ALIBABA CLOUD

Alibaba Cloud

Log Service

Data shipping

Document Version: 20201012

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.Overview 05
2.Ship logs to OSS 06
2.1. Ship log data from Log Service to OSS 06
2.2. JSON-formatted data 10
2.3. CSV-formatted data 11
2.4. Parquet-formatted data 12
2.5. Decompression tools for Snappy compressed files 15
2.6. Authorization in the RAM console 16
3.Send logs to an SIEM system 21
3.1. Introduction 21
3.2. Send logs to an SIEM system over HTTPS 22
3.3. Send logs to an SIEM system over Syslog 28
3.4. Ship data to Splunk by using the Splunk add-on for Lo... 34
4.Ship log data from Log Service to TSDB 44
5.Best practices 47
5.1. Connect to a data warehouse 47

1.Overview

Log Service provides the data shipping and consumption features. You can ship log data to Alibaba Cloud services such as OSS and MaxCompute in real time by using the Log Service console. You can also use an SDK or the API of Log Service to consume log data in real time. This topic describes the benefits, scenarios, and shipping destinations of the data shipping feature.

In the Log Service console, you can ship log data to other Alibaba Cloud services. Then, you can store or consume the log data by using other systems such as E-MapReduce. After you enable the log shipping feature, Log Service ships the collected log data to the specified cloud service at regular intervals.

Scenarios

The log shipping feature can be used to connect Log Service with data warehouses.

Benefits

Using the Log Service console to ship logs has the following benefits:

- Ease of use

  You can specify the log shipping settings in the Log Service console. Then you can use the settings to ship log data from Logstores to other Alibaba Cloud services such as OSS.

- High efficiency

  Log Service stores log data that is collected from multiple servers. This improves efficiency when you ship log data to other Alibaba Cloud services such as OSS.

- Effective management

  You can ship log data from different projects or Logstores to different OSS buckets or MaxCompute tables. This facilitates log data management.

Log shipping destinations

- OSS

  For information about how to ship log data to OSS, see Ship log data from Log Service to OSS.

  - We recommend that you use E-MapReduce to convert the data type of the shipped log data into the data type that OSS supports.
  - After you ship log data to OSS, you can use Data Lake Analytics (DLA) to analyze the data.

- MaxCompute

  For information about how to ship data to MaxCompute by using the data integration feature of DataWorks, see Use Data Integration to ship data collected by LogHub to destinations.

- SIEM

  For information about how to ship data to a security information and event management (SIEM) system by using the log shipping feature of Log Service, see Ship data from Log Service to SIEM.

- Table Store

  For information about how to ship data to Table Store by using the transfer feature of Table Store, see The transfer feature of Table Store.

2.Ship logs to OSS

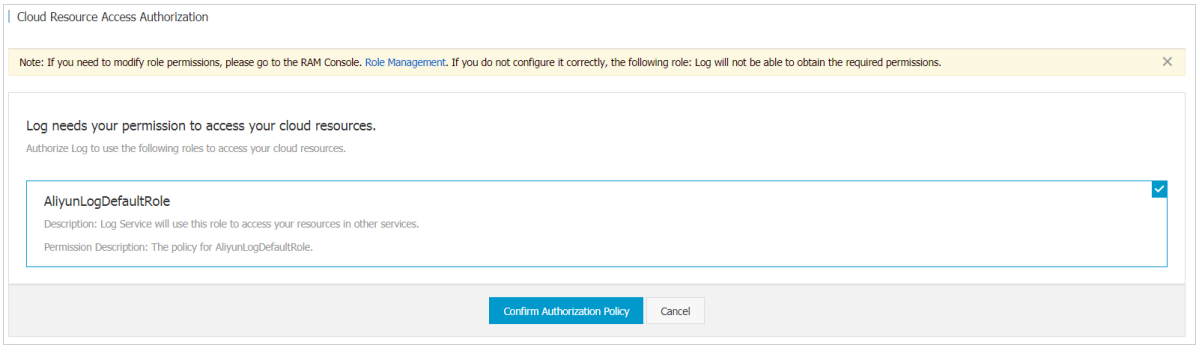
2.1. Ship log data from Log Service to OSS

You can use Log Service to collect log data and ship the log data to Object Storage Service (OSS) for storage and analysis. This topic describes how to ship log data from Log Service to OSS.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and a Logstore.
- Log data is collected. For more information, see Log collection.
- OSS is activated. A bucket is created in the region where the Log Service project resides. For more information, see Activate OSS.
- Log Service is authorized to access cloud resources in OSS. You can authorize Log Service on the Cloud Resource Access Authorization page.

  To ship log data across Alibaba Cloud accounts or by using a RAM user, see Authorization in the RAM console.



Context

You can ship log data from Log Service to OSS for storage and consumption.

- You can set a custom retention period for log data in OSS. Permanent retention is supported.
- You can use data processing platforms such as E-MapReduce and Data Lake Analytics (DLA) or use custom programs to consume log data in OSS.

Procedure

1. Log on to the Log Service console.
2. On the page that appears, click the source project in the **Projects** section.
3. On the page that appears, choose **Log Management** > **Logstores**. In the Logstore list, click the > icon of the source Logstore, and then choose **Data Transformation** > **Export** > **Object Storage Service (OSS)**.
4. On the OSS Shipper page, click **Enable**.
5. In the **Shipping Notes** dialog box, click **Ship**.
6. Configure a rule for shipping log data to OSS. The following table describes the configuration parameters.

Parameter	Description
OSS Shipper Name	The name of the shipping rule. The name can contain only lowercase letters, digits, hyphens (-), and underscores (_). It must start and end with a lowercase letter or digit and must be 3 to 63 characters in length.

Parameter	Description
OSS Bucket	The name of the destination OSS bucket. Note You must specify the name of an existing OSS bucket. The specified OSS bucket must reside in the same region as the source project.
OSS Prefix	The destination directory in the OSS bucket.
Shard Format	The partition format of the directory of a shipping task. The default format is %Y/%m/%d/%H/%M. The value of this parameter cannot start with a forward slash (/). For information about partition format examples, see Partition format. For information about parameters of the partition format, visit strptime API.
RAM Role	The ARN of the RAM role assumed by Log Service. The RAM role is used to control access to the specified OSS bucket. Example: acs:ram::45643:role/aliyunlogdefaultrole. For more information, see Obtain the ARN of a RAM role.
Shipping Size	The maximum size of raw log data that can be shipped to the OSS bucket in a shipping task. Valid values: 5 to 256. Unit: MB. If the size of shipped data exceeds the specified value, a new shipping task is created.
Storage Format	The storage format of log data in OSS. Data shipped to OSS can be in the JSON, CSV, or Parquet format. For more information, see CSV-formatted data.
Compress	Specifies whether to compress log data that is shipped to OSS. ○ No Compress: Log data that is shipped to OSS is not compressed. ○ Compress (snappy): The Snappy utility is used to compress the log data that is shipped to OSS. This frees more storage space of the OSS bucket.
Ship Tags	Specifies whether to ship log tags.
Shipping Time	The time interval between two successive shipping tasks. Valid values: 300 to 900. Default value: 300. Unit: seconds. If the specified interval is reached, a new shipping task is created.

7. Click **OK**.

Note
- After you specify a shipping rule, multiple shipping tasks are concurrently running. A new task is created if the size of data shipped from a shard reaches the specified threshold or the specified shipping interval is reached.
- After shipping tasks are created, you can check whether the shipping rule satisfies your business requirements based on the task status and the data shipped to OSS.

View OSS data

After log data is shipped to OSS, you can view the log data in the OSS console, by using an API or SDK, or by using another method. For more information, see OSS bucket management.

The following example is a sample OSS directory:

```
oss:// OSS-BUCKET/OSS-PREFIX/PARTITION-FORMAT_RANDOM-ID
```

OSS-BUCKET specifies the name of the destination OSS bucket. OSS-PREFIX specifies the prefix of the directory in the destination OSS bucket. PARTITION-FORMAT specifies the partition format of the directory of a shipping task. The partition format is calculated based on the creation time of a shipping task. For more information, see strptime API. RANDOM-ID is the unique identifier of a shipping task.

> **Note** A directory in an OSS bucket is created based on the creation time of a shipping task. For example, a shipping task is created at 00:00:00 on June 23, 2016 to ship the data that is written to Log Service after 23:55:00 of June 22, 2016. The shipping interval is 5 minutes. To retrieve all logs shipped on June 22, 2016, you must check all objects in the *2016/06/22/00/* directory. You must also check the *2016/06/23/00/* directory for objects that are generated in the first 10 minutes of June 23, 2020

Partition format

For each shipping task, log data is written to a directory of an OSS object. A directory is in the *oss:// OSS-BUCKET/OSS-PREFIX/PARTITION-FORMAT_RANDOM-ID* format. A partition format is obtained by formatting the creation time of a shipping task. The following table describes the partition formats and directories when a shipping task is created at 19:50:43 on January 20, 2017.

OSS Bucket	OSS Prefix	Partition format	OSS directory
test-bucket	test-table	%Y/%m/%d/%H/%M	*oss://test-bucket/test-table/2017/01/20/19/50_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/_14849130433515253 51_2850008*
test-bucket	log_ship_oss_example	%Y%m%d%H	*oss://test-bucket/log_ship_oss_example/2017012019_1484913043351525 351_2850008*

To use the time information in partition formats when you use Hive, MaxCompute, or DLA to analyze OSS data, you can set partition formats in key-value pairs, for example, *oss://test-bucket/log_ship_oss_example/year=2017/mon=01/day=20/log_195043_1484913043351525351_2850008.parquet*. In this example, the three partition keys are specified: year, mon, and day.

Related operations

After shipping tasks are created, you can modify the rule of the tasks, disable the tasks, view the status and error messages of the tasks, and retry failed tasks on the **OSS Shipper** page.

- Modify the shipping rule of tasks

  Click **Settings** to modify the shipping rule of tasks. For information about parameters, see Procedure.

- Disable shipping tasks

  Click **Disable**. All tasks are disabled.

- View the status of tasks and error messages

  You can view the status of all log shipping tasks that were performed in the last two days.

  - Status of a shipping task

State	Description
Succeeded	The shipping task succeeded.
Running	The shipping task is in progress. Check whether the task succeeds at a later time.
Failed	The shipping task failed. If the task failed and cannot be restarted because of external causes, troubleshoot the failure based on the error message and retry the task.

  - Error message

    If a shipping task failed, an error message is returned for the task.

