Alibaba Cloud

Log Service

Developer Guide

Document Version: 20220518

C–) Alibaba Cloud

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
⚠ Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Danger:** Resetting will result in the loss of user configuration data.
🔔 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	🔔 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
🔊 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	🔊 **Notice:** If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings> Network> Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
Courier font	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents
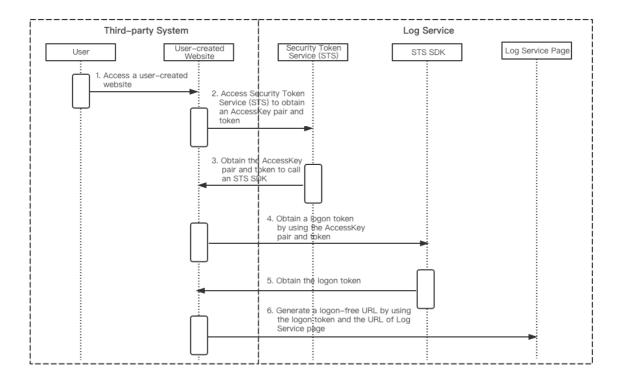
1.Other visualization methods

1.1. Embed console pages and share log data

Log Service allows you to embed console pages in a self-managed website. These pages include query and analysis pages and dashboard pages. This feature allows you to view log data with ease.

Context

After you configure the required features of Log Service, you can collect, query, share log data, and access dashboards by using a website. This prevents the high management costs that are incurred by multiple Resource Access Management (RAM) users. Log Service allows you to embed specified query and analysis pages and dashboard pages in a self-managed website. You can access these pages without the need to log on to the Log Service console. You can use RAM to control access to these pages. For example, you can grant the read-only permissions to a RAM user. For more information, see Step 2: Grant permissions to the RAM user.

The following figure shows the access process.

Procedure

1. Log on to the self-managed website. Use Security Token Service (STS) to obtain a temporary AccessKey pair and an STS token.

   ○ For information about STS, see Use STS to enable cross-account access to Log Service resources.

   ○ For information about how to authorize access to a specific Logstore, see Create a RAM user and authorize the RAM user to access Log Service and Create a RAM role whose trusted entity is an

Alibaba Cloud account and authorize the RAM role to access Log Service.

2. Call the STS SDK to obtain a temporary AccessKey pair and STS Token.

> ⑦ Note   The STS token may contain special characters. Before you use the token, encode the special characters by using URL encoding.

Sample request:

```
http://signin.aliyun.comsignin-intl.aliyun.com/federation?Action=GetSigninToken
                &AccessKeyId=<Temporary AccessKey ID that is returned by STS>
                &AccessKeySecret=<Temporary AccessKey secret that is returned by ST
S>

                &SecurityToken=<Token that is returned by STS>
                &TicketType=mini
```
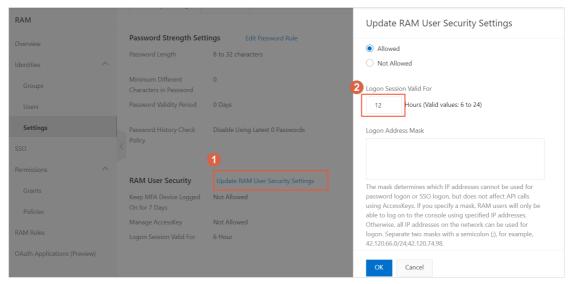
3. Modify the validity period of the STS token.

By default, the validity period of an STS token ranges from 15 minutes to 60 minutes. You can perform the operations in this step to modify the maximum validity period of an STS token to 12 hours.

i. In the RAM console, change the value of the Maximum Session Duration parameter to 12 hours for the specified RAM role.

ii. In the RAM console, change the value of the Logon Session Valid For parameter to 12 hours for the specified RAM user.

iii. Change the value of the setDurationSeconds field in the logon-free code to 43200L.

The following example shows the JAVA code:

```
AssumeRoleRequest assumeRoleReq = new AssumeRoleRequest();
assumeRoleReq.setRoleArn(roleArn);
assumeRoleReq.setRoleSessionName(roleSession);
assumeRoleReq.setMethod(MethodType.POST);
assumeRoleReq.setDurationSeconds(43200L);
```

4. Generate a logon-free URL.

i. Obtain a URL to access Log Service resources.

■ Obtain a complete query and analysis page:

```
https://sls4service.console.aliyun.com/lognext/project/<Project name>/logsearch/<
Logstore name>?hideTopbar=true&hideSidebar=true
```

■ Obtain a query and analysis page:

```
https://sls4service.console.aliyun.com/lognext/project/<Project name>/logsearch/<
Logstore name>?isShare=true&hideTopbar=true&hideSidebar=true
```

■ Obtain a dashboard page:

```
https://sls4service.console.aliyun.com/lognext/project/<Project name>/dashboard/<
Dashboard ID>?isShare=true&hideTopbar=true&hideSidebar=true
```

> ⑦ Note   The preceding Dashboard ID appears only in the URL of a dashboard page.
> This ID is not the name that appears on the dashboard.

ii. Generate a logon-free URL by using the logon token and the URL of the Log Service page.

Sample request:

```
http://signin.aliyun.comsignin-intl.aliyun.com/federation?Action=Login
                       &LoginUrl=<The URL to which you are redirected if your
logon fails. This URL is set to the URL to which you are redirected if the HTTP sta
tus code 302 is returned. >
                       &Destination=<The URL of a Log Service query and analys
is page or dashboard page in the Log Service console. If parameters exist, call the
encodeURL function to transcode the parameters. >
                       &SigninToken=<Logon token>
```

Examples

The following sample scripts are written in Java, PHP, and Python:

- 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
- Go

1.2. Customize the UI of embedded pages

This topic describes how to set parameters to customize the UI of embedded pages in the Log Service console.

Log Service allows you to embed console pages in a self-managed website without the need to log on to the console. This simplifies log query and analysis. Log Service also provides a series of UI parameters. You can set these parameters to customize the UI of console pages that are embedded in a self-managed website. For more information, see Embed console pages and share log data.

URL format

UI parameters are encoded in the URL based on the following format:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?Parameter1&Parameter2
```

> ⑦ Note
> - All parameters except the *${ProjectName}*, *${LogstoreName}*, *${savedsearchID}*, and *${dashboardID}* must be encoded after the question mark (?) at the end of a URL.
> - You can set multiple parameters at a time. The parameters are separated by ampersands (&).
> - To use the dark theme, you can add theme=dark&sls_iframe=true to a URL.

Common parameters

The following table describes the common parameters that you can use to customize console pages.

Parameter	Type	Required	Description	Example
hiddenBack	boolean	No	Specifies whether to hide the back button on the Search & Analysis page of the Logstore.	hiddenBack=true
hiddenChangeProject	boolean	No	Specifies whether to hide the Switch button of the project.	hiddenChangeProject=true
hiddenOverview	boolean	No	Specifies whether to hide the Overview tab of the project.	hiddenOverview=true
ignoreTabLocalStorage	boolean	No	Specifies whether to close the accessed tabs.	ignoreTabLocalStorage=true
queryTimeType	long	No	Specifies the time range of a query. For more information, see Values of the queryTimeType parameter. Valid values: • 1 to 26: Each integer in the interval indicates a time range. • -2: a custom time range (relative). You must specify a start time and end time. Example: start:-10m,end:now. • -3: a custom time frame. You must specify a start time and end time. Example: start:-2h,end:absolute. • 99: a custom time range. If you set queryTimeType to 99, you must set the startTime and endTime parameters to timestamps.	queryTimeType=1
startTime	Timestamp (date)	No	Specifies the start time of a query. This parameter is available only when you set queryTimeType to 99.	startTime=1547776643
endTime	Timestamp (date)	No	Specifies the end time of a query. This parameter is available only when you set queryTimeType to 99.	endTime=1547776731

Values of the queryTimeType parameter

queryTimeType	Description
1	1 minute (relative)
2	15 minutes (relative)
3	1 hour (relative)
4	4 hours (relative)

queryTimeType	Description
5	1 day (relative)
6	1 week (relative)
7	30 days (relative)
8	1 minute (time frame)
9	15 minutes (time frame)
10	1 hour (time frame)
11	4 hours (time frame)
12	1 day (time frame)
13	1 week (time frame)
14	30 days (time frame)
15	Today (time frame)
16	Yesterday (time frame)
17	The day before yesterday (time frame)
18	This week (time frame)
19	Previous week (time frame)
20	This month (time frame)
21	This quarter (time frame)
22	Today (relative)
23	5 minutes (relative)
24	This year (time frame)
25	This month (relative)
26	Previous month (time frame)
27	This week (relative)
99	Specifies a custom time range. If you set queryTimeType to 99, you must specify the startTime and endTime parameters.

The following examples show the URL parameters and results:

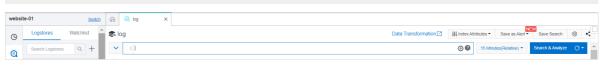- To hide the back button, Switch button and Overview tab, use the following URL:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${Logstor
eName}?hiddenBack=true&hiddenChangeProject=true&hiddenOverview=true
```


