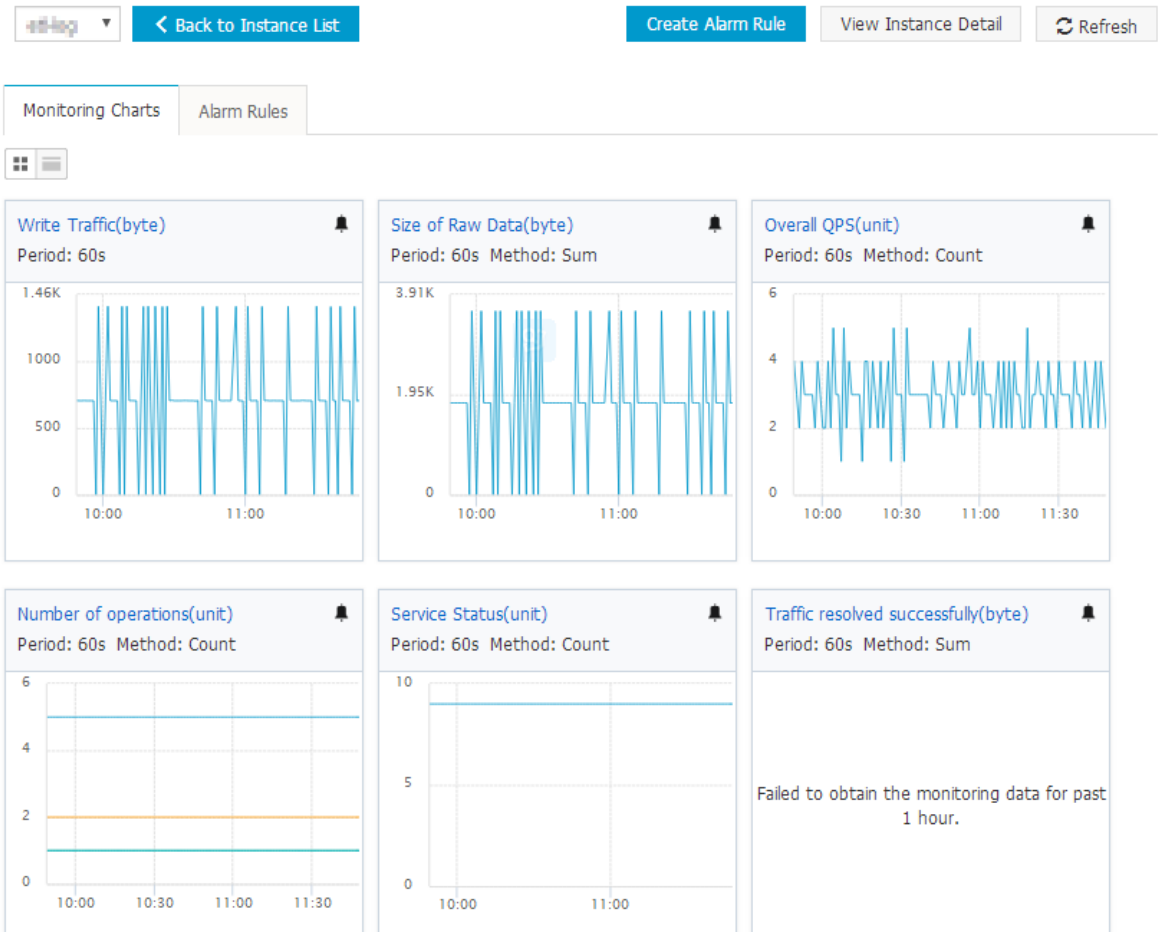
Note:
You must authorize the sub-accounts before using them to configure the cloud monitoring.

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 Service**in 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 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 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 x**CloudMonitor console** > **Log Service**click **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**.