Error message	Description	Solution
UnAuthorized	The error message returned because permissions are not granted to the AliyunLogDefaultRole role.	Troubleshoot the error by checking the following configurations: ■ The AliyunLogDefaultRole role must be created for the Alibaba Cloud account to which the destination OSS bucket belongs. Check whether the AliyunLogDefaultRole role is created. ■ A permission policy must be attached to the AliyunLogDefaultRole role. Check whether the Alibaba Cloud account ID configured in the permission policy is valid. ■ Check whether the AliyunLogDefaultRole role is granted write permissions on the destination OSS bucket. ■ Check whether the ARN of the AliyunLogDefaultRole role that you entered in the RAM Role field is correct.
ConfigNotExist	The error message returned because the task does not exist.	Check whether the task is disabled. Enable the data shipping feature, configure a shipping rule for the task, and then retry the task.
InvalidOssBucket	The error message returned because the destination OSS bucket does not exist.	Troubleshoot the error by checking the following configurations: ■ Check whether the destination OSS bucket resides in the same region as the source project. ■ Check whether the specified bucket name is correct.
InternalServerError	The error message returned because an internal error has occurred in Log Service.	Retry the failed task.

- Retry a task

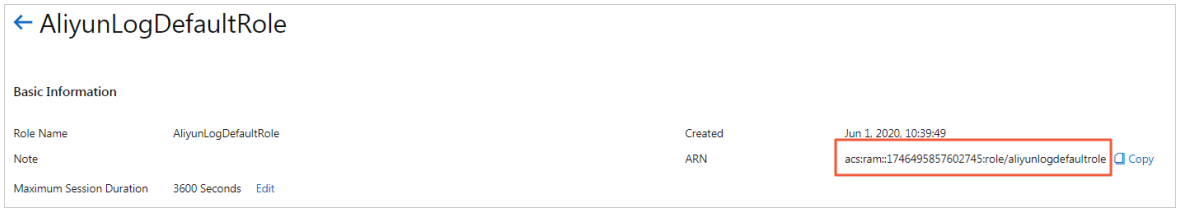
  If a task fails, Log Service retries the task by default. You can also manually retry the task. By default, Log Service retries tasks that failed in the past two days. The minimum interval between retries is 15 minutes. If a task fails for the first time, Log Service retries the task 15 minutes later. If the task fails for the second time, Log Service retries the task 30 minutes later. If the task fails for the third time, Log Service retries the task 60 minutes later.

  To retry a failed task, you can click **Retry All Failed Tasks** or **Retry** in the Actions column. You can also use an API or SDK to retry a task.

FAQ

How do I obtain the ARN of a RAM role?

1. To obtain the ARN of a RAM role, log on to the RAM console.
2. In the left-side navigation pane, click **RAM Roles**.
3. On the RAM Roles page, click the role named AliyunLogDefaultRole.
4. On the page that appears, obtain the ARN of the role in the **Basic Information** section.



2.2. JSON-formatted data

This topic describes how to ship data from Log Service to OSS and store the data in the JSON format.

You can set the storage format of data that is shipped to OSS. The following table shows how to set the storage format to JSON. For more information, see Configure a data shipping rule.

Compressed	File extension	Example	Description
No	N/A	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937	You can download the raw JSON object to the local host and open the object as a text file. The following example is a sample file: {"__time__":1453809242,"__topic__":"","__source__":"10.170. ***.***","ip":"10.200. **.***","time":"26/Jan/2016:19:54:02 +0800","url":"POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature> HTTP/1.1","status":"200","user-agent":"aliyun-sdk-java"}

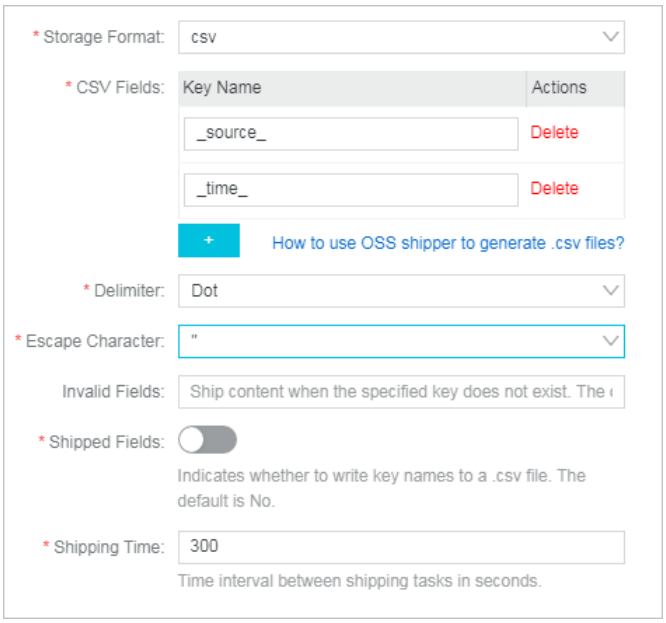
Compressed	File extension	Example	Description
snappy	.snappy	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.snappy	For more information about the Snappy utility, see Decompression tools for Snappy compressed files.

2.3. CSV-formatted data

This topic describes how to ship data from Log Service to OSS and store the data in the comma-separated values (CSV) format.

Configuration parameters

You can set the storage format of data that is shipped to OSS. The following table shows how to set the storage format to CSV. For more information, see Configure a data shipping rule.



The following table describes the configuration parameters in the preceding figure. For more information, visit Common Format and MIME Type for CSV Files or CSV Format in PostgreSQL.

Parameter	Description
CSV Fields	The names of the log fields that you want to ship to OSS. You can view log fields on the **Raw Logs** tab of a Logstore and enter the names of the fields that you want to ship to OSS in the Key Name column. The log fields that you can ship to OSS include the fields in the log content and the reserved fields such as __time__, __topic__, and __source__. For more information about reserved fields, see Reserved fields. **Note** The keys that you enter in the **CSV Fields** field must be unique.
Delimiter	You can use commas (,), vertical bars (|), spaces, or tabs to delimit fields.
Escape Character	If a field contains a delimiter, you must use an escape character to enclose the field. This ensures that the field is not delimited.
Invalid Fields	If a key that you specify in the **CSV Fields** field does not exist, enter the value of the key in the Invalid Fields field.

Parameter	Description
Shipped Fields	If you turn on the **Shipped Fields** switch, field names are written in the CSV file.

Directories in OSS buckets

The following table lists the directories in OSS buckets that store data shipped from Log Service.

Compressed	File extension	Example	Description
No	.csv	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.csv	You can download the raw JSON object to the local host and open the object as a text file.
snappy	.snappy.csv	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.snappy.csv	For more information about the Snappy utility, see Decompression tools for Snappy compressed files.

Examples

To use HybridDB for MySQL or HybridDB for PostgreSQL to consume CSV files in OSS, set the fields as described in the following list:

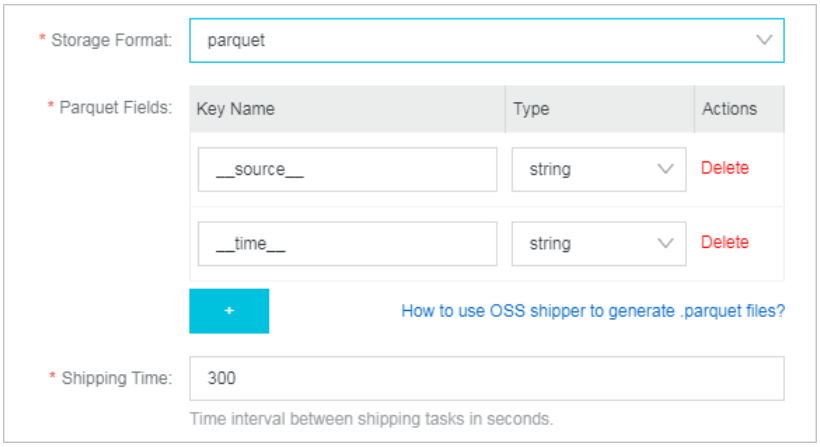
- Delimiter: Select comma (,).
- Escape Character: Select double quotation marks (").
- Invalid Fields: Null.
- Shipped Fields: Turn off the switch. By default, no key name is added in the first line of the CSV file in HybridDB for MySQL or HybridDB for PostgreSQL.

2.4. Parquet-formatted data

This topic describes how to ship data from Log Service to OSS and store the data in the Parquet format.

Configuration parameters

You can set the storage format of data that is shipped to OSS. The following figure shows how to set the storage format to Parquet. For more information, see Configure a data shipping rule.



The following table describes the configuration parameters in the preceding figure.

Parameter	Description

Parameter	Description
Key Name	The name of the log field that you want to ship to OSS. You can view log fields on the **Raw Logs** tab of a Logstore. You can also enter the names of the fields that you want to ship to OSS in the Key Name column. When the fields are shipped to OSS, they are stored in the Parquet format in the order that the field names are entered. The names of the fields are the column names in OSS. The log fields that you can ship to OSS include the fields in the log content and the reserved fields such as __time__, _topic__, and __source__. For more information about reserved fields, see Reserved fields. The value of a field in the Parquet format is null in the following two scenarios: • The field does not exist in logs. • The value of the field fails to be converted from the string type to a non-string type, for example, double or Int64. Note The keys that you enter in the **Parquet Keys** field must be unique.
Type	The Parquet storage format supports six data types: string, Boolean, Int32, Int64, float, and double. Log fields are converted from the string type to a data type that the Parquet storage format supports. If the data type of a log field fails to be converted, the value of the log field is null.

Directories in OSS buckets

The following table lists the directories in OSS buckets that store data shipped from Log Service.

Compressed	File extension	Example	Description
No	.parquet	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.parquet	After you download the OSS buckets to the local server, you can use the parquet-tools utility to open the buckets. For more information about the parquet-tools utility, visit parquet-tools.
Yes (compressed by using Snappy)	.snappy.parquet	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_937.snappy.parquet	After you download the OSS buckets to the local server, you can use the parquet-tools utility to open the buckets. For more information about the parquet-tools utility, visit parquet-tools.

Data consumption

- You can use E-MapReduce, Spark, and Hive to consume data. For more information, visit LanguageManual DDL.
- You can use inspection tools to consume data.

  The parquet-tools utility can be used to inspect Parquet files, view the schema of the data stored in the files, and read the data. You can compile the utility or download the parquet-tools-1.6.0rc3-SNAPSHOT utility that Log Service provides to consume data.