- To hide the back button on the Search & Analysis page of a specified Logstore, use the following URL:
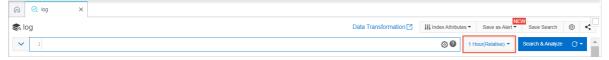
```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${Logstor
eName}?hiddenBack=true
```


- To set the query and analysis time, use the following URL:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${Logstor
eName}?queryTimeType=3
```


Parameters of the Search & Analysis page

The following table describes the parameters that you can set to customize the Search & Analysis page of a Logstore.

Parameter	Type	Required	Description	Example
ProjectName	string	Yes	The name of the project.	website-01
LogstoreName	string	Yes	The name of the Logstore.	logstore01
queryString	string	No	The query statement that is encoded in Base64. For example, the `*|select count(*)` statement is `KnxzZWxlY3QgY291bnQoKik=` after it is encoded in Base64.	KnxzZWxlY3QgY291bnQoKik=
readOnly	boolean	No	Specifies whether to hide the buttons on the Search & Analysis page, such as **Share**, **Index Attributes**, **Save Search**, and **Save as Alert**.	readOnly=true

Parameter	Type	Required	Description	Example
encode	string	No	Specifies whether to encode a query string. To prevent special characters in the value of the queryString parameter, you can set the encode parameter to base64. Then, the value of the queryString parameter is a Base64-encoded string.	encode=base64
hiddenEtl	boolean	No	Specifies whether to hide the Data Transformation button.	hiddenEtl=true
hiddenShare	boolean	No	Specifies whether to hide the Share button.	hiddenShare=true
hiddenIndexSetting	boolean	No	Specifies whether to hide the Index Attributes button.	hiddenIndexSetting=true
hiddenSavedSearch	boolean	No	Specifies whether to hide the Save Search button.	hiddenSavedSearch=true
hiddenAlert	boolean	No	Specifies whether to hide the Save as Alert button.	hiddenAlert=true
hiddenQuickAnalysis	boolean	No	Specifies whether to hide the Quick Analysis search bar.	hiddenQuickAnalysis=true
hiddenDownload	boolean	No	Specifies whether to hide the download icon.	hiddenDownload=true
hiddenPSql	boolean	No	Specifies whether to hide the Dedicated SQL button.	hiddenPSql=true
keyDispalyMode	string	No	Specifies the display mode of log entries. • Single: Log entries are displayed in a single line. • Multi: Log entries are displayed in multiple lines.	keyDispalyMode=single

The following examples show the URL parameters and results:

- To set a string for a query, use the following URL:

  For example, the `*|select count(*)` statement is `KnxzZWxlY3QgY291bnQoKik=` after it is encoded in Base64.

  ```
  https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreName}?encode=base64&queryString=KnxzZWxlY3QgY291bnQoKik=
  ```

- To hide buttons such as **Index Attributes** and **Save as Alert**, use the following URL:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${Logstor
eName}?readOnly=true
```
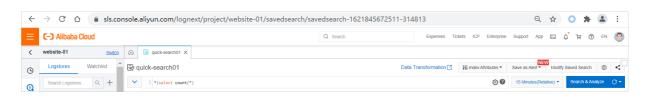

Parameters on the saved search page of a Logstore

The following table describes the parameters that you can set to customize the saved search page of a Logstore.

Parameter	Type	Required	Description	Example
ProjectName	string	Yes	The name of the project.	website-01
savedSearchName	string	No	The name of the saved search.	quick-search01
savedsearchID	string	Yes	The ID of the saved search. ⑦ **Note** After you create a saved search, you can obtain the ID of the saved search in the URL. For more information, see Obtain the ID of a saved search.	savedsearch-1621845672511-314813

The following example shows the URL parameters and result:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/savedsearch/${savedse
archID}
```

Embedded parameters of dashboards

The following table describes the embedded parameters that you can set to customize dashboard pages.

Parameter	Type	Required	Description	Example
ProjectName	string	Yes	The name of the project.	website-01
dashboardName	string	No	The name of the dashboard.	Website Logs
dashboardID	string	Yes	The ID of the dashboard. ⓘ **Note** After you create a dashboard, you can obtain the ID of the dashboard in the URL. For more information, see Description.	dashboard-1609817292009-742588
filters	string	No	The filter conditions. The value of this parameter must be transcoded by encodeURIComponent(). For example, `filters=key1:value1&filters=key2:value2` is transcoded to `filters=key1%3Avalue1%26filters%3Dkey2%3Avalue2`.	filters=key1%3Avalue1%26filters%3Dkey2%3Avalue2
token	JsonString	No	The variables to be replaced. The value of this parameter must be transcoded. For example,token=`"projectname","value":"1"}, {"key": "region", "value": [{"key":"hangzhou"}]` is transcoded to `token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D`.	token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D

Parameter	Type	Required	Description	Example
readOnly	boolean	No	Specifies whether to hide the buttons on the dashboard page, such as the **Edit** and **Alerts** buttons.	readOnly=true
hiddenFilter	boolean	No	Specifies whether to hide the filter conditions.	hiddenFilter=true
hiddenToken	boolean	No	Specifies whether to hide the variables to be replaced.	hiddenToken=true
hiddenProject	boolean	No	Specifies whether to hide the project information.	hiddenProject=true
hiddenEdit	boolean	No	Specifies whether to hide the Edit button.	hiddenEdit=true
hiddenReport	boolean	No	Specifies whether to hide the Subscribe button.	hiddenReport=true
hiddenTitleSetting	boolean	No	Specifies whether to hide the title setting button.	hiddenTitleSetting=true
hiddenReset	boolean	No	Specifies whether to hide the Reset Time button.	hiddenReset=true
hiddenPSql	boolean	No	Specifies whether to hide the Dedicated SQL button.	hiddenPSql=true
autoFresh	string	No	The interval at which the dashboard is refreshed, for example, 30 seconds or 5 minutes. The minimum refresh interval is 15 seconds.	autoFresh=5m

The following examples show the URL parameters and results:

- To enable read-only mode for the dashboard page, use the following URL:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardID}?readOnly=true
```
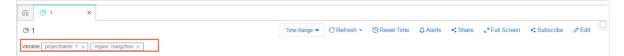

- To add two filter conditions of key 1=value 1 and key 2=value 2, use the following URL. The filter condition `filters=key1:value1&filters=key2:value2` is transcoded to `filters%3Dkey1%3Avalue1%26filters%3Dkey2%3Avalue2`.