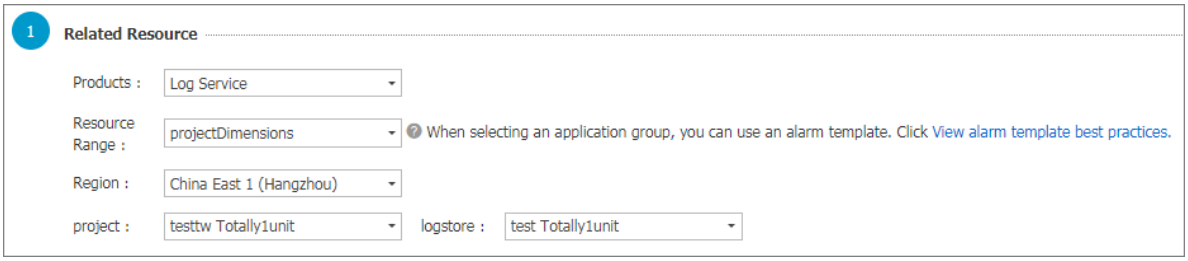
      You can select **All Resources**, **Application Group**, or **projectDimensions**.

      - **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 **Project**and **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*.

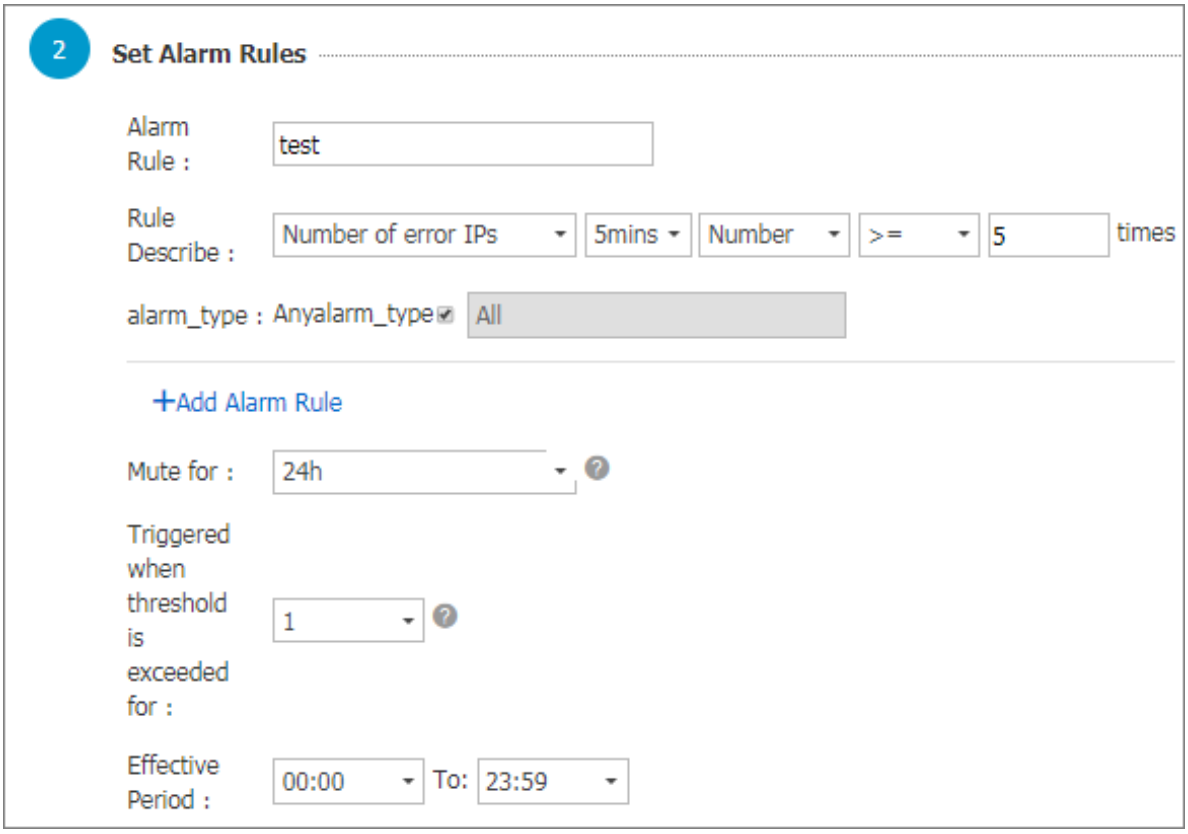
c. Select the**alarm_type**. By default, **Any alarm_type** is selected.