- To view the schema of the data stored in a Parquet file, use the following sample code:

```
$ java -jar parquet-tools-1.6.0rc3-SNAPSHOT.jar schema -d 00_1490803532136470439_124353.snappy.parquet | hea
d -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
.......
```

- To view the data stored in a Parquet file, use the following sample code:

```
$ java -jar parquet-tools-1.6.0rc3-SNAPSHOT.jar head -n 2 00_1490803532136470439_124353.snappy.parquet
__time__ = 1490803230
ip = 10.200.98.220
__source__ = *. *. *.*
method = POST
__topic__ =
seq = 1667821.0
status = 200
time = 30/Mar/2017:00:00:30 +0800
url = /PutData? Category=YunOsAccountOpLog&AccessKeyId=**********&Date=Fri%2C%2028%20Jun%202013%200
6%3A53%3A30%20GMT&Topic=raw&Signature=********************************** HTTP/1.1
__time__ = 1490803230
ip = 10.200.98.220
__source__ = *. *. *.*
method = POST
__topic__ =
seq = 1667822.0
status = 200
time = 30/Mar/2017:00:00:30 +0800
url = /PutData? Category=YunOsAccountOpLog&AccessKeyId=**********&Date=Fri%2C%2028%20Jun%202013%200
6%3A53%3A30%20GMT&Topic=raw&Signature=********************************** HTTP/1.1
```

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

2.5. Decompression tools for Snappy compressed files

If you use Snappy to compress data when you ship data from Log Service to OSS, you can use decompression tools to decompress the data. This topic describes the tools that you can use to decompress Snappy compressed OSS buckets. The decompression tools include Snappy decompressor for C++, Snappy decompressor for Java, Snappy decompressor for Python, and Linux-based Snappy decompressor.

Snappy decompressor for C++

Download the C++ library from the snappy page and use the Snappy.Uncompress method to decompress Snappy compressed OSS buckets.

Snappy decompressor for Java

Download the Java library from the xerial snappy-java page and use the Snappy.Uncompress or Snappy.SnappyInputStream method to decompress Snappy compressed OSS buckets. The SnappyFramedInputStream method is not supported.

> **Note** If you use Snappy decompressor for Java 1.1.2.1, some Snappy compressed OSS buckets may fail to be decompressed. For information about the possible exceptions, visit Bad handling of the MAGIC HEADER. To avoid this issue, you can use Snappy decompressor for Java 1.1.2.6 or later.

```
<dependency>
<groupId>org.xerial.snappy</groupId>
<artifactId>snappy-java</artifactId>
<version>1.0.4.1</version>
<type>jar</type>
<scope>compile</scope>
</dependency>
```

- 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));
}
```

Snappy decompressor for Python

1. Download and install Snappy decompressor for Python.
2. Run the decompression script.

   The following example is a sample decompression script:

```
import snappy
compressed = open('/tmp/temp.snappy').read()
snappy.uncompress(compressed)
```

> **Note** The following two commands cannot be used to decompress Snappy compressed OSS buckets. These commands can be used only in Hadoop mode (hadoop_stream_decompress) or streaming mode (stream_decompress).
>
> ```
> $ python -m snappy -c uncompressed_file compressed_file.snappy
> $ python -m snappy -d compressed_file.snappy uncompressed_file
> ```

Linux-based Snappy decompressor

Log Service allows you to decompress Snappy compressed files on a Linux-based server. Click snappy_tool to download the decompressor. Replace 03_1453457006548078722_44148.snappy and 03_1453457006548078722_44148 in the following code with the values specific to your environment and then run the following code:

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

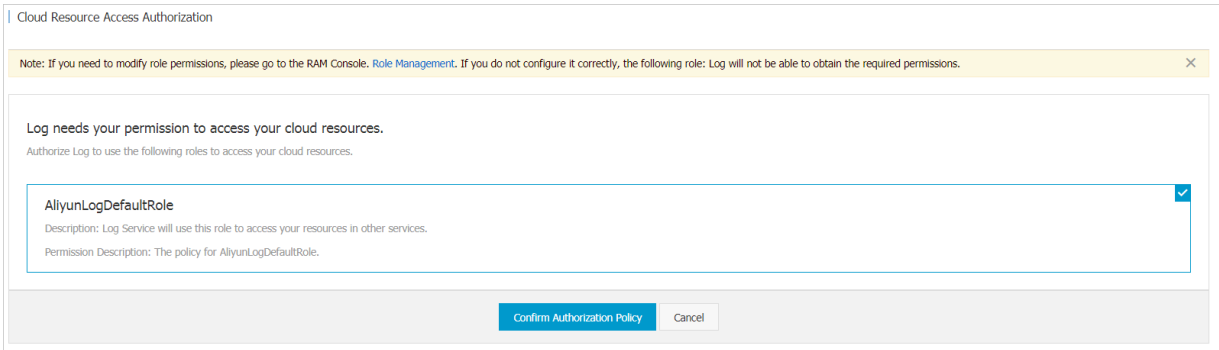
2.6. Authorization in the RAM console

You can use a RAM user to ship log data from Log Service to OSS. You can also ship log data from Log Service of an Alibaba Cloud account (Alibaba Cloud Account A) to Object Storage Service (OSS) of another Alibaba Cloud account (Alibaba Cloud Account B). Before you ship log data in the two scenarios, you must use RAM to authorize the RAM user or Alibaba Cloud accounts. This topic describes how to use RAM to grant required permissions in the two scenarios.

Prerequisites

Log Service is authorized to access the destination OSS bucket. For more information, see Cloud resource access authorization.

After the authorization is completed, Log Service can assume a role by using STS and write data to the destination OSS bucket.



Overview

When you ship log data in different scenarios, you must use RAM to grant relevant permissions.

- For information about how to grant finer-grained access to an OSS bucket, see Modify a permission policy.
- For information about how to ship log data from a Log Service project of Alibaba Cloud Account A to an OSS bucket of Alibaba Cloud Account B, see Ship log data across Alibaba Cloud accounts.
- For information about how to use a RAM user to ship log data from a Log Service project to an OSS bucket of the same Alibaba Cloud account, see Use a RAM user to ship log data from Log Service to OSS of the same Alibaba Cloud account.
- For information about how to use a RAM user to ship log data from a Log Service project of Alibaba Cloud Account A to an OSS bucket of Alibaba Cloud Account B, see Use a RAM user to ship log data across Alibaba Cloud accounts.

Modify a permission policy

After you implement cloud resource access authorization for Log Service, the AliyunLogRolePolicy policy is attached to the AliyunLogDefaultRole role. Log Service assumes the role and thus is authorized to ship log data to each OSS bucket. To implement finer-grained access control, you can detach the AliyunLogDefaultRole policy from the AliyunLogRolePolicy role, and attach a custom policy to the AliyunLogDefaultRole role. For more information, see Implement access control based on RAM policies.

Ship log data across Alibaba Cloud accounts

To ship log data from a Log Service project of Alibaba Cloud Account A to an OSS bucket of Alibaba Cloud Account B, you must use RAM to authorize the two Alibaba Cloud accounts. The following procedure describes how to implement the authorization.

1. Create a role named AliyunLogDefaultRole for Alibaba Cloud Account B on the cloud resource access authorization page.
2. Log on to the RAM console by using Alibaba Cloud Account B.
3. In the left-side navigation pane, click **RAM Roles**.
4. On the RAM Roles page, click the role named AliyunLogDefaultRole in the RAM Role Name column.
5. On the page that appears, click the **Trust Policy Management** tab. On this tab, click **Edit Trust Policy**.

   In the Edit Trust Policy dialog box, add a data entry in the format of Alibaba Cloud account ID@log.aliyuncs.com in the **Service** field. The Alibaba Cloud account ID is the ID of Alibaba Cloud Account A. To view the ID of Alibaba Cloud Account A, log on to the RAM console by using Alibaba Cloud Account A. On the page that appears, click the profile picture in the upper-right corner. On the page that appears, choose **Account Management** > **Security Settings**. The ID of the Alibaba Cloud Account A is displayed on the Security Settings page. This policy indicates that Log Service of Alibaba Cloud Account A is authorized to manage the cloud resources of Alibaba Cloud Account B by using a temporary STS token.

```
{
"Statement": [
 {
  "Action": "sts:AssumeRole",
  "Effect": "Allow",
  "Principal": {
   "Service": [
    "The ID of Alibaba Cloud Account A@log.aliyuncs.com",
    "log.aliyuncs.com"
   ]
  }
 }
],
"Version": "1"
}
```

6. Obtain the ARN of the RAM role. In the **Basic Information** section of the AliyunLogDefaultRole page, view the ARN of the AliyunLogDefaultRole role, for example, acs:ram::13234:role/logrole.

   When you configure a log shipping task, enter this ARN in the **RAM Role** field.

Use a RAM user to ship log data across Alibaba Cloud accounts

To use RAM User A1 of Alibaba Cloud Account A to ship log data from Log Service to an OSS bucket of Alibaba Cloud Account B, you must attach the PassRole policy to RAM User A1.

1. Complete the configurations for Alibaba Cloud Account B, as described in Ship log data across Alibaba Cloud accounts.
2. Log on to the RAM console by using Alibaba Cloud Account A.
3. Create a RAM user named RAM User A1. For more information, see Create a RAM user.
4. Attach the AliyunRAMFullAccess policy to RAM User A1.
   i. In the left-side navigation pane, choose **Identities** > **Users**.
   ii. On the Users page, find RAM User A1, and click **Add Permissions** in the Actions column.

iii. In the Add Permissions dialog box, click the **System Policy** tab under the Select Policy field. In the Authorization Policy Name list, click **AliyunRAMFullAccess**. The policy appears in the Selected column. Then, click **OK**.

After this policy is attached to RAM User A1, RAM User A1 is granted full access to RAM.

To grant permissions only on OSS to RAM User A1, you can attach a custom policy to RAM User A1. The following example is a sample script of a custom policy. The value of the Resource field is the ARN of the role named AliyunLogDefaultRole of Alibaba Cloud Account B. For more information about how to create a custom policy, see Use a RAM user to ship log data from Log Service to OSS of the same Alibaba Cloud account.