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboardID}?filters=key1%3Avalue1%26filters%3Dkey2%3Avalue2
```

- To add multiple conditions projectname=1 and region=hangzhou to replace variables, use the following URL. token=[{"key": "projectname","value":"1"}, {"key": "region", "value": `"projectname","value":"1"}, {"key": "region", "value": "hangzhou"}]` is transcoded to `token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D` .

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboa
rdID}?token=%5B%7B%22key%22%3A%20%22projectname%22%2C%22value%22%3A%221%22%7D%2C%20%7B%22
key%22%3A%20%22region%22%2C%20%22value%22%3A%20%22hangzhou%22%7D%5D
```
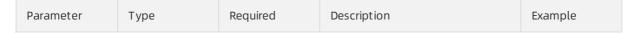

- To refresh a dashboard every 5 minutes, use the following URL:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/dashboard/${dashboa
rdId}?autoFresh=5m
```


Tree structure parameters

The following table describes the tree structure parameters that you can set to customize the left-side navigation pane of the console.

Parameter	Type	Required	Description	Example

Parameter	Type	Required	Description	Example
treeConfig	JSON	No	Specifies the tree structure of the left-side navigation pane. If you set treeEncode to base64, you need to transcode the value of treeConfig by using Base64 first. For example, `{"logstore": {"expand":true,"resourceList":["delete-log"],"template":["savedsearch","alert"]}}` is transcoded to `eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGlzdCI6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlIjpbInNhdmVkc2VhcmNoIiwiYWxlcnQiXX19` by using Base64.	eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGlzdCI6WyJkZWxldGUtbG9nIl0sInRlbXBsYXRlIjpbInNhdmVkc2VhcmNoIiwiYWxlcnQiXX19
treeEncode	string	No	The encoding method for treeConfig. Default value: null. This value indicates that treeConfig is not encoded. Only Base64 is supported.	treeEncode=base64

The following example shows the related parameters in the treeConfig parameter:

```
{
    "logstore": {
        "search": true,
        "expand": true,
        "resourceList": [
            "L1",
            "L2"
        ],
        "template": [
            "favor",
            "logtail",
            "import",
            "etl",
            "savedsearch",
            "alert",
            "export",
            "consumergroup",
            "dashboard"
        ]
```

```
    },
    "machineGroup": {
        "search": true,
        "resourceList": [
            "m1",
            "m2"
        ]
    },
    "savedSearch": {
        "search": true,
        "resourceList": [
            "s1",
            "s2"
        ]
    },
    "alarm": {
        "search": true,
        "resourceList": [
            "a1",
            "a2"
        ]
    },
    "dashboard": {
        "search": true,
        "resourceList": [
            "d1",
            "d2"
        ]
    },
    "etl": {
        "search": true,
        "resourceList": [
            "e1",
            "e2"
        ]
    }
}
```

Parameters of treeConfig

Parameter	Type	Required	Description
logstore	object	No	Specifies whether to display the nodes in the left-side navigation pane of the Logstores tab.
template	string[]	No	Specifies whether to display the icons on the Logstores tab. For more information, see Parameters of template.
machineGroup	object	No	Specifies whether to display the nodes in the left-side navigation pane of the Machine Groups tab.

Parameter	Type	Required	Description
savedSearch	object	No	Specifies whether to display the nodes in the left-side navigation pane of the Saved Search tab.
alert	object	No	Specifies whether to display the nodes in the left-side navigation pane of the Alerts tab.
dashboard	object	No	Specifies whether to display the nodes in the left-side navigation pane of the Dashboard tab.
etl	object	No	Specifies whether to display the nodes in the left-side navigation pane of the Data Transformation tab.

Resource parameters

| Parameter	Type	Required	Description
search	boolean	No	Specifies whether to display the search box. Default value: true. This value indicates that the search box is displayed.
resourceList	String[]	No	Displays the resource list of the specified resource. If this parameter is set to an empty array, an empty list is displayed. If this parameter is unspecified, all resources are displayed in the list based on an exact match. By default, all resources of the list are displayed.
expand	boolean	No	Specifies whether to expand the list. Default value: false. This value indicates that the list is not expanded. This parameter is valid only for the Logstore list.

Parameters of template
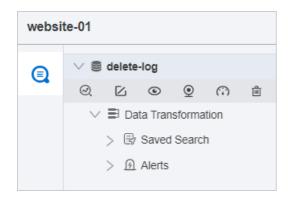
Parameter	Type	Required	Description
favor	string	No	The watchlist.
logtail	string	No	The Logtail configurations.
import	string	No	The data import configurations.
etl	string	No	The data transformation tasks.
savedsearch	string	No	The saved searches.

Parameter	Type	Required	Description
alert	string	No	The alerts.
export	string	No	The data export tasks.
consumergroup	string	No	The data consumption.
dashboard	string	No	The visual dashboards.

The following example shows the URL parameters and result:

To customize the left-side navigation pane, use the following URL:

```
https://sls4service.console.aliyun.com/lognext/project/${ProjectName}/logsearch/${LogstoreN
ame}?treeconfig=eyJsb2dzdG9yZSI6eyJleHBhbmQiOnRydWUsInJlc291cmNlTGlzdCI6WyJkZWxldGUtbG9nIl0
sInRlbXBsYXRlIjpbInNhdmVkc2VhcmNoIiwiYWxlcnQiXX19&hiddenBack=true&hiddenChangeProject=true&
hiddenOverview=true&treeEncode=base64&ignoreTabLocalStorage=true
```


Advanced dashboard parameters

If you embed an inline frame into a dashboard, the height of the inline frame cannot be calculated. The scroll bar of the dashboard and the scroll bar of the inline frame may appear at the same time. In this case, you can set advanced dashboard parameters to adjust the dashboard height.

To fix this issue, you can use the postMessage method of Log Service to obtain the height of the dashboard and specify the height for the inline frame. The following sample code is used:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>POST message test</title>
</head>
<style>
  * {
    padding: 0;
    margin: 0;
  }
  iframe {
    display: block;
    width: 100%;
  }
</style>
<body>
  <script>
    window.addEventListener('message',function(e){
      console.log(e.data.dashboardHeight)
      document.getElementById('test').style.height = e.data.dashboardHeight + 'px'
    });
  </script>
  <div style="height: 700px;">somethings</div>
  <iframe id="test" src="http://sls4service.console.aliyun.com/lognext/project/${projectNam
e}/dashboard/${dashboardName}?product=${productCode}">
</body>
</html>
```
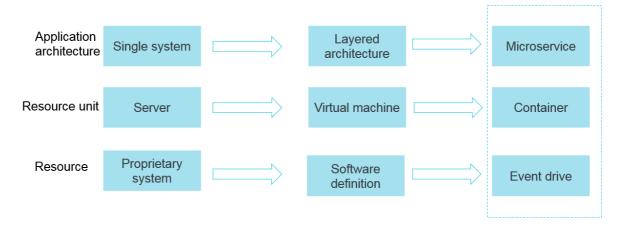
1.3. Connect Log Service with Jaeger

This topic describes how to connect Log Service with Jaeger.

Context

Containers and serverless programming improve efficiency in software delivery and deployment. The evolution of the standard architecture includes the following changes:

- The application architecture changes from single system-based to microservice-based. Business logic changes to calls and requests between microservices.
- In terms of resources, traditional servers are gradually replaced with virtual resources.

The standard architecture becomes more elastic due to these changes. However, these changes also pose more challenges to operations and maintenance (O&M) and diagnostics. To address this issue, a series of development and operations (DevOps)-oriented diagnostic and analysis systems emerge. These systems include centralized logging systems, centralized metrics systems, and distributed tracing systems.

In addition to Jaeger, Alibaba Cloud also supports the OpenTracing link tracing service XTrace.

Features of logging, metrics, and tracing systems
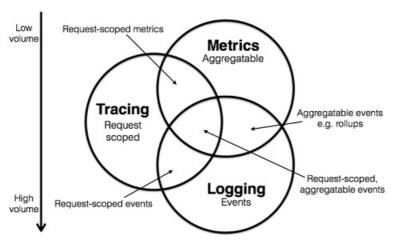
- Logging

  A logging system records discrete events. For example, you can use the debug information or error messages of applications to troubleshoot issues.

- Metrics

  A metrics system records data that can be aggregated. For example, you can use the depth of a queue as a metric and update the queue when an element is added or removed. You can also create a counter that calculates the number of HTTP requests.

- Tracing

  A tracing system records the information during a request. For example, a system records the execution process and duration of a remote procedure call (RPC). A tracing system allows you to troubleshoot the performance issues of a system with ease.

The preceding information allows you to classify the existing systems. For example, Zipkin is a tracing system, and Prometheus is a metrics system and will integrate more and more tracing features. More and more systems will focus on logging and integrate more features of other fields. These systems include ELK and Alibaba Cloud Log Service. The aim is to approach the intersection area, as shown in the preceding figure.