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**.

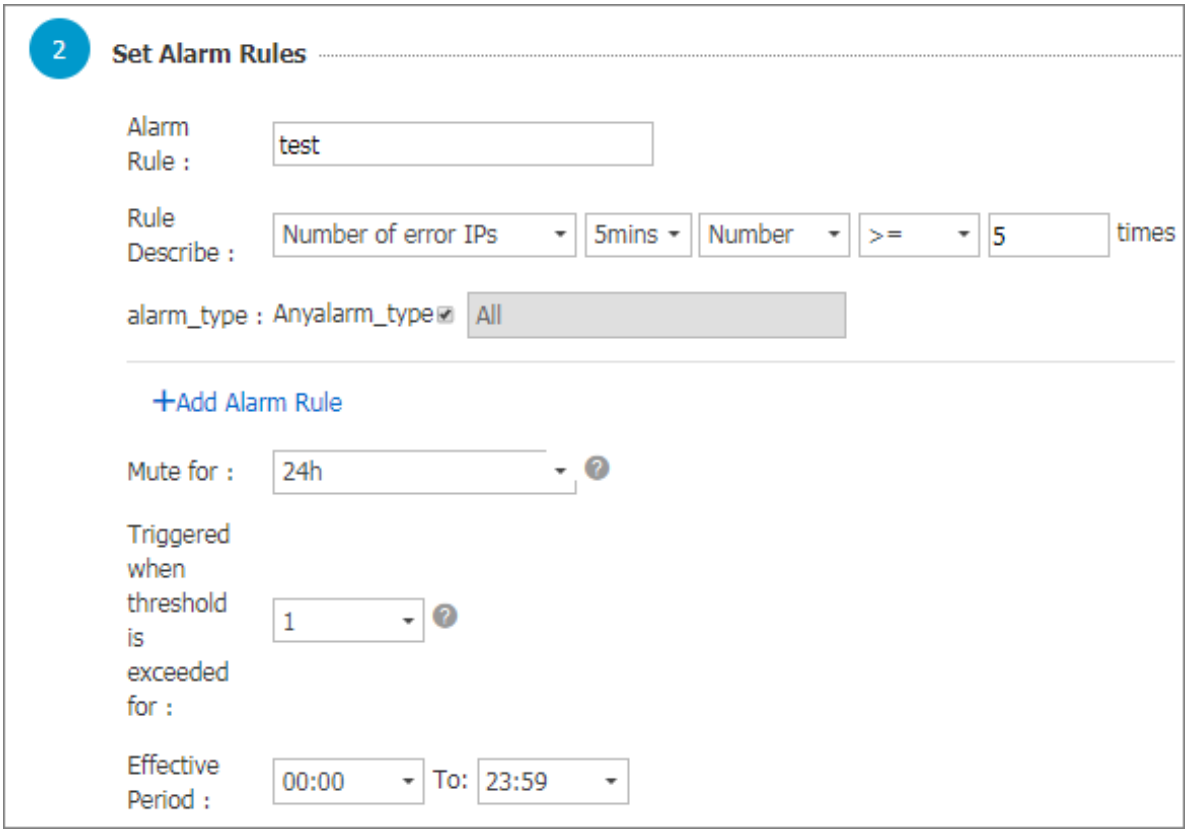
   a. **Notification contact**. Send a notification in the **contact group level**.

   b. **Alarm level**. Select **Warning**or **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