```
{
"Statement": [
{
"Action": "ram:PassRole",
"Effect": "Allow",
"Resource": "acs:ram::1111111:role/aliyunlogdefaultrole"
}
],
"Version": "1"
}
```

5. Obtain the ARN of the RAM role. In the **Basic Information** section of the AliyunLogDefaultRole page, view the ARN of the AliyunLogDefaultRole role, for example, acs:ram::13234:role/logrole.

   When you use RAM User A1 to configure a log shipping task, enter this ARN in the **RAM Role** field.

Use a RAM user to ship log data from Log Service to OSS of the same Alibaba Cloud account

To use a RAM user to create log shipping tasks, you must use your Alibaba Cloud account to grant the relevant permissions to the RAM user.

1. Log on to the RAM console by using an Alibaba Cloud account.
2. Create a permission policy.
   i. In the left-side navigation pane, choose **Permissions > Policies**.
   ii. On the Policies page, click **Create Policy**.

iii. On the **Create Custom Policy** page, set the parameters, and then click **OK**. The following table describes the parameters.

Parameter	Description
Policy Name	The name of the policy.
Configuration Mode	Select **Script**.
Policy Document	The content of the policy. Replace the content in the editor with the following script: **Note** The policy must include the PassRole permission. `{ "Version": "1", "Statement": [ { "Effect": "Allow", "Action": "log:*", "Resource": "*" }, { "Effect": "Allow", "Action": "ram:PassRole", "Resource": "*" } ] }`

3. Create a RAM user. For more information, see Create a RAM user.
4. Authorize the RAM user.
   i. In the left-side navigation pane, choose **Identities** > **Users**.
   ii. On the Users page, find the RAM user, and click **Add Permissions** in the Actions column.
   iii. Click the **Custom Policy** tab under the Select Policy field, select the policy that you created in Step 2, and then click **OK**.

3.Send logs to an SIEM system

3.1. Introduction

Log Service allows for sending logs to a security information and event management (SIEM) system. This ensures that all logs related to regulations and audits on Alibaba Cloud can be imported to your security operations center (SOC).

Terms

- **SIEM**: security information and event management (SIEM) systems, such as Splunk and IBM QRadar.
- **Splunk HEC**: Splunk HTTP Event Collector (HEC) can be used to receive and send logs over HTTP or HTTPS.

Deployment suggestions

- Hardware specifications:
  - Operating system: Linux, such as Ubuntu x64.
  - CPU: 2.0+ GHz x 8 cores.
  - Memory: 32 GB (recommended) or 16 GB.
  - Network interface controller (NIC): 1 Gbit/s.
  - Available disk space: at least 2 GB. We recommend that you have an available disk space of 10 GB or greater.
- Network specifications:

  The bandwidth between your network environment and Alibaba Cloud must be greater than the speed at which data is generated on Alibaba Cloud. Otherwise, logs cannot be consumed in real time. Assume that the peak speed for data generation is about twice that of the average speed and 1 TB of raw logs are generated every day. If data is compressed at a ratio of 5:1 before transmission, we recommend that you use a bandwidth of around 4 MB/s (32 Mbit/s).
- Python: You can use Python to consume logs. For more information about using Java, see Use consumer groups to consume log data.

Python SDK

- We recommend that you use a standard CPython interpreter.
- You can run the **python3 -m pip install aliyun-log-python-sdk -U** command to install the Log Service SDK for Python.
- For more information about how to use the Log Service SDK for Python, see User Guide.

Consumer library

The consumer library is an advanced log consumption mode in Log Service. The consumer library provides consumer groups to facilitate consumer management. In comparison to reading data by using the SDK, you can focus on the business logic rather than worrying about the implementation details of Log Service. In addition, the consumer library allows you to ignore failover and load balancing between consumers.

In Log Service, a Logstore can have multiple shards. The consumer library is used to allocate shards to consumers in a consumer group. The allocation rules are described 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 affiliation of shards with this consumer group is adjusted to balance consumption loads. However, the preceding allocation rules still apply and you cannot view the allocation details of shards.

The consumer library can also store checkpoints, which allows you to consume data starting from a breakpoint after a program crash is fixed. This ensures that the data is consumed only once.

Spark Streaming, Storm, and Flink Connector are all implemented based on the consumer library.

Log sending methods

We recommend that you write the required program based on consumer groups to consume logs from Log Service in real time. Then, you can send logs to the SIEM system over HTTPS or Syslog.

- For more information about how to send logs over HTTPS, see Send logs to an SIEM system over HTTPS.
- For more information about how to send logs over Syslog, see Send logs to an SIEM system over Syslog.

3.2. Send logs to an SIEM system over HTTPS

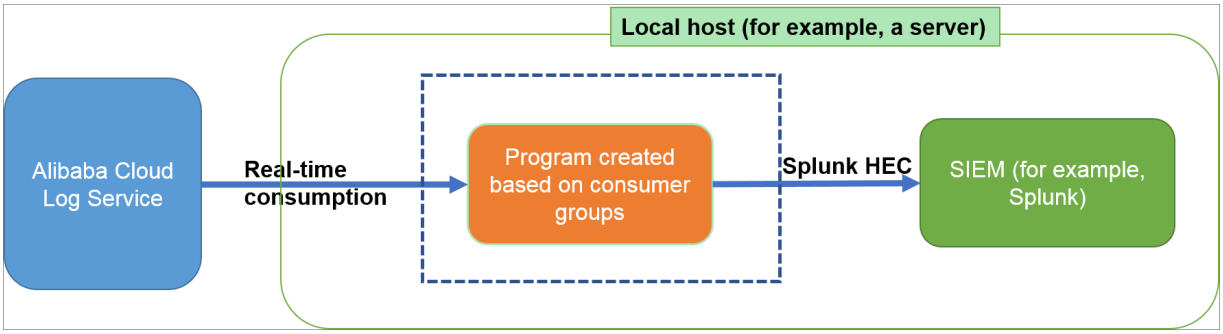
This topic describes how to send logs on Alibaba Cloud to a security information and event management (SIEM) system by using Splunk HTTP Event Collector (HEC).

Assume that the SIEM system, such as Splunk, is deployed to an on-premises environment. To ensure security, no ports are opened to allow users to access the SIEM system from an external environment.

> **Note** Code examples in this topic are used for reference only. For more information about the latest code examples, see GitHub or GitHub (applicable to the Logstore that has multiple data sources).

Workflow

We recommend that you write the required program based on consumer groups in Log Service. Then, you can call API operations provided by Splunk HEC to send logs to Splunk.



Example: Write a main program

The following code shows the control logic of a main program:

```
def main():
    option, settings = get_monitor_option()

    logger.info("*** start to consume data...")
    worker = ConsumerWorker(SyncData, option, args=(settings,) )
    worker.start(join=True)

if __name__ == '__main__':
    main()
```

Example: Configure the program

- Configure the following information:
  - Log file of the program: facilitates subsequent testing or diagnosis of potential issues.
  - Basic configuration items: includes consumer group settings and connection to Log Service.

- Advanced options for consumer groups: adjusts performance. We do not recommend that you change these settings.
- Parameters and options for the SIEM system (Splunk in this example).

- Code example:

  Read the code comments in the following example and change parameter settings based on your business needs:

```
#encoding: utf8
import os
import logging
from logging.handlers import RotatingFileHandler

root = logging.getLogger()
handler = RotatingFileHandler("{0}_{1}.log".format(os.path.basename(__file__), current_process().pid), maxBytes=10
0*1024*1024, backupCount=5)
handler.setFormatter(logging.Formatter(fmt='[%(asctime)s] - [%(threadName)s] - {%(module)s:%(funcName)s:%(lin
eno)d} %(levelname)s - %(message)s', datefmt='%Y-%m-%d %H:%M:%S'))
root.setLevel(logging.INFO)
root.addHandler(handler)
root.addHandler(logging.StreamHandler())

logger = logging.getLogger(__name__)

def get_option():
    ##########################
    # Basic configuration items
    ##########################

    # Obtain parameters and options for Log Service from environment variables.
    endpoint = os.environ.get('SLS_ENDPOINT', '')
    accessKeyId = os.environ.get('SLS_AK_ID', '')
    accessKey = os.environ.get('SLS_AK_KEY', '')
    project = os.environ.get('SLS_PROJECT', '')
    logstore = os.environ.get('SLS_LOGSTORE', '')
    consumer_group = os.environ.get('SLS_CG', '')

    # The starting point of data consumption. This parameter is valid when you run the program for the first time. Wh
en you run the program the next time, the consumption will continue from the latest consumption checkpoint.
    # You can use the BEGIN...END statement or a specific ISO time.
    cursor_start_time = "2018-12-26 0:0:0"

    ##########################
    # Advanced options
    ##########################

    # We do not recommend that you modify the consumer name, especially when concurrent consumption is required
.
```

```
    consumer_name = "{0}-{1}".format(consumer_group, current_process().pid)

    # The heartbeat interval. If the server does not receive a heartbeat report for a specific shard within twice the s
pecified interval, it indicates that the consumer is offline. In this case, the server will allocate the task to another co
nsumer.
    # We recommend that you set a greater interval when the network performance is poor.
    heartbeat_interval = 20

    # The maximum interval between two data consumption processes. If data is generated at a fast speed, you do n
ot need to adjust the setting of this parameter.
    data_fetch_interval = 1

    # Create a consumer group that contains the consumer.
    option = LogHubConfig(endpoint, accessKeyId, accessKey, project, logstore, consumer_group, consumer_name,
                          cursor_position=CursorPosition.SPECIAL_TIMER_CURSOR,
                          cursor_start_time=cursor_start_time,
                          heartbeat_interval=heartbeat_interval,
                          data_fetch_interval=data_fetch_interval)

    # Splunk options
    settings = {
            "host": "10.1.2.3",
            "port": 80,
            "token": "a023nsdu123123123",
            'https': False,          # Optional. A Boolean variable.
            'timeout': 120,          # Optional. An integer.
            'ssl_verify': True,      # Optional. A Boolean variable.
            "sourcetype": "",        # Optional. The sourcetype field is a default field defined by Splunk.
            "index": "",             # Optional. The index field is a default field defined by Splunk.
            "source": "",            # Optional. The source field is a default field defined by Splunk.
        }

    return option, settings
```

Example: Consume and send data

The following example shows how to collect data from Log Service and send the data to Splunk.