For more information, see Metrics, tracing, and logging.
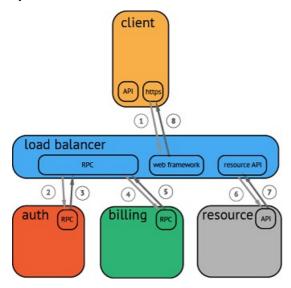
Tracing

The following tracing systems are well-known:

- Google Dapper
- Google Operations
- Twitter Zipkin
- Go Appdash
- Taobao EagleEye
- Alibaba Cloud Ditecting
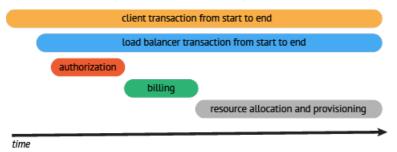- Ant Financial Yuntu
- Shenma sTrace
- AWS X-Ray

A large variety of distributed tracing systems exist and develop at high speed.

- Code tracking
- Data storage
- Query display

The following figure shows the lifecycle of a distributed call. When a client makes a request, the request is first sent to the load balancer. The request passes through the authentication service and billing service. Then, the request is sent to the requested resources. After the lifecycle is complete, the system returns a result.
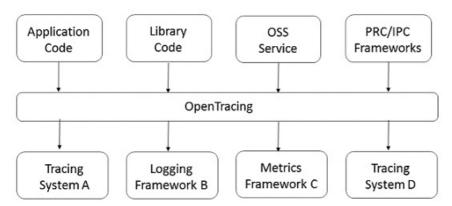
After the system collects and stores the data, the distributed tracing system can be used to generate a timing diagram that contains a timeline. However, during data collection, the system obtains user code in an intrusive manner and the API operations of different systems are incompatible. This poses a significant challenge if you switch tracing systems.

OpenTracing

The OpenTracing standard was introduced to prevent API compatibility issues among different distributed tracing systems. OpenTracing is a lightweight standardization layer. This layer is located between applications or class libraries and tracing or log analysis programs. Fore more information, see OpenTracing.

- Benefits:
  - OpenTracing is included in Cloud Native Computing Foundation (CNCF) and provides standard concepts and data standards for global distributed tracing systems.
  - OpenTracing provides platform-independent and vendor-independent APIs. This allows developers to add or change the implementation of tracing systems with ease.
- Data model:

  In OpenTracing, a span is the primary building block of a trace. A trace can be treated as a directed acyclic graph (DAG). The DAG consists of multiple spans. The relationships between spans are called references. The following example shows a trace that consists of eight spans.

```
Causal relationships between spans in a single trace.
       [Span A]  <------(the root span)
           |
    +------+------+
    |             |
 [Span B]     [Span C] <------(ChildOf: Span C is a child node of Span A.)
    |             |
 [Span D]     +---+-------+
              |           |
          [Span E]    [Span F] >>> [Span G] >>> [Span H]
                                      ^
                                      ^
                                      ^
                          (FollowsFrom: Span G is invoked after Span F.)
```

  To view more details about a trace, use a timeline-based timing diagram, as shown in the following example.

```
The time relationship between spans in a trace.
--|-------|-------|-------|-------|-------|-------|-------|->time
  [Span A·············································]
    [Span B········································]
      [Span D····································]
    [Span C·································]
        [Span E·······]         [Span F··] [Span G··] [Span H··]
```

  Each span includes the following objects:

  - An operation name.
  - A start timestamp.
  - A finish timestamp.
  - A set of span tags. Span tags are key-value pairs. In a key-value pair, the key must be a string and the value can be a string, Boolean, or numeric value.
  - A set of span logs. Each log contains a key-value pair and a timestamp. In a key-value pair, the key must be a string and the value can be of an arbitrary type. However, not all OpenTracing tracers must support every value type.
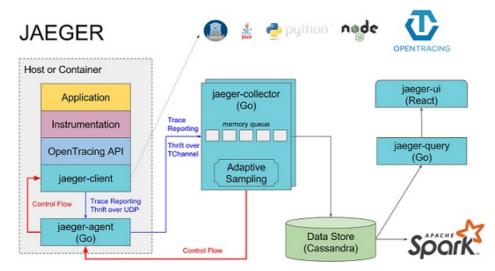
- A SpanContext. Each SpanContext contains the following components:

    - An OpenTracing tracer must transmit the current trace status, for example, trace ID and span ID, across process boundaries based on a specified span.

    - Baggage items are key-value pairs that are included in a trace. These key-value pairs must be transmitted across process boundaries.

  - References: indicate the relationships between zero or more related spans. Relationships are established between spans based on a specified SpanContext.

  For more information about the OpenTracing data model, see OpenTracing data model.

For more information about the tracers that are supported by OpenTracing, see OpenTracing. Popular OpenTracing tracers include Jaeger and Zipkin. Fore more information, see Jaeger and Zipkin.

Jaeger

Jaeger is an open-source distributed tracing system that is provided by Uber. This system is compatible with the OpenTracing API.
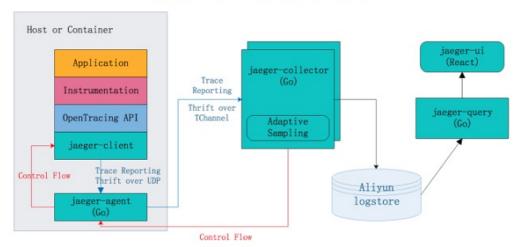

Jaeger consists of the following components, as shown in the preceding figure.

- Jaeger client: implements language-specific SDKs that conform to the OpenTracing standard. An application calls an API operation to write data. A client library transmits trace information to an agent based on the sampling policy that is specified in the application.

- Agent: works as a network daemon that listens to a UDP port. An agent is used to receive span data and send the span data to a collector in batches. An agent is a basic component that is deployed to all hosts. An agent allows clients libraries to connect to a collector without the required connection details.

- Collector: receives data from agents and writes the data to a data store. A collector is a stateless component. You can run an arbitrary number of jaeger-collectors.

- Data store: works as a pluggable component that allows you to write data to Cassandra and Elasticsearch.

- Query: receives query requests, retrieves trace information from a data store, and then shows the result in the user interface (UI). The Query service is stateless. You can start multiple instances, and deploy the instances behind NGINX load balancers.

Jaeger on Alibaba Cloud Log Service

Jaeger on Alibaba Cloud Log Service is a Jaeger-based distributed tracing system that stores tracing data to Log Service for persistent storage. The tracing data can be queried and displayed by using the Jaeger API. For more information, see Jaeger on Alibaba Cloud Log Service.

Benefits:

- Jaeger allows you to only store data to Cassandra and Elasticsearch for persistent storage. You must maintain the stability of data stores and adjust the storage capacity. Jaeger on Alibaba Cloud Log Service uses Log Service to process large amounts of data. This allows you to benefit from the Jaeger distributed tracing technology, and focus on data stores.

- The Jaeger UI can query and show traces, but requires more support for analysis and troubleshooting. Jaeger on Alibaba Cloud Log Service allows you to use the query and analysis features of Log Service to analyze system issues in an efficient manner.

- Compared with Jaeger that uses Elasticsearch as the data store, the costs of Log Service are only 13% of the price of Elasticsearch when the pay-as-you-go billing method is adopted. For more information, see Comparison between Log Service and the ELK stack in log query and analysis scenarios.

For more information about how to configure Jaeger on Alibaba Cloud Log Service, visit GitHub.

1.4. Connect to Log Service by using JDBC

This topic describes how to connect to Log Service by using Java Database Connectivity (JDBC). This topic also describes how to query and analyze logs by using the MySQL protocol and SQL syntax.

Prerequisites

Indexes are configured for the fields that you want to query, and the switches in the Enable Analytics column of the fields are turned on. For more information, see Configure indexes.

Context

MySQL is a popular relational database service. A large number of software products can obtain MySQL data by using the MySQL protocol and SQL syntax. To connect to Log Service by using JDBC, you must be familiar with the SQL syntax. Log Service allows you to query and analyze logs based on the MySQL protocol. You can use a standard MySQL client to connect to Log Service. Then, you can use standard SQL syntax to query and analyze logs. The following clients support the MySQL protocol: MySQL client, JDBC, and Python MySQLdb.

Use scenarios of JDBC:

- Use visualization tools such as DataV, Tableau, or Grafana to connect to Log Service in compliance with the MySQL protocol. For more information, see Analyze logs by using the JDBC API.

- Use libraries such as JDBC in Java or MySQLdb in Python to access Log Service and process query results from an application.

Sample log

Each log of a bike-sharing service contains information about a user. The information includes the age, gender, battery usage, bike ID, operation latency, latitude, lock type, longitude, operation type, operation result, and unlocking type. The logs are stored in the Logstore:ebike Logstore of the project:trip_demo project. The project resides in the China (Hangzhou) region.