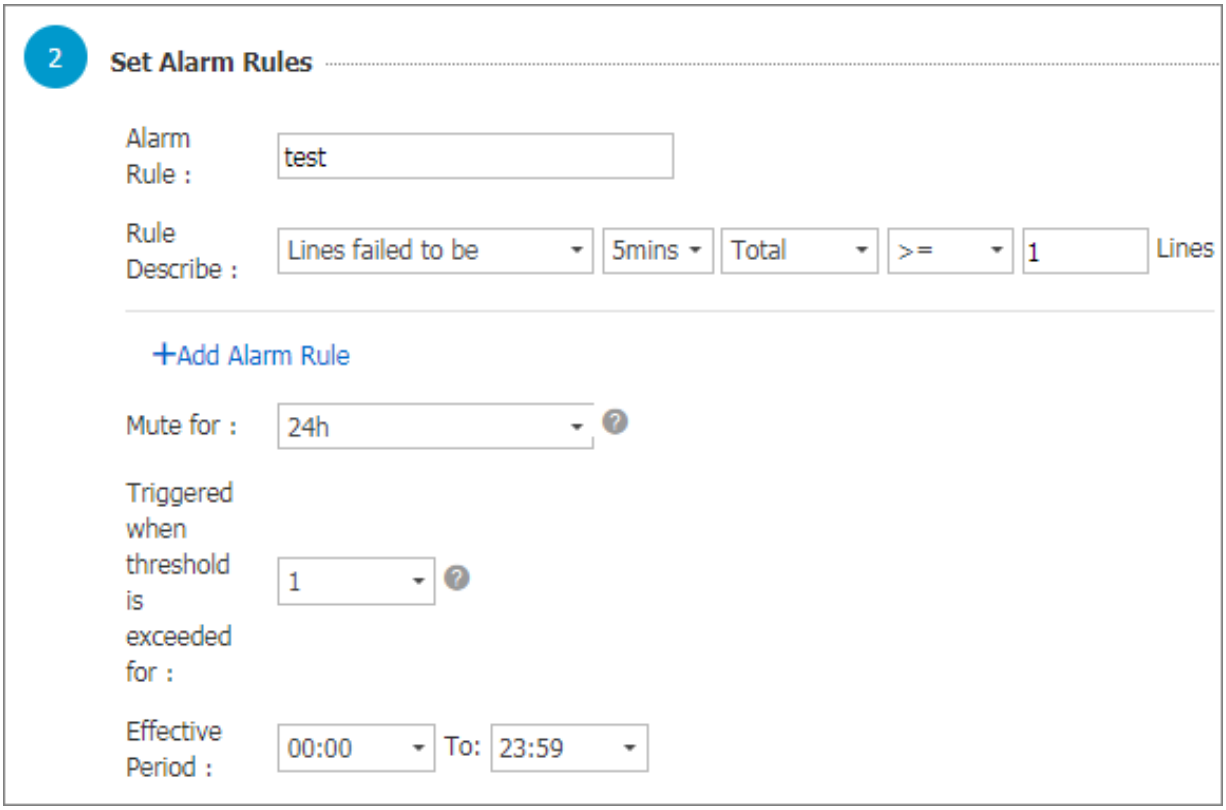
Errors may occur because of incorrect configurations when Logtail is running. For example, some log formats do not match or a log file is repetitively collected. For more information, see Basic questions of Logtail. To find such errors in time, you can monitor the metrics such as lines failed to be resolved 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