```
from aliyun.log.consumer import *
from aliyun.log.pulllog_response import PullLogResponse
from multiprocessing import current_process
import time
import json
import socket
import requests

class SyncData(ConsumerProcessorBase):
    """
```

```
    The consumer consumes data from Log Service and sends it to Splunk.
    """
    def __init__(self, splunk_setting):
      """Initiate Splunk and check network connectivity."""
        super(SyncData, self).__init__()

        assert splunk_setting, ValueError("You need to configure settings of remote target")
        assert isinstance(splunk_setting, dict), ValueError("The settings should be dict to include necessary address and confidentials.")

        self.option = splunk_setting
        self.timeout = self.option.get("timeout", 120)

        # Test connectivity to Splunk.
        s = socket.socket()
        s.settimeout(self.timeout)
        s.connect((self.option["host"], self.option['port']))

        self.r = requests.session()
        self.r.max_redirects = 1
        self.r.verify = self.option.get("ssl_verify", True)
        self.r.headers['Authorization'] = "Splunk {}".format(self.option['token'])
        self.url = "{0}://{1}:{2}/services/collector/event".format("http" if not self.option.get('https') else "https", self.option['host'], self.option['port'])

        self.default_fields = {}
        if self.option.get("sourcetype"):
            self.default_fields['sourcetype'] = self.option.get("sourcetype")
        if self.option.get("source"):
            self.default_fields['source'] = self.option.get("source")
        if self.option.get("index"):
            self.default_fields['index'] = self.option.get("index")

    def process(self, log_groups, check_point_tracker):
        logs = PullLogResponse.loggroups_to_flattern_list(log_groups, time_as_str=True, decode_bytes=True)
        logger.info("Get data from shard {0}, log count: {1}".format(self.shard_id, len(logs)))
        for log in logs:
            # Send data to Splunk.
            event = {}
            event.update(self.default_fields)
            event['time'] = log[u'__time__']
            del log['__time__']

            json_topic = {"actiontrail_audit_event": ["event"] }
            topic = log.get("__topic__", "")
            if topic in json_topic:
```

```
            try:
                for field in json_topic[topic]:
                    log[field] = json.loads(log[field])
            except Exception as ex:
                pass
        event['event'] = json.dumps(log)

        data = json.dumps(event, sort_keys=True)

            try:
                req = self.r.post(self.url, data=data, timeout=self.timeout)
                req.raise_for_status()
            except Exception as err:
                logger.debug("Failed to connect to remote Splunk server ({0}). Exception: {1}", self.url, err)

                # TODO: Add code to handle errors. For example, you can add the code to retry or send a notification in res
ponse to an error.

    logger.info("Complete send data to remote")

    self.save_checkpoint(check_point_tracker)
```

Example: Start the program

Assume that the program is named sync_data.py. The following code shows how to start the program:

```
export SLS_ENDPOINT=<Endpoint of your region>
export SLS_AK_ID=<YOUR AK ID>
export SLS_AK_KEY=<YOUR AK KEY>
export SLS_PROJECT=<SLS Project Name>
export SLS_LOGSTORE=<SLS Logstore Name>
export SLS_CG=<Consumer group name. You can set it to "syc_data".>

python3 sync_data.py
```

Example: Send data from a Logstore that has multiple sources

For a Logstore that has multiple data sources, you must set a public executor to limit the number of processes. For more information about the code example, see Send logs from a multi-source Logstore to Splunk. Note that the main function of the following example is different from the preceding one.

```
exeuctor, options, settings = get_option()

    logger.info("*** start to consume data...")
    workers = []

    for option in options:
        worker = ConsumerWorker(SyncData, option, args=(settings,) )
        workers.append(worker)
        worker.start()

    try:
        for i, worker in enumerate(workers):
            while worker.is_alive():
                worker.join(timeout=60)
            logger.info("worker project: {0} logstore: {1} exit unexpected, try to shutdown it".format(
                options[i].project, options[i].logstore))
            worker.shutdown()
    except KeyboardInterrupt:
        logger.info("*** try to exit **** ")
        for worker in workers:
            worker.shutdown()

        # wait for all workers to shutdown before shutting down executor
        for worker in workers:
            while worker.is_alive():
                worker.join(timeout=60)

    exeuctor.shutdown()
```

Limits

You can configure up to 10 consumer groups for each Logstore in Log Service. If the system displays the `ConsumerGroupQuotaExceed` error message, we recommend that you log on to the Log Service console to delete consumer groups that you no longer need.

View and monitor data consumption

You can log on to the Log Service console to view the status of a consumer group. For more information, see View consumer group status.

Concurrent consumption

You can start multiple consumer group-based programs for multiple consumers to consume data at the same time.

```
nohup python3 sync_data.py &
nohup python3 sync_data.py &
nohup python3 sync_data.py &
...
```

> **Note** The names of all consumers are unique within a consumer group because these names are suffixed with process IDs. The data of one shard can be consumed by only one consumer. If a Logstore contains 10 shards and each consumer group contains only one consumer, up to 10 consumer groups can consume the data of all shards at the same time.

Throughput

In preceding examples, Python 3 is used to run the program without limits on the bandwidth or receiving speed, such as the receiving speed on Splunk. A single consumer consumes about 20% of single-core CPU resources. In this case, the consumption speed of raw logs can reach 10 MB/s. Therefore, if 10 consumers consume data at the same time, the consumption speed of raw logs can reach 100 MB/s per CPU core. Each CPU core can consume up to 0.9 TB of raw logs every day.

High availability

A consumer group stores checkpoints on the server. When the data consumption process of one consumer stops, another consumer automatically takes over the process and continues the process from the checkpoint of the last consumption. You can start consumers on different servers. If a server stops or is damaged, a consumer on another server can take over the consumption process and continue the process from the checkpoint. To have sufficient consumers, you can start more consumers than the number of shards on different servers.

HTTPS

To use HTTPS to encrypt the data transmitted between your program and Log Service, you must set the prefix of the service endpoint to `https://`, for example, `https://cn-beijing.log.aliyuncs.com`.

The server certificate *.aliyuncs.com issued by GlobalSign. Most Linux and Windows servers are preconfigured to trust this certificate by default. If a server does not trust this certificate, you can visit the following website to download and install a valid certificate: Certificates.

3.3. Send logs to an SIEM system over Syslog

Syslog is a widely used logging standard. Almost all security information and event management (SIEM) systems, such as IBM Qradar and HP Arcsight, can receive logs over Syslog. This topic describes how to send logs on Alibaba Cloud to an SIEM system over Syslog.

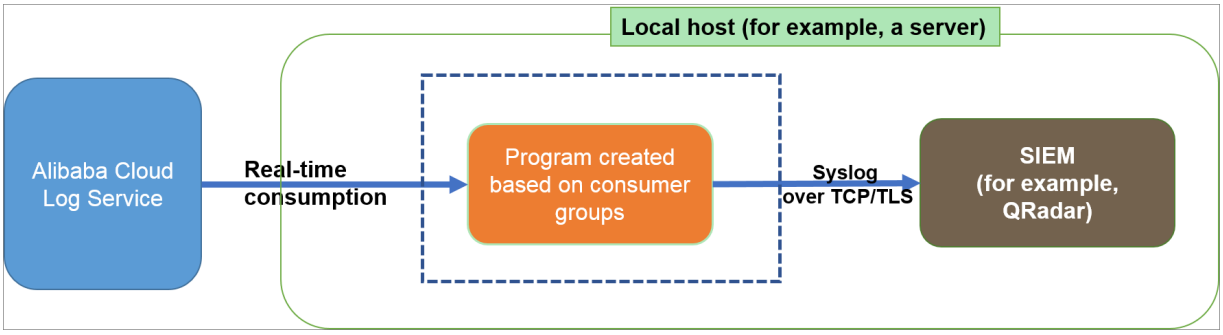
Background information

- **Syslog** is defined in RFC 5424 and RFC 3164. RFC 3164 was published in 2001, and RFC 5424 was an upgraded edition published in 2009. We recommend that you use RFC 5424 because this edition is compatible with the earlier edition and solves many issues.
- **Syslog over TCP/TLS:** Syslog defines the standard format of log messages. Both TCP and UDP support Syslog to ensure the stability of data transmission. RFC 5425 defines the use of Transport Layer Security (TLS) to provide a secure connection for the transport of Syslog messages. We recommend that you send Syslog messages over TCP or TLS if your SIEM system supports TCP or TLS.
- **Syslog facility:** the program component defined by earlier versions of Unix. You can select `user` as the default facility.
- **Syslog severity:** the severity defined for Syslog messages. You can set the log of the specified content to a higher severity level based on your business needs. The default value is `info`.

> **Note** Code examples in this topic are used for reference only. For more information about the latest code examples, see GitHub.

Workflow

We recommend that you configure the required program based on consumer groups in Log Service. Then, you can use the program to send Syslog messages over TCP or TLS to the SIEM system. We recommend that you send Syslog messages over TCP or TLS if your SIEM system supports TCP or TLS.



Example: Write a main program

The following code shows the control logic of a main program:

```
def main():
    option, settings = get_monitor_option()

    logger.info("*** start to consume data...")
    worker = ConsumerWorker(SyncData, option, args=(settings,) )
    worker.start(join=True)

if __name__ == '__main__':
    main()
```

Example: Configure the program

- Configure the following information:
  - Log file of the program: facilitates subsequent testing or diagnosis of potential issues.
  - Basic configuration items: includes consumer group settings and connection to Log Service.
  - Advanced options for consumer groups: adjusts performance. We do not recommend that you change these settings.
  - Parameters and options for the Syslog server of the SIEM system.

    > **Note** If the SIEM system supports sending Syslog messages over TCP or TLS, you must set proto to TLS and configure a valid SSL certificate.

- Code example:

  Read the code comments in the following example and change parameter settings based on your business needs:

```
#encoding: utf8
import os
import logging
from logging.handlers import RotatingFileHandler

root = logging.getLogger()
handler = RotatingFileHandler("{0}_{1}.log".format(os.path.basename(__file__), current_process().pid), maxBytes=10
0*1024*1024, backupCount=5)
```

```
handler.setFormatter(logging.Formatter(fmt='[%(asctime)s] - [%(threadName)s] - {%(module)s:%(funcName)s:%(lineno)d} %(levelname)s - %(message)s', datefmt='%Y-%m-%d %H:%M:%S'))
root.setLevel(logging.INFO)
root.addHandler(handler)
root.addHandler(logging.StreamHandler())

logger = logging.getLogger(__name__)

def get_option():
    ############################
    # Basic configuration items
    ############################

    # Obtain parameters and options for Log Service from environment variables.
    endpoint = os.environ.get('SLS_ENDPOINT', '')
    accessKeyId = os.environ.get('SLS_AK_ID', '')
    accessKey = os.environ.get('SLS_AK_KEY', '')
    project = os.environ.get('SLS_PROJECT', '')
    logstore = os.environ.get('SLS_LOGSTORE', '')
    consumer_group = os.environ.get('SLS_CG', '')

    # The starting point of data consumption. This parameter is valid when you run the program for the first time. When you run the program the next time, the consumption will continue from the latest consumption checkpoint.
    # You can use the BEGIN...END statement or a specific ISO time.
    cursor_start_time = "2018-12-26 0:0:0"

    ############################
    # Advanced options
    ############################

    # We do not recommend that you modify the consumer name, especially when concurrent consumption is required.
    consumer_name = "{0}-{1}".format(consumer_group, current_process().pid)

    # The heartbeat interval. If the server does not receive a heartbeat report for a specific shard within twice the specified interval, it indicates that the consumer is offline. In this case, the server will allocate the task to another consumer.
    # We recommend that you set a greater interval when the network performance is poor.
    heartbeat_interval = 20

    # The maximum interval between two data consumption processes. If data is generated at a fast speed, you do not need to adjust the setting of this parameter.
    data_fetch_interval = 1

    # Create a consumer group that contains the consumer.
    option = LogHubConfig(endpoint, accessKeyId, accessKey, project, logstore, consumer_group, consumer_name,
                          cursor_position=CursorPosition.SPECIAL_TIMER_CURSOR
```