Sample log:

```
Time :10-12 14:26:44
__source__: 192.168.0.0
__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
```

JDBC-based query and analysis

1. Create a Maven project and add the following JDBC dependency to the pom.xml file:

```
<dependency>
 <groupId>MySQL</groupId>
 <artifactId>mysql-connector-java</artifactId>
 <version>5.1.6</version>
</dependency>
```

2. Create a Java program to query logs by using JDBC. The following sample code shows how to create a Java program to query logs by using JDBC.

   You must include the __date__ or __time__ field in a WHERE clause to limit the time range of a query. The data type of the __date__ field is timestamp, and the data type of the __time__ field is bigint. Examples:

   ○ __date__ > '2017-08-07 00:00:00' and __date__ < '2017-08-08 00:00:00'

   ○ __time__ > 1502691923 and __time__ < 1502692923

> 📢 **Notice** If you use libraries such as JDBC in Java or MySQLdb in Python to access Log Service from an application, you must use an internal endpoint of Log Service. The internal endpoint is accessible over the classic network or a virtual private cloud (VPC). If you use a different endpoint, the following error occurs: `com.mysql.jdbc.exceptions.jdbc4.CommunicationsException: Communications link failure,Caused by: java.net.ConnectException: Connection timed out: connect`.

```java
/**
* 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[]){
     final String endpoint = "trip-demo.cn-hangzhou-intranet.log.aliyuncs.com"; // The
project name and an internal endpoint of Log Service. You must replace the name and end
point with actual values. For more information, see Endpoints.
     final String port = "10005"; // The port that is used for JDBC-based access. The d
efault port is 10005.
     final String project = "trip-demo"; // The name of the project.
     final String logstore = "ebike"; // The name of the Logstore.
     final String accessKeyId = "";  // The AccessKey ID of your Alibaba Cloud account.
For more information, see AccessKey pair.
     final String accessKey = "";  // The AccessKey secret of your Alibaba Cloud accoun
t.
     Connection conn = null;
     Statement stmt = null;
     try {
         // Step 1: Load the JDBC driver.
         Class.forName("com.mysql.jdbc.Driver");
         // Step 2: Create a connection string.
         conn = DriverManager.getConnection("jdbc:mysql://"+endpoint+":"+port+"/"+proje
ct,accessKeyId,accessKey);
         // Step 3: Create a statement.
         stmt = conn.createStatement();
         // Step 4: Define a query statement to query the number of logs that were gene
rated on October 11, 2017 and whose op field is set to unlock.
         String sql = "select count(1) as pv,avg(latency) as avg_latency from "+logstor
e+"  " +
                 "where     __date__  >=  '2017-10-11 00:00:00'   " +
                 "    and  __date__   <   '2017-10-12 00:00:00'" +
                 " and     op ='unlock'";
         // Step 5: Execute the query statement.
         ResultSet rs = stmt.executeQuery(sql);
         // Step 6: Extract the query result.
         while(rs.next()){
             //Retrieve by column name
             System out print("pv:");
```

```
                System.out.print("pv:");
                // Display pv in the result.
                System.out.print(rs.getLong("pv"));
                System.out.print(" ; avg_latency:");
                // Display avg_latency in the result.
                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();
                }
            }
        }
    }
}
```

1.5. Connect Log Service with DataV

DataV is an Alibaba Cloud data visualization service. DataV allows you to build professional visualization applications by using a graphical user interface (GUI) with ease. You can use DataV to visualize log analysis data. This topic describes how to connect Log Service with DataV and show data on a dashboard.

Prerequisites

- Log data is collected. For more information, see Data collection.

- The index feature is enabled and configured. For more information, see Configure indexes.
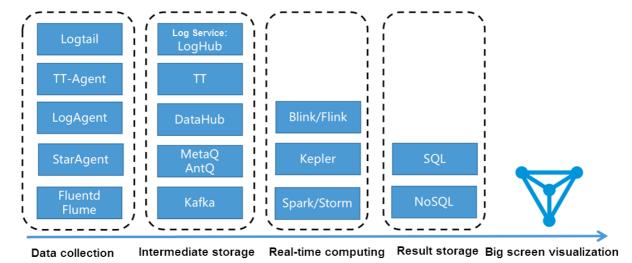
Context

Rea-time dashboards are widely used in large online promotions. Real-time data dashboards are based on a stream computing architecture. The architecture consists of the following modules:

- Data collection: collects data from each source in real time.

- Intermediate storage: uses Kafka Queues to decouple production systems and consumption systems.

- Real-time computing: subscribes to real-time data and uses computing rules to compute data on the dashboard.

- Result storage: stores the computing results in SQL and NoSQL databases.

- Visualization: calls API operations to obtain results and visualize the results.

Alibaba Group provides multiple services to support these modules, as shown in the following figure.

You can connect Log Service with DataV by calling the API operations that are related to the log search and analytics feature. Then, you can use DataV to show data on a dashboard.

Features

The following computing methods are supported:

- Real-time computing (streaming computing): fixed computing and dynamic data.

- Offline computing (data warehouse and offline computing): dynamic computing and fixed data.

For scenarios that have a high-level requirement of timeliness, Log Service allows you to enable the real-time data indexing feature on logs that are stored in LogHub. You can use the log search and analytics feature to query and analyze these logs in an efficient manner. This method has the following benefits:

- Fast: Billions of data records can be processed and queried within one second. Each query statement can have a maximum of five specified conditions. Hundreds of millions of data records can be analyzed and aggregated within one second without the need to wait and predict the results. Each

query statement can have a maximum of five aggregate functions and a group by clause.

- Real-time display: Up to 99.9% of logs can be displayed on the data dashboard within one second after these logs are generated.

- Dynamic data refresh: When you modify analysis methods or import data to Logstores, the display result is refreshed in real time.

This method has the following limits:

- Data volume: Up to 10 billion lines of data can be computed at a time. If you need to compute more data, you must set multiple time ranges.

- Flexibility: Only SQL-92 syntax is supported. User-defined functions (UDFs) are not supported.

Procedure

1. Create a DataV data source.

   i. Log on to the DataV console.

   ii. On the **Data Sources** tab, click **Add Source**. The following table describes the parameters.

Parameter	Description
The type of the database.	In the Add Data Source dialog box, select **Log Service** from the Type drop-down list.
Name	Enter a name for the data source, for example, log_service_api.
AppKey	The AccessKey ID of the Alibaba Cloud account, or the AccessKey ID of a RAM user that has the read permission on Log Service.
AppSecret	The AccessKey secret of the Alibaba Cloud account, or the AccessKey secret of a RAM user that has the read permission on Log Service.
Endpoint	The endpoint of the region where the Log Service project resides. For more information, see Endpoints.

2. Open a canvas.

   ○ On the **Projects** tab, select a project, and click **Edit**.

   ○ On the **Projects** tab, click **Create Project**. For more information about how to create a project, see Use a template to create a PC端 project.

3. Create a line chart and add a filter.

   i. Create a line chart.

   In the **Widgets** pane, choose **Line > Line Chart**. In the right-side Line Chart pane, click the **Data** tab, and click **Set**. In the **Configure Datasource** pane, set the required parameters. The following table describes the parameters.

Parameter	Description
Data Source Type	Select **Log Service** from the drop-down list.

Parameter	Description
Select Data Source	Select the log_service_api data source that you created in *Step 1*.

Query	The following code shows a sample query.
	
	``
	{
	 "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
	}
	``

The following table describes the parameters in the preceding query.

Parameter	Description
projectName	The name of the project.
logstoreName	The name of the Logstore.
topic	The log topic. Do not set this parameter if you have not set a log topic.

Parameter	Description
from, to	The from and to parameters refer to the start time and end time during which logs are obtained. ? **Note** In this example, the parameters are set to `:from` and `:to` . When you test your parameter settings, you can enter a UNIX timestamp, for example, 1509897600. After the test, change the UNIX timestamp to `:from` and `:to` . Then, you can specify a time range in the URL parameters. For example, the URL in the preview is `http://datav.aliyun.com/screen/86312` . After the URL `http://datav.aliyun.com/screen/86312?from=1510796077&to=1510798877` is opened, the values are computed based on the specified time.
query	The query criteria. For more information about the query syntax, see Log analysis overview. ? **Note** The time in the query must be in the yyyy/MM/dd HH:mm:ss format, for example, 2017/07/11 12:00:00. You must use the following syntax to align the time to the hour. Then, you can convert the time into the specified 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.

ii. After the configuration is complete, click **Data Response Result**.

iii. Create a filter.