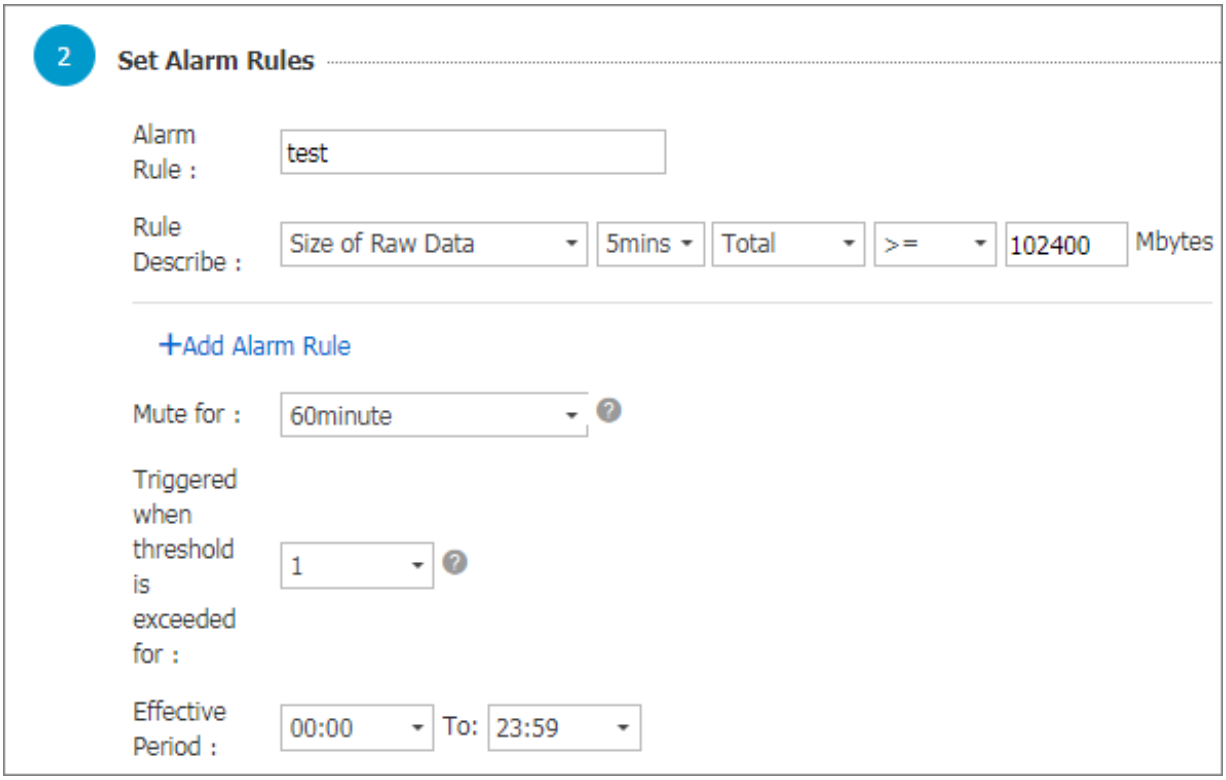
Each shard in a Logstore provides the write capability of 5 MB/s (500 times per second), which is sufficient in most cases. When the capability limit is exceeded, Log Service attempts to serve (rather than deny) your requests, but does not guarantee the availability of data that exceeds the limit during 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