```
                cursor_position=cursor_position.SPECIAL_TIMER_CURSOR,
                cursor_start_time=cursor_start_time,
                heartbeat_interval=heartbeat_interval,
                data_fetch_interval=data_fetch_interval)

    # Syslog options
    settings = {
            "host": "1.2.3.4", # Required.
            "Port": 514, # Required. The port number.
            "protocol": "tcp", # Required. Valid values: tcp, udp, and tls. The tls value is only applicable to Python 3.
            "sep": "||",      # Required. The separator that separates key-value pairs. In this example, the separator is t
wo consecutive vertical bars (||).
            "cert_path": None, # Optional. The location where the TLS certificate is stored.
            "timeout": 120,    # Optional. The timeout period. The default value is 120 seconds.
            "facility": syslogclient.FAC_USER,  # Optional. You can refer to values of the syslogclient.FAC_* parameter i
n other examples.
            "severity": syslogclient.SEV_INFO,  # Optional. You can refer to values of other syslogclient.SEV_*.
            "hostname": None,  # Optional. The hostname. The default value is the name of the local host.
            "tag": None        # Optional. The tag. The default value is a hyphen (-).
        }

    return option, settings
```

Example: Consume and send data

The following example shows how to collect data from Log Service and send the data to the Syslog server in the SIEM system. Read the code comments in the following example and configure parameters as required:

```
from syslogclient import SyslogClientRFC5424 as SyslogClient


class SyncData(ConsumerProcessorBase):
    """
    The consumer consumes data from Log Service and sends it to the Syslog server.
    """
    def __init__(self, splunk_setting):
      """Initiate the Syslog server and check its network connectivity."""
        super(SyncData, self).__init__()   # remember to call base's init

        assert target_setting, ValueError("You need to configure settings of remote target")
        assert isinstance(target_setting, dict), ValueError("The settings should be dict to include necessary address and
confidentials.")

        self.option = target_setting
        self.protocol = self.option['protocol']
        self.timeout = int(self.option.get('timeout', 120))
        self.sep = self.option.get('sep', "||")
        self.host = self.option["host"]
        self.port = int(self.option.get('port', 514))
        self.cert_path=self.option.get('cert_path', None)
```

```
        # try connection
        with SyslogClient(self.host, self.port, proto=self.protocol, timeout=self.timeout, cert_path=self.cert_path) as clien
t:
            pass

    def process(self, log_groups, check_point_tracker):
        logs = PullLogResponse.loggroups_to_flattern_list(log_groups, time_as_str=True, decode_bytes=True)
        logger.info("Get data from shard {0}, log count: {1}".format(self.shard_id, len(logs)))
        try:
            with SyslogClient(self.host, self.port, proto=self.protocol, timeout=self.timeout, cert_path=self.cert_path) as cli
ent:
                for log in logs:
                    # Put your sync code here to send to remote.
                    # the format of log is just a dict with example as below (Note, all strings are unicode):
                    #    Python2: {"__time__": "12312312", "__topic__": "topic", u"field1": u"value1", u"field2": u"value2"}
                    #    Python3: {"__time__": "12312312", "__topic__": "topic", "field1": "value1", "field2": "value2"}
                    # suppose we only care about audit log
                    timestamp = datetime.fromtimestamp(int(log[u'__time__']))
                    del log['__time__']

                    io = six.StringIO()
                    first = True
                # TODO: Modify the formatted content based on your business needs. The data is transmitted by using key-valu
e pairs that are separated with two consecutive vertical bars (||).
                    for k, v in six.iteritems(log):
                        io.write("{0}{1}={2}".format(self.sep, k, v))

                    data = io.getvalue()

                # TODO: Modify the facility and severity settings based on your business needs.
                    client.log(data, facility=self.option.get("facility", None), severity=self.option.get("severity", None), timesta
mp=timestamp, program=self.option.get("tag", None), hostname=self.option.get("hostname", None))

        except Exception as err:
            logger.debug("Failed to connect to remote syslog server ({0}). Exception: {1}".format(self.option, err))

            # TODO: Add code to handle errors. For example, you can add the code to retry or send a notification in respon
se to an error.
            raise err

        logger.info("Complete send data to remote")

        self.save_checkpoint(check_point_tracker)
```

Example: Start the program

Assume that the program is named sync_data.py. The following code shows how to start the program:

```
export SLS_ENDPOINT=<Endpoint of your region>
export SLS_AK_ID=<YOUR AK ID>
export SLS_AK_KEY=<YOUR AK KEY>
export SLS_PROJECT=<SLS Project Name>
export SLS_LOGSTORE=<SLS Logstore Name>
export SLS_CG=<Consumer group name. You can set it to "syc_data".>

python3 sync_data.py
```

Limits

You can configure up to 10 consumer groups for each Logstore in Log Service. If the system displays the `ConsumerGroupQuotaExceed` error message, we recommend that you log on to the Log Service console to delete consumer groups that you no longer need.

View and monitor data consumption

You can log on to the Log Service console to view the status of a consumer group. For more information, see View consumer group status.

Concurrent consumption

You can start multiple consumer group-based programs for multiple consumers to consume data at the same time.

```
nohup python3 sync_data.py &
nohup python3 sync_data.py &
nohup python3 sync_data.py &
...
```

> **Note** The names of all consumers are unique within a consumer group because these names are suffixed with process IDs. The data of one shard can be consumed by only one consumer. If a Logstore contains 10 shards and each consumer group contains only one consumer, up to 10 consumer groups can consume the data of all shards at the same time.

Throughput

In preceding examples, Python 3 is used to run the program without limits on the bandwidth or receiving speed, such as the receiving speed on Splunk. A single consumer consumes about 20% of single-core CPU resources. In this case, the consumption speed of raw logs can reach 10 MB/s. Therefore, if 10 consumers consume data at the same time, the consumption speed of raw logs can reach 100 MB/s per CPU core. Each CPU core can consume up to 0.9 TB of raw logs every day.

High availability

A consumer group stores checkpoints on the server. When the data consumption process of one consumer stops, another consumer automatically takes over the process and continues the process from the checkpoint of the last consumption. You can start consumers on different servers. If a server stops or is damaged, a consumer on another server can take over the consumption process and continue the process from the checkpoint. To have sufficient consumers, you can start more consumers than the number of shards on different servers.

3.4. Ship data to Splunk by using the Splunk add-on for Log Service

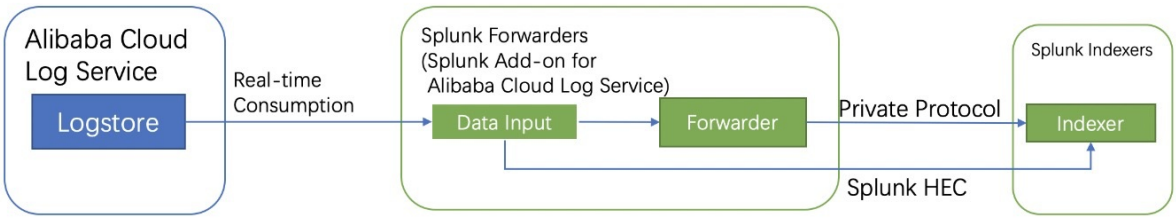
This topic describes how to use the Splunk add-on for Log Service to send log data from Log Service to Splunk.

Implementation

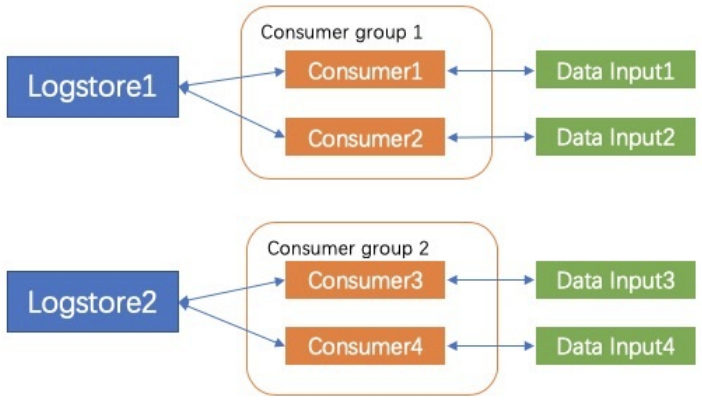
The following list describes how the Splunk add-on ships log data:

- Create consumer groups by using Splunk data inputs and use the consumer groups to consume log data from Log Service in real time.
- Splunk forwarders forward the log data to Splunk indexers by using the Splunk private protocol or HTTP Event Collector (HEC).

> **Note** The Splunk add-on is used only to collect log data. You must install the add-on on Splunk heavy forwarders. However, you do not need to install the add-on on Splunk indexers or search heads.



Mechanism



- A data input is a consumer that consumes log data.
- A consumer group consists of multiple consumers. Each consumer in a consumer group consumes different data from a Logstore.
- Each Logstore has multiple shards.
  - Each shard can be allocated to only one consumer.
  - Each consumer can consume data from multiple shards.
- The name of a consumer consists of the name of the consumer group to which the consumer belongs, the hostname, the process name, and the type of the protocol used to send Splunk events. This naming convention ensures that each consumer name in a consumer group is unique.

For more information, see Use consumer groups to consume log data.

Before you begin

- Obtain an AccessKey pair that is used to access Log Service.

  You can use the AccessKey pair of a RAM user to access a Log Service project. For more information, see AccessKey or Configure an AccessKey pair for a RAM user to access a source Logstore and a destination Logstore.

  You can use the permission assistant feature to grant permissions to a RAM user. For more information, see Use the permission assistant to grant permissions. The following example shows the common permission policy configured for a RAM user.

  > **Note** <Project name> specifies the name of the target project in Log Service. <Logstore name> specifies the name of the target Logstore. Replace the values based on your business scenarios. You can use the wildcard character (*) to specify multiple projects and Logstores.