In the Configure Datasource dialog box, select **Data Filter**, and click the plus (+) icon next to the **Add Filter** drop-down list.

Enter a function in the New Filter field by using the following example syntax:

```
return Object.keys(data).map((key) => {
let d= data[key];
d["pv"] = parseInt(d["pv"]);
return d;
});
```

- In the filter, convert the result that is used by the y-axis into the INT type. In this example, the y-axis indicates the PV, and the pv column must be converted.

- The results contain the t and pv columns. You can set the x-axis to t and the y-axis to pv.

4. Create a pie chart and add a filter.

   i. Create a **carousel pie chart**.

   In the **Widgets** pane, choose **Pie > Carousel Pie Chart**. In the right-side Carousel Pie Chart pane, click the **Data** tab, and click **Set**. In the **Configure Datasource** pane, set the required parameters. The following table describes the parameters.

Parameter	Description
Data Source Type	Select **Log Service** from the drop-down list.
Select Data Source	Select the log_service_api data source that you created in *Step 1*.
Query	The following code shows a sample query. ```\n{\n "projectName": "dashboard-demo",\n "logStoreName": "access-log",\n "topic": "",\n "from": 1509897600,\n "to": 1509984000,\n "query": "*| select count(1) as pv ,method group by\nmethod" ,\n "line": 100,\n "offset": 0\n}\n``` The parameters in the preceding code are described in Step 3.

ii. After the configuration is complete, click **Data Response Result**.

iii. Add a filter.

In the Configure Datasource pane, click **Data Filter**, and then click the plus sign (+) next to the
Add Filter field.

Enter a function in the New Filter field by using the following example syntax:

```
return Object.keys(data).map((key) => {
let d= data[key];
d["pv"] = parseInt(d["pv"]);
return d;
})
```

Enter method in the type field and pv in the value field for the carousel pie chart.

5. Use callback functions to retrieve a time range in real time. Perform the following steps to show
the real-time logs of the last 15 minutes.

i. Create a static data source with the default settings, and add a filter. Example:

```
return [{
  value:Math.round(Date.now()/1000)
}];
```

```
return [{
  value:Math.round((Date.now() - 24 * 60 * 60 * 1000)/1000)
}];
```

ii. On the **Interaction** tab, select **Enable** in the Trigger Event when Number Changed field, and
add the required values to the Bound Variable column.

iii. On the Data tab, the following example shows how to use the `:from` and `:to`
parameters to implement callback functions.

```
{
 "projectName": "dashboard-demo",
 "logStoreName": "access-log",
 "topic": "",
 "from": ":from",
 "to": ":to",
 "query": "*| select count(1) as pv ,referer group by pv desc limit 30" ,
 "line": 100,
 "offset": 0
}
```

6. Preview and publish the DataV dashboard.

i. Click the **Preview** icon to preview the DataV dashboard.

ii. Click the **Publish** icon to publish the DataV dashboard.

Example

You need to collect the page view (PV) statistics of your website across China during the Computing
Conference and show the data on a dashboard. You have configured full log data collection and
enabled the log search and analytics feature in Log Service. You need to only enter a query statement
in the Query field to obtain the PV statistics. During this period, the requirement often changes. In this
example, the following changes are made:

- Original requirement: On the first day of the conference, you need the statistics of unique visitors (UVs) for the present day.

  You need to query the values of the forward field under NGINX in all the access logs. The forward field records one or more IP addresses of each visitor. Each log has a forward field. You can use the following statement that includes the `approx_distinct(forward)` function to remove repeated IP addresses and obtain the UV statistics in a time range from 00:00 for the first day of the conference to the present time.

  ```
  * | select approx_distinct(forward) as uv
  ```

- First change: On the second day of the conference, you need the statistics of PVs of the *yunqi.aliyun. com* domain.

  You can add the following filter condition that starts with host to the statement:

  ```
  host:yunqi.aliyun.com | select approx_distinct(forward) as uv
  ```

- Second change: If the NGINX access logs contain multiple IP addresses, you can enter the following statement to reserve only the first IP address:

  ```
  host:yunqi.aliyun.com | select approx_distinct(split_part(forward,',',1)) as uv
  ```

- Third change: On the third day of the conference, you need to remove statistics that are generated by UC browser advertisement from the access statistics.

  You can add the following filter condition that starts with not to the statement:

  ```
  host:yunqi.aliyun.com not URL:uc-iflow  | select approx_distinct(split_part(forward,',',1
  )) as uv
  ```

1.6. Connect Log Service to Grafana

This topic describes how to use Grafana to display and analyze NGINX logs that are collected by Log Service.

Prerequisites

- NGINX logs are collected. For more information, see Collect logs in NGINX mode.
- The indexing feature is enabled, and indexes are created. For more information, see Collect and analyze NGINX access logs.
- The aliyun-log-grafana-datasource-plugin software package is downloaded. For more information, visit .

  > ⑦ Note   In this topic, aliyun-log-grafana-datasource-plugin V2.9 is used.

- Grafana is installed. For more information, see Grafana documentation.

> ⑦ **Note**
>
> - In this topic, Grafana 8.0.6 is used.
>
> - If you install Grafana on your computer, you must enable port 3000 in your browser settings.
>
> - If you want to use pie charts, you can run the following command to install the Pie Chart plug-in:
>
> ```
> grafana-cli plugins install grafana-piechart-panel
> ```

Version compatibility between Grafana and aliyun-log-grafana-datasource-plugin

The following table describes the version compatibility between Grafana and aliyun-log-grafana-datasource-plugin.

Grafana	aliyun-log-grafana-datasource-plugin
8.0.0 and later	V2.x
Earlier than 8.0.0	V1.0

Step 1: Install the Log Service plug-in

The following procedure describes how to install the Log Service plug-in for Grafana:

1. Run the following commands to decompress the project package to the plug-in directory of Grafana.

   - If Grafana is installed by using a YUM repository or an RPM package, run the following command:

     ```
     unzip aliyun-log-grafana-datasource-plugin-master.zip -d /var/lib/grafana/plugins
     ```

   - If Grafana is installed by using a *.tar.gz* file, run the following command:

     {PATH_TO} specifies the installation directory of Grafana.

     ```
     unzip aliyun-log-grafana-datasource-plugin-master.zip -d {PATH_TO}/grafana-8.0.6/data
     /plugins
     ```

2. Modify the configuration file of Grafana.

   i. Open the configuration file.

      - If Grafana is installed by using a YUM repository or an RPM package, open the */etc/grafana/grafana.ini* file.

      - If Grafana is installed by using a *.tar.gz* file, open the *{PATH_TO}/grafana-8.0.6/conf/defaults.ini* file.

   ii. Find [plugins] in the configuration file to configure the allow_loading_unsigned_plugins parameter.

      ```
      allow_loading_unsigned_plugins = aliyun-log-service-datasource
      ```

3. Restart the Grafana service.

    i. Run the **kill** command to terminate the Grafana process.

    ii. Run the following commands to start the Grafana service:

- If Grafana is installed by using a YUM repository or an RPM package, run the following command:

```
systemctl restart grafana-server
```

- If Grafana is installed by using a *.tar.gz* file, run the following command:

```
./bin/grafana-server web
```

Step 2: Add a data source for Grafana

The following procedure describes how to add the Log Service plug-in as a data source for Grafana:

1. Log on to Grafana.

2. In the left-side navigation pane, choose ⚙ > **Data Sources**.

3. On the **Data Sources** tab, click **Add data source**.

4. On the **Add data source** page, click **Select** in the **LogService** card.

5. Configure the data source.

   The following table describes the parameters.