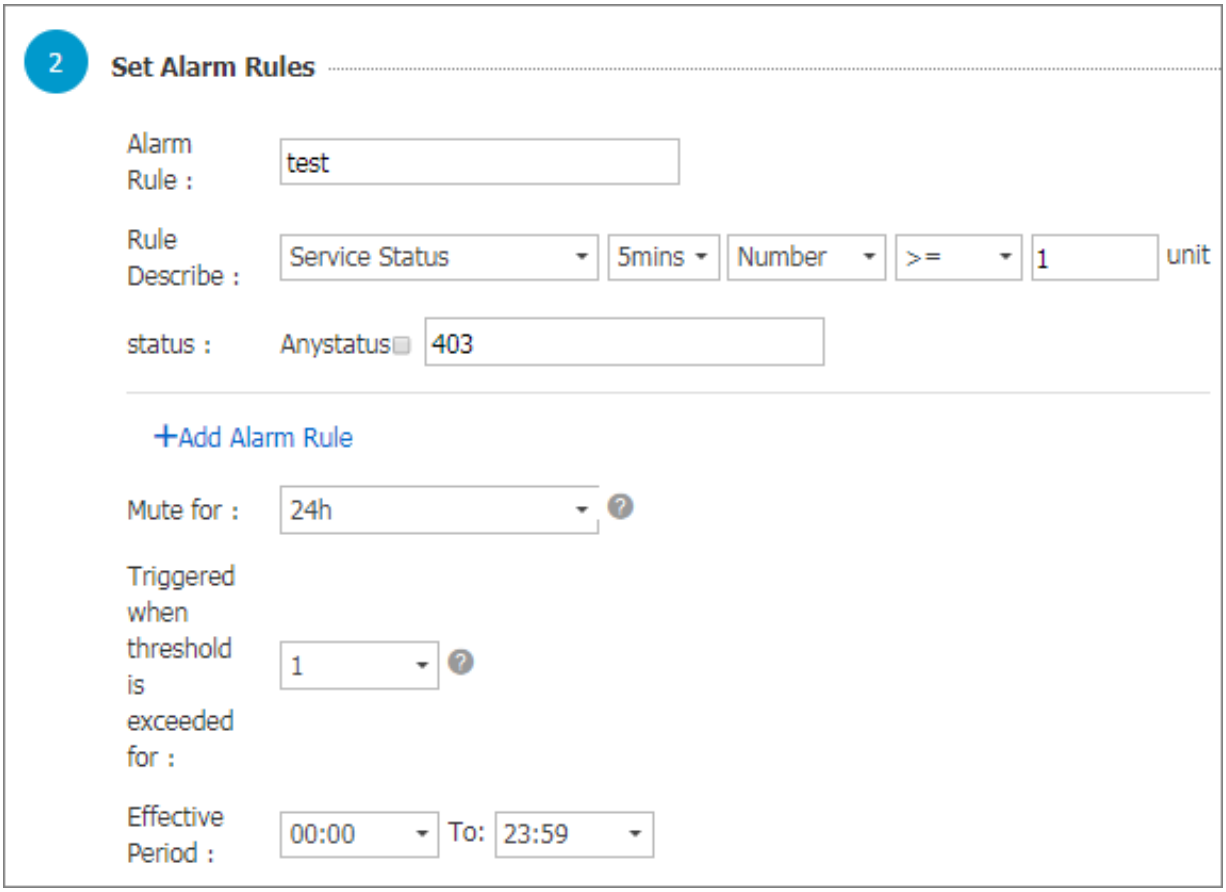


Solution 2: Set an alarm rule on service status

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



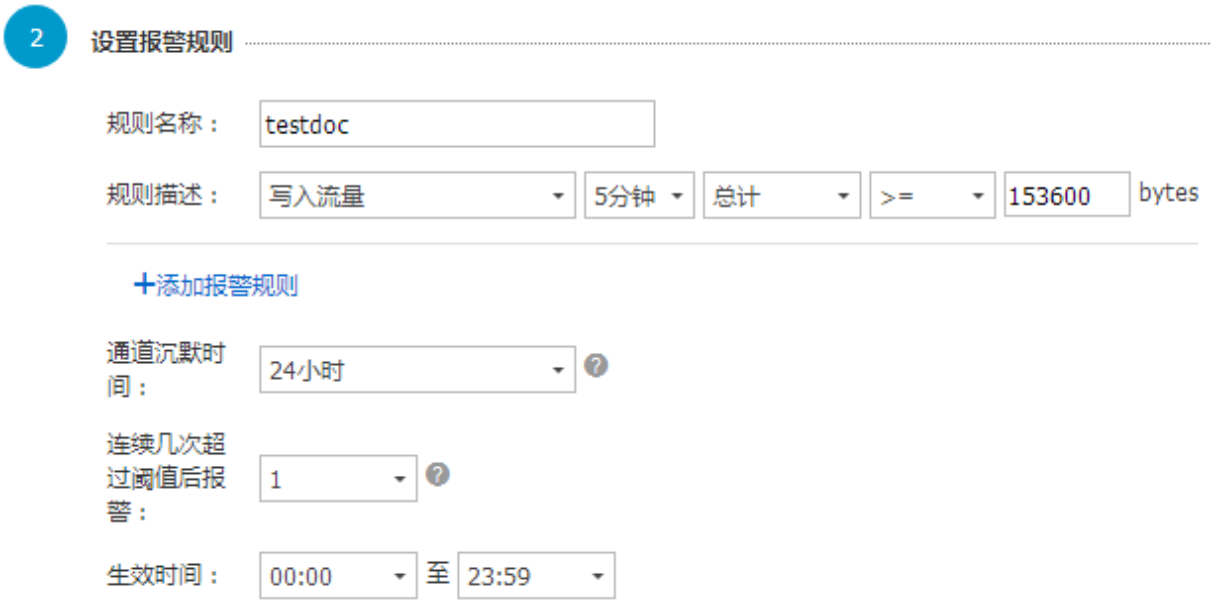
Monitor write traffic of projects

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