```
{
 "Version": "1",
 "Statement": [
  {
   "Action": [
    "log:ListShards",
    "log:GetCursorOrData",
    "log:GetConsumerGroupCheckPoint",
    "log:UpdateConsumerGroup",
    "log:ConsumerGroupHeartBeat",
    "log:ConsumerGroupUpdateCheckPoint",
    "log:ListConsumerGroup",
    "log:CreateConsumerGroup"
   ],
   "Resource": [
    "acs:log:*:*:project/<Project name>/logstore/<Logstore name>",
    "acs:log:*:*:project/<Project name>/logstore/<Logstore name>/*"
   ],
   "Effect": "Allow"
  }
 ]
}
```

- Check the version of Splunk and the operating system on which Splunk runs.
  - Make sure the latest version of the add-on is used.
  - Make sure that the operating system is Linux, macOS, or Windows.
  - Make sure that the version of Splunk heavy forwarders is 8.0 or later and the version of Splunk indexers is 7.0 or later.
- Configure HEC on Splunk. For more information, see Configure HTTP Event Collector on Splunk Enterprise.

  If you use HEC to send events to Splunk indexers, make sure that HEC is configured on Splunk. If you use the Splunk private protocol to send events to Splunk indexers, skip this step.

  > **Note** You must create one or more Event Collector tokens before you can use HEC. The indexer acknowledgment feature cannot be enabled when you create an Event Collector token.

Install the Splunk add-on

You can log on to the Splunk web interface and use one of the following methods to install the add-on:

> **Note** The Splunk add-on is used only to collect log data. You must install the add-on on Splunk heavy forwarders. However, you do not need to install the add-on on Splunk indexers or search heads.

- Method 1
  i. Click the  icon.
  ii. On the **Apps** page, click **Find More Apps**.
  iii. On the Browse More Apps page, search for **Alibaba Cloud Log Service Add-on for Splunk**, and click **Install**.
  iv. After the installation is complete, restart Splunk as prompted.
- Method 2
  i. Click the  icon.
  ii. On the **Apps** page, click **Install app from file**.
  iii. On the Upload app page, select the target .tgz file from your local host, and click **Upload**.

  You can click App Search Results and download the target .tgz file on the Alibaba Cloud Log Service Add-on for Splunk page.
  iv. Click **Install**.
  v. After the installation is complete, restart Splunk as prompted.

Configure the Splunk add-on

1. On the Splunk web interface, click **Alibaba Cloud Log Service Add-on for Splunk**.
2. Configure an account.On the page that appears, click **Configuration**. On the Configuration page, click the **Account** tab. On this tab, click Add. In the Add Account dialog box, configure an AccessKey pair that you use to access Log Service.

   > **Note** You must enter an AccessKey ID in the Username field and the corresponding AccessKey secret in the Password field.

3. Configure the severity level of Splunk add-on logs.Click **Configuration**. On the Configuration page, click the **Logging** tab. On this tab, select a severity level from the Log level drop-down list.
4. Create a data input.
   i. Click **inputs**. On the **Inputs** page,
   ii. click **Create New Input**. In the Add sls_datainput dialog box, configure the parameters of the data input.

Parameters

Parameter	Required	Description	Example values
Name	Yes	The unique name of the data input. Data type: string.	N/A
Interval	Yes	The interval that the data input restarts after exit. Unit: seconds. Data type: integer.	Default value: 10.

Parameter	Required	Description	Example values
Index	Yes	The Splunk index. Data type: string.	N/A
SLS AccessKey	Yes	The Alibaba Cloud AccessKey pair that consists of an AccessKey ID and an AccessKey secret. Data type: string. Note You must enter an AccessKey ID in the Username field and the corresponding AccessKey secret in the Password field.	The AccessKey pair that you enter when you configure an account for the data input.
SLS endpoint	Yes	The endpoint of Log Service. Data type: string. For more information, see Endpoints.	■ cn-huhehaote.log.aliyuncs.com ■ https://cn-huhehaote.log.aliyuncs.com
SLS project	Yes	The name of a Log Service project. Data type: string. For more information, see Manage a project.	N/A
SLS logstore	Yes	The name of a Log Service Logstore. Data type: string. For more information, see Manage a Logstore.	N/A
SLS consumer group	Yes	The name of a consumer group. Data type: string. If you want to use multiple data inputs to consume data from the same Logstore, you must configure the same consumer group name for the data inputs. For more information, see Use consumer groups to consume log data.	N/A

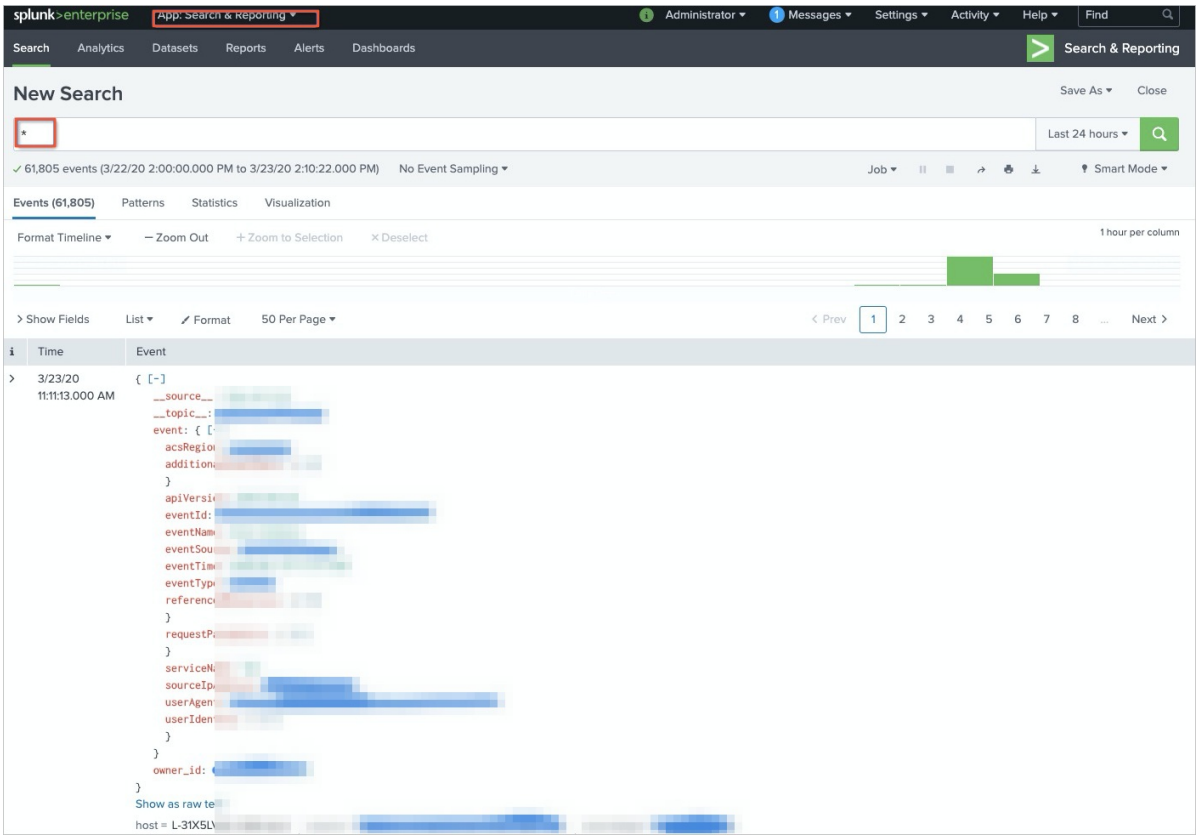
Parameter	Required	Description	Example values
SLS cursor start time	Yes	The start time of log data consumption. Data type: string. This parameter is valid only when the first consumer group is created. Then, data is consumed from the last checkpoint. Note The start time is the log receiving time.	Valid values: begin, end, and a time in the ISO 8601 format (for example, 2018-12-26 0:0:0+8:00).
SLS heartbeat interval	Yes	The heartbeat interval between the consumer and the server. Data type: integer. Unit: seconds.	Default value: 60.
SLS data fetch interval	Yes	The interval at which logs are pulled from Log Service. Data type: integer. If the log receiving rate is low, we recommend that you do not set this parameter to a small value. Unit: seconds.	Default value: 1.
Topic filter	No	Filters out log data by topic. The semicolon (;) is used to separate multiple topics. Data type: string. If a log entry is matched, it is not sent to Splunk.	TopicA;TopicB. This value indicates that log entries with the topic TopicA or TopicB are dropped.
Unfolded fields	No	Maps the fields in a log entry to the topic of the log entry in the format of {" topicA": ["field_nameA1", "field_nameA2", ...], "topicB": ["field_nameB1", "field_nameB2", ...], ...}	{"actiontrail_audit_event": ["event"] }. This value indicates that the event field is mapped to the log topic actiontrail_audit_event in the JSON format.
Event source	No	The source of Splunk events. Data type: string.	N/A
Event source type	No	The type of the Splunk event data source. Data type: string.	N/A
Event retry times	No	The number of retries to consume data. Data type: integer.	Default value: 0. This value indicates unlimited retries.

Parameter	Required	Description	Example values
Event protocol	Yes	The protocol used to send Splunk events to a Splunk indexer. If you use the Splunk private protocol to send Splunk events, you do not need to specify the following parameters in the table.	■ HTTP for HEC ■ HTTPS for HEC ■ Private protocol
HEC host	Yes	The HEC host. This parameter is valid only when you use HEC to send Splunk events. Data type: string. For more information, see Set up and use HTTP Event Collector in Splunk Web.	N/A
HEC port	Yes	The HEC port. This parameter is valid only when you use HEC to send Splunk events. Data type: integer.	N/A
HEC token	Yes	The HEC token. This parameter is valid only when you use HEC to send Splunk events. Data type: string. For more information, see HEC token.	N/A
HEC timeout	Yes	The HEC timeout period. This parameter is valid only when you use HEC to send Splunk events. Data type: integer. Unit: seconds.	Default value: 120.

Operations

- Query data

Make sure that the data input is in the Enabled state. On the Splunk web interface, click **Search & Reporting**. On the **App: Search & Reporting** page, query audit logs that are sent to Splunk.



- Query Log Service operational logs
  - Enter `index="_internal" | search "SLS info"` in the search bar to query Log Service INFO logs.
  - Enter `index="_internal" | search "error` in the search bar to query Log Service ERROR logs.

Performance and security

- Performance

  The performance of the add-on and data transmission bandwidth depend on the following factors:

  - Endpoint: You can access Log Service by using an endpoint of the public network, classic network, virtual private clouds (VPC), or the global acceleration-based public network. In most cases, we recommend that you use a classic network endpoint or a VPC endpoint. For more information, see Endpoints.
  - Bandwidth: the bandwidth of data transmission between Log Service and Splunk heavy forwarders and between Splunk heavy forwarders and indexers.
  - Processing capability of Splunk indexers: the capabilities of indexers to receive data from Splunk heavy forwarders.
  - Number of shards: A higher number of shards in a Logstore indicates a higher data transmission capability. You must decide the number of shards in a Logstore based on the receiving rate of raw logs. For more information, see Manage shards.
  - Number of Splunk data inputs: A higher number of data inputs in a consumer group that is configured for a Logstore indicates a higher throughput.

    > **Note** The number of shards in a Logstore affects the concurrent consumption of the Logstore.

  - Number of CPU cores and memory resources occupied by Splunk heavy forwarders: In most cases, one Splunk data input consumes 1 GB to 2 GB of memory resources and 1 CPU core.

If sufficient memory and CPU resources are allocated, one Splunk data input can consume log data at a rate of 1 MB to 2 MB per second.

For example, if logs are received in a Logstore at a rate of 10 MB per second, you must create at least 10 shards in the Logstore and configure 10 data inputs in the Splunk add-on. If you deploy the Splunk add-on on a single server, the server must have 10 idle CPU cores and 12 GB of available memory resources.

- High availability

  A consumer group stores checkpoints on the server. When a consumer stops consuming data, another consumer continues to consume data from the last checkpoint. You can create Splunk data inputs on multiple servers. If a server stops running or is damaged, a Splunk data input on another server continues to consume data from the last checkpoint. You can also launch more Splunk data inputs than the number of shards on multiple servers. This allows data to be consumed from the last checkpoint if an exception occurs.

- HTTPS-based data transmission
  - Log Service

    To use HTTPS to encrypt the data transmitted between your program and Log Service, you must set the prefix of the endpoint to https://, for example, https://cn-beijing.log.aliyuncs.com.