Parameter	Description
Name	The name of the data source.
Default	The Default switch. In this example, turn on the switch.
Endpoint	The endpoint of the Log Service project. Example: `http://cn-qingdao.log.aliyuncs.com`. Enter an endpoint based on your business requirements. For more information, see Endpoints.
Project	The name of the project.
Logstore	The name of the Logstore.
AccessKeyId	The AccessKey ID provided by Alibaba Cloud. The AccessKey ID is used to identify the user. To ensure the security of your account, we recommend that you use the AccessKey pair of a RAM user. For more information about how to obtain an AccessKey pair, see AccessKey pair.
AccessKeySecret	The AccessKey secret provided by Alibaba Cloud. The AccessKey secret is used to authenticate the key of the user. To ensure the security of your account, we recommend that you use the AccessKey pair of a RAM user. For more information about how to obtain an AccessKey pair, see AccessKey pair.

6. Click **Save & Test**.

Step 3: Add a dashboard

Perform the following steps to add a dashboard for Grafana:

1. In the left-side navigation pane, choose ➕ > **Dashboards**.

2. In the **New Panel** panel, click **Choose Visualization**.

3. Configure template variables.

   After you configure template variables in Grafana, you can select different variable values to view different results in the same panel.

   i. Configure a template variable for a time interval.

      a. In the upper-right corner of the **New dashboard** page, select a time interval and click the ⚙ icon.

      b. Click **Variables**.

      c. Click **Add variable**.

      d. Configure the parameters for the template variable. Then, click **Add**.

         The following table describes the parameters.

Parameter	Description
Name	Enter the name of the template variable. Example: myinterval. If you use a template variable in a query criterion, you must add two dollar signs ($$) preceding the name of the template variable. Example: `$$myinterval` .
Type	Select **Interval**.
Label	Enter **time interval**.
Values	Enter **1m,10m,30m,1h,6h,12h,1d,7d,14d,30d**.
Auto Option	Turn on **Auto Option** and retain the default values of the other parameters.

ii. Configure a template variable for a domain.

    a. On the **Variables** page, click **New**.

    b. Configure the parameters for the template variable. Then, click **Add**. The following table describes the parameters.

Parameter	Description
Name	Enter the name of the template variable. Example: hostname. If you use a template variable in a query criterion, you must add a dollar sign ($) preceding the name of the template variable. Example: `$hostname`.
Type	Select **Custom**.
Label	Enter the name of the domain.
Custom Options	Enter `*,example.com,example.org,example.net` in the "Values separated by comma" field. This value specifies that you can view information about access to all domains. You can also view information about access to the `example.com`, `example.org`, or `example.net` domain.
Selection Options	Retain the default value of the parameter.

iii. In the left-side navigation pane, click **Save**.

4. Add panels based on your business requirements.

   ○ Add a graph panel to display page views (PVs) and unique visitors (UVs).

    a. In the upper-right corner of the page, click the icon.

    b. In the **New Panel** panel, click **Add Query**.

    c. Click the icon and select **Graph** from the **Visualization** drop-down list.

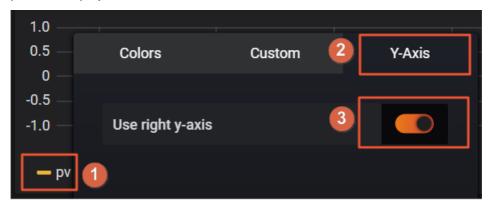   d. Click the icon. Then, enter **UV&PV** in the **Title** field.

e. Click the ⊚ icon, select **Logservice** from the **Query** drop-down list, and then configure the following parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: `$hostname| select approx_distinct(remote_addr) as uv ,count(1) as pv , __time__ - __time__ % $$myinterval as time group by time order by time limit 1000` In the query result, `$hostname` is replaced by the name of the domain that you specify, and `$$myinterval` is replaced by the time interval that you specify. ◁) **Notice** Two dollar signs ($$) must precede myinterval, and one dollar sign ($) must precede hostname.
X-Column	Enter **time**. The value is a UNIX timestamp with second precision.
Y-Column	Enter **uv,pv**.

f. If the difference between the value of UV and the value of PV is significant, use a dual Y-axis panel to display UV and PV statistics.

g. Click the 🖫 icon to save your settings.

○ Add a graph panel to display inbound traffic and outbound traffic.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)." section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ```$hostname	select sum(body_byte_sent) as net_out, sum(request_length) as net_in,__time__ - __time__ % $$myinterval as time group by __time__ - __time__ % $$myinterval limit 10000``` In the query result, `$hostname` is replaced by the name of the domain that you specify, and `$$myinterval` is replaced by the time interval that you specify. ◁) **Notice** Two dollar signs ($$) must precede myinterval, and one dollar sign ($) must precede hostname.
X-Column	Enter **time**. The value is a UNIX timestamp with second precision.
Y-Column	Enter **net_in,net_out**.

○ Add a pie chart panel to display the proportion of requests that are sent by using each HTTP request method.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ```$hostname	select count(1) as pv ,method group by method``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	Enter **pie**.
Y-Column	Enter **method,pv**.

○ Add a pie chart panel to display the proportion of responses that carry each HTTP status code.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ```$hostname	select count(1) as pv ,status group by status``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	Enter **pie**.
Y-Column	Enter **status,pv**.

○ Add a pie chart panel to display the request sources of frequently accessed pages.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ```$hostname	select count(1) as pv , referer group by referer order by pv desc``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	Enter **pie**.
Y-Column	Enter **referer,pv**.

○ Add a table panel to display the pages that are loaded at the highest latencies.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ```$hostname	select URL as top_latency_URL ,request_time order by request_time desc limit 10``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	You do not need to configure this parameter.
Y-Column	Enter **top_latency_url,request_time**.

○ Add a table panel to display frequently accessed pages.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
| Query | Enter the query statement that you want to execute. Example:<br><br>```
$hostname | select count(1) as pv, split_part(URL,'?',1) as
path group by split_part(URL,'?',1) order by pv desc limit
20
```<br><br>In the query result, `$hostname` is replaced by the name of the domain that you specify. |
| X-Column | You do not need to configure this parameter. |
| Y-Column | Enter path,pv. |

○ Add a table panel to display the pages that have the most access requests to which HTTP status code 200 is not returned.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
| Query | Enter the query statement that you want to execute. Example:<br><br>```
$hostname not status:200| select count(1) as pv , url group
by url order by pv desc
```<br><br>In the query result, `$hostname` is replaced by the name of the domain that you specify. |
| X-Column | You do not need to configure this parameter. |
| Y-Column | Enter url,pv. |

○ Add a singlestat panel to display the average latency.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ``` $hostname	select avg(request_time) as response_time, avg(upstream_response_time) as upstream_response_time ,__time__ - __time__ % $$myinterval as time group by __time__ - __time__ % $$myinterval limit 10000 ``` In the query result, `$hostname` is replaced by the name of the domain that you specify, and `$$myinterval` is replaced by the time interval that you specify. > 🔊 **Notice** Two dollar signs ($$) must precede myinterval, and one dollar sign ($) must precede hostname.
X-Column	Enter **time**.
Y-Column	Enter **upstream_response_time,response_time**.

○ Add a logs panel to display the details about logs.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

> ⑦ **Note** Each page can display a maximum of 100 logs. Therefore, the maximum value of the **Logs Per Page** parameter is 100.

○ Add a pie chart panel to display client statistics.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ``` $hostname	select count(1) as pv, case when regexp_like(http_user_agent , 'okhttp') then 'okhttp' when regexp_like(http_user_agent , 'iPhone') then 'iPhone' when regexp_like(http_user_agent , 'Android') then 'Android' else 'unKnown' end as http_user_agent group by http_user_agent order by pv desc limit 10 ``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	Enter **pie**.
Y-Column	Enter **http_user_agent,pv**.

○ Add a graph panel to display the number of requests to which each HTTP status code is

returned over a 1-minute time range.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ``` $hostname	select to_unixtime(time) as time,status,count from (select time_series(__time__, '1m', '%Y-%m-%d %H:%i', '0') as time,status,count(*) as count from log group by status,time order by time limit 10000) ``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	Enter time.
Y-Column	Enter *col1#:#col2*. *col1* is the aggregated column, and *col2* is one of the other columns.

○ Add a worldmap panel to display the distribution of source IP addresses.

For more information about how to add a panel, see the "Add a graph panel to display page views (PVs) and unique visitors (UVs)" section of this topic. The following table describes the parameters.

Parameter	Description
Query	Enter the query statement that you want to execute. Example: ``` $hostname	select count(1) as pv ,geohash(ip_to_geo(arbitrary(remote_addr))) as geo,ip_to_country(remote_addr) as country from log group by country having geo <>'' limit 1000 ``` In the query result, `$hostname` is replaced by the name of the domain that you specify.
X-Column	Enter **map**.
Y-Column	Enter **country,geo,pv**.

Parameter	Description
Location Data	Select **geohash**.
Location Name Field	Enter **country**.
geo_point/geohash Field	Enter **geo**.
Metric Field	Enter **pv**.

5. View the results.

   In the upper section of the Dashboard page, you can select a time interval. You can also filter the results by using the time interval and hostname query criteria.

FAQ

- Where are Grafana logs stored?

  Grafana logs are stored in the following directories:

  - macOS: */usr/local/var/log/grafana*
  - Linux: */var/log/grafana*

- What do I do if **aliyun-log-plugin_linux_amd64: permission denied** appears in logs?

  Grant the EXECUTE permission on the *dist/aliyun-log-plugin_linux_amd64* directory of the plug-in directory.

2.Use Cloud Shell
2.1. Overview of Cloud Shell

Cloud Shell provided by Alibaba Cloud is integrated with Log Service CLI. You can use Cloud Shell to manage Log Service resources and download data from Log Service without the need to deploy or configure Log Service CLI.

What is Cloud Shell?

Cloud Shell allows you to manage Alibaba Cloud resources by running commands. When you start Cloud Shell in a browser, Cloud Shell automatically assigns you a Linux VM that is pre-installed with various cloud management tools and system tools. The VM is provided free of charge. The management tools include Alibaba Cloud CLI and Terraform, and the system tools include SSH, VIM, and jq. For more information, see What is Cloud Shell?

Cloud Shell is integrated with Log Service CLI. You can use Cloud Shell to manage Log Service resources and download data from Log Service without the need to download or install Log Service CLI.

If Cloud Shell detects no input within 30 minutes, the VM that is assigned to you is released. This helps improve resource utilization. If you want to use the features of Cloud Shell for a long period of time, you can download and deploy Log Service CLI. For more information, see Overview of Log Service CLI.

Scenarios of Cloud Shell

Scenario	Description
Use Cloud Shell to manage Log Service resources	Use Cloud Shell to manage resources. For example, create projects, create Logstores, and query logs.
Use Cloud Shell to download logs from Log Service	Use Cloud Shell to download logs.

2.2. Use Cloud Shell to manage Log Service resources

Cloud Shell is integrated with Log Service CLI. Cloud Shell allows you to manage Log Service resources and data by running commands. The commands are in a unified format, which facilitates your management.

Context

This topic provides examples on how to manage the following Log Service resources:

- Projects. For example, you can create, query, or delete a project.
- Logstores. For example, you can create, query, modify, or delete a Logstore.
- Consumer groups, shards, machine groups, and data shipping tasks. For example, you can create a data shipping task.

Step 1: Start Cloud Shell

1.

2. In the upper-right corner of the page, click the ⌷ icon.

3. Follow the on-screen instructions to complete the verification.

4. If you want to persistently store files and scripts that are commonly used, click **Create Now** in the **Storage Space** dialog box.

   You can associate and mount an Apsara File Storage NAS (NAS) file system to persistently store files and scripts that are commonly used. This way, the files are not deleted when the file system is released. Cloud Shell automatically creates a Performance NAS file system that uses the pay-as-you-go billing method. You are charged for the usage of the file system. For more information, see Billing of General-purpose NAS file systems.

   > ⑦ **Note**   When you connect to Cloud Shell for the first time, Cloud Shell automatically creates a VM for you within 30 seconds. If you open multiple Cloud Shell windows, all the windows are connected to the same VM. The number of VMs does not increase when you open a new Cloud Shell window.

   After Cloud Shell is started, a Cloud Shell window appears.

   ```
   Welcome to Alibaba Cloud Shell!
   Type "help" to learn about Cloud Shell
   Type "aliyun" to use Alibaba Cloud CLI
   You may be interested in these tutorials below.
   -------------------------  |  -------------------------
   Download logs from Log Service            |  cloudshell://tutorial/sls-download-lo
   g
   Use Alibaba Cloud CLI to manage cloud resources |  cloudshell://tutorial/aliyun-cli
   For more tutorials, visit https://api.aliyun.com/#/lab
   shell@Alicloud:~$
   ```

Step 2: Use CLI commands to manage Log Service resources

Cloud Shell is integrated with Log Service CLI. You can use Cloud Shell to manage log data in Log Service without the need to configure Log Service CLI.

You can use Log Service CLI subcommands to perform various operations. For example, you can create a project or a Logstore. For more information, see Overview of Log Service CLI.

- Syntax:

  ```
  aliyunlog log <subcommand> [parameters | global options]
  ```

  - subcommand: the supported subcommand. For more information, see Commands.
  - parameters: the supported parameters. For more information about the parameters that are supported for each subcommand, see the description of each subcommand.
  - global options: the supported global parameters. For more information, see Global parameters.

> ⑦ **Note**
>
> - When you manage Log Service resources that belong to a different region from Cloud Shell, you must change the region. You can configure `--region-endpoint` to specify the region that you want to use.
> - The Cloud Shell server is located in the China (Shanghai) region. If you download logs from a Logstore that does not reside in the China (Shanghai) region, you are charged for data transfers over the Internet. For more information, see Log Service pricing.

- Examples
  - Create a project.

    ```
    aliyunlog log create_project --project_name="project-a" --project_des="project created
    from cli"
    ```

  - Create a Logstore.

    ```
    aliyunlog log create_logstore --project_name="aliyun-test-project" --logstore_name="log
    store-a"
    ```

  - Query logs.

    ```
    aliyunlog log get_log_all --project="aliyun-test-project" --logstore="logstore-a" --fro
    m_time="2021-05-28 15:33:00+8:00" --to_time="2021-05-28 15:40:00+8:00" --query="level:I
    nformation|select event_id as Key1,COUNT(*) as Key2 group by Key1" --reverse=true
    ```

Step 3: (Optional) Obtain help information

To obtain the help information about a subcommand, use the following syntax:

```
aliyunlog log [subcommand] help
```

For example, you can run the following command to view the usage description of the list_project subcommand:

```
aliyunlog log list_project help
```

References

- Use Cloud Shell
- Overview of Log Service CLI

2.3. Use Cloud Shell to download logs from Log Service

You can use Cloud Shell to download a large number of log files from Log Service and upload the files to Object Storage Service (OSS) within a short period of time. This topic describes how to use Cloud Shell to download logs.

Procedure

1.

2.

3.

4. In the search box, enter a query statement. Then, select a time range and click **Search & Analyze**.

5. On the **Raw Logs** tab, click the ⬇ icon.

6. In the **Log Download** dialog box, select **Download All Logs with Cloud Shell** and click **OK**.

> ? **Note**   The Cloud Shell server is located in the China (Shanghai) region. If you download logs from a Logstore that does not reside in the China (Shanghai) region, you are charged for data transfers over the Internet. For more information, see Log Service pricing.

7. In the **Download File** message, click **OK**.

8. (Optional)In the Cloud Shell window, upload the downloaded log file to an OSS bucket.

    i. Run the following command to view the OSS buckets that you can manage so that you can determine in which OSS bucket you can store the log file:

```
aliyun oss ls
```

If the command is successfully run, the system returns a list of OSS buckets that you can manage.

```
CreationTime                            Region     StorageClass    BucketName
2021-09-24 02:03:08 +0000 UTC      oss-cn-beijing        Standard    oss://fyytse
t
2021-09-26 03:27:10 +0000 UTC      oss-cn-hangzhou       Standard    oss://demo
Bucket Number is: 2
```

    ii. Go to the *aliyunlog* directory and run the `ll` command to find the downloaded log file.

```
total 8640
drwxr-xr-x 2 shell shell   4096 Sep 28 10:21 ./
drwxr-xr-x 6 shell shell   4096 Sep 28 10:24 ../
-rw-r--r-- 1 shell shell 3198090 Sep 28 10:03 data_20210928_100331.txt
```

    iii. Upload the log file to the OSS bucket.

```
aliyun oss cp data_20210928_100331.txt oss://demo --region cn-hangzhou
```

If the following information is returned, the file is uploaded:

```
Succeed: Total num: 1, size: 3,198,090. OK num: 1(upload 1 files).
average speed 12792000(byte/s)
0.250823(s) elapsed
```