    The server certificate *.aliyuncs.com is issued by GlobalSign. By default, most Linux and Windows servers are preconfigured to trust this certificate. If the server does not trust this certificate, see Install a trusted root CA or self-signed certificate.

  - Splunk

    To use HTTPS-based HEC, you must enable the SSL feature when you enable HEC in the Global Settings dialog box. For more information, see Configure HTTP Event Collector on Splunk Enterprise.

- AccessKey pair protection

  The AccessKey pair that you use to access Log Service and HEC tokens are encrypted and stored in Splunk.

FAQ

- What can I do if a configuration error occurs?
  - Check the configurations of the data inputs. For information about configuration parameters, see Parameters.
  - Check the configurations of Log Service. Example error: failed to create a consumer group.
    - Command: `index="_internal" | search "error"`
    - Exception logs:

      ```
      aliyun.log.consumer.exceptions.ClientWorkerException:
      error occour when create consumer group,
      errorCode: LogStoreNotExist,
      errorMessage: logstore xxxx does not exist
      ```

    - Check whether the number of consumer groups configured for a Logstore exceeds the quota.

      You can configure a maximum of 20 consumer groups for a Logstore. We recommend that you delete unnecessary consumer groups. If more than 20 consumer groups are configured for a Logstore, the ConsumerGroupQuotaExceed error is returned.

- What do I do if a permission error occurs?

- Check whether you are authorized to access Log Service.
  - Command: `index="_internal" | search "error"`
  - Exception logs:

    ```
    aliyun.log.consumer.exceptions.ClientWorkerException:
    error occour when create consumer group,
    errorCode: SignatureNotMatch,
    errorMessage: signature J70VwxYH0+W/AciA4BdkuWxK6W8= not match
    ```

- Check whether you are authorized to access HEC.
  - Command: `index="_internal" | search "error"`
  - Exception logs:

    ```
    ERROR HttpInputDataHandler - Failed processing http input, token name=n/a, channel=n/a, source_IP=127.0.0.1
    , reply=4, events_processed=0, http_input_body_size=369

    WARNING pid=48412 tid=ThreadPoolExecutor-0_1 file=base_modinput.py:log_warning:302 |
    SLS info: Failed to write [{"event": "{\"__topic__\": \"topic_test0\", \"__source__\": \"127.0.0.1\", \"__tag__:__clie
    nt_ip__\": \"10.10.10.10\", \"__tag__:__receive_time__\": \"1584945639\", \"content\": \"goroutine id [0, 15849456
    37]\", \"content2\": \"num[9], time[2020-03-23 14:40:37|1584945637]\"}", "index": "main", "source": "sls log", "sou
    rcetype": "http of hec", "time": "1584945637"}] remote Splunk server (http://127.0.0.1:8088/services/collector) u
    sing hec.
    Exception: 403 Client Error: Forbidden for url: http://127.0.0.1:8088/services/collector, times: 3
    ```

  - Possible causes
    - HEC is not configured or started.
    - The HEC-relevant parameters of data inputs are invalid. For example, if you use HTTPS-based HEC, you must enable the SSL feature.
    - The indexer acknowledgment feature is disabled.

- What do I do if a consumption delay occurs?

  You can view the status of consumer groups in the Log Service console. For more information, see View consumer group status.

  Increase the number of shards in the Logstore or create more data inputs in the same consumer group. For more information, see Performance and security.

- What do I do if network jitters occur?
  - Command: `index="_internal" | search "SLS info: Failed to write"`
  - Exception logs:

    ```
    WARNING pid=58837 tid=ThreadPoolExecutor-0_0 file=base_modinput.py:log_warning:302 |
    SLS info: Failed to write [{"event": "{\"__topic__\": \"topic_test0\", \"__source__\": \"127.0.0.1\", \"__tag__:__client_
    ip__\": \"10.10.10.10\", \"__tag__:__receive_time__\": \"1584951417\", \"content2\": \"num[999], time[2020-03-23 16:
    16:57|1584951417]\", \"content\": \"goroutine id [0, 1584951315]\"}", "index": "main", "source": "sls log", "sourcetyp
    e": "http of hec", "time": "1584951417"}] remote Splunk server (http://127.0.0.1:8088/services/collector) using hec.

    Exception: ('Connection aborted.', ConnectionResetError(54, 'Connection reset by peer')), times: 3
    ```

  Splunk events are automatically retransmitted if network jitters occur. If the problem persists, contact your network administrator for troubleshooting.

- Modify the start time of data consumption

  > Note The SLS cursor start time parameter is valid only when you create a consumer group for the first time. From the next time, data is consumed from the last checkpoint.

  i. On the **Input** page of the Splunk Web UI, disable the target data input.

  ii. Log on to the Log Service console. Find the Logstore from which data is consumed, and delete the consumer group under **Data Consumption**.

  iii. On the **Input** page of the Splunk Web UI, find the target data input, and choose **Actions** > **Edit**. In the dialog box that appears, modify the SLS cursor start time parameter. Restart the data input.

4.Ship log data from Log Service to TSDB

This topic describes how to ship log data from Log Service to TSDB. The timestamp of a log entry in Log Service is mapped to a timestamp in a data entry in TSBD. This facilitates time-series data storage and satisfy different business requirements.

Context

Logs are essential data that you can use to process historical data, troubleshoot errors, and monitor system activities. In addition, logs are necesssary data sources for data analysts, developers, and O&M personnel.

Log Service allows you to collect log data from multiple cloud services and transform the data based on your business requirements. You can also ship log data from Log Service to TSDB. Timestamps in Log Service are mapped to timestamps in TSDB. This way, log data that is shipped to TSDB is stored as time-series data.

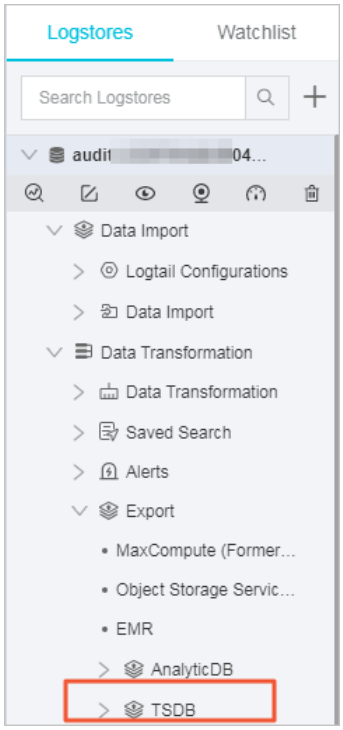
Prerequisites

- Log Service and TSDB are activated.
- The source Log Service project and the destination TSDB instance reside in the same region.

Create a log shipping task

The following procedure describes how to create a log shipping task.

1. Log on to the Log Service console. On the page that appears, click the source project. On the page that appears, click the Logstores tab.
2. On the **Logstores** tab, click the > icon of the source Logstore, and then choose **Data Transformation** > **Export** > **TSDB**.



3. Click the plus sign (+) next to TSDB. You are redirected to the task configuration page.

> **Note** If the data in the source Logstore is not transformed, you are prompted to transform the data before shipping. If the data meets the format requirements, you can skip the data transformation and ship the data to TSDB.

4. Complete parameter configurations.

The following table describes the parameters in the preceding figure.

Parameter	Description
Shipping Name	The name of the log shipping task.
Shipping Description	The description of the log shipping task. We recommend that you enter an informative description for easy management.
TSDB Instance	The destination TSDB instance.
Metric Name	The metric to which the shipped log data belongs. If no metric is available in a TSDB instance, a metric is automatically created when you ship log data to the instance.
AccessKeyID	The AccessKey ID of the current Alibaba Cloud account that you use to access the destination TSDB instance.
AccessKeySecret	The AccessKey secret of the current Alibaba Cloud account that you use to access the destination TSDB instance.
Time Point Mapping	Maps the timestamp in a log entry to the timestamp of a metric value. By default, the log generation time is used as the timestamp of a metric value in TSDB. The default value of this parameter is in the ${logTag:time} format. You can also specify a field in a log entry and map the value of the field to the timestamp of a metric value. In this case, the value of this parameter is in the ${logColumn:xx} format. If the value of the specified field fails to replace the log generation time, the log generation time is used as the timestamp. If you specify a field, you must specify a field whose value can be converted to a UNIX timestamp in seconds or milliseconds. The timestamp of a log entry in Log Service is accurate to the second. You can specify a field if multiple log entries are generated in the same second. You can also use the following expression to specify a contextual field of a log entry: `${context:variable}` For more information about the contextual information of a log entry, see Context query. logColumn in the ${logColumn:xx} expression indicates that the value of the specified field in a log entry is mapped to a timestamp in TSDB. logTag in the ${logTag:xx} expression indicates that the value of the specified tag field in a log entry is mapped to a timestamp in TSDB. Common tag fields: ◦ __topic__: the topic of a log group that consists of multiple log entries. If you specify this tag field, the expression is ${logTag:__topic__}. ◦ __source__: the IP address of the server that generates the log entry. If you specify this tag field, the expression is ${logTag:__source__}. ◦ __time__: the timestamp of the log entry, accurate to the second. If you specify this tag field, the expression is ${logTag:__time__}.
Field Mapping	Maps log fields to fields in TSDB. If only one field can be mapped, log data is shipped to TSDB in the single-value model. If multiple fields can be mapped, log data is shipped to TSDB in the multi-value model.
Tag Mapping	Maps log fields to tags of a metric. Enter the tag name in the first text box and the tag value in the second text box. You can use the following expression to specify a contextual field: `${context:variable}` For more information about the contextual information of a log entry, see Context query.

Parameter	Description
Shipping Time	The log generation time of the first log entry to be shipped.

5. Click **Submit**.

Related operations

- After a log shipping task is started, you can click the task name in the console and stop the task or view the status of the task on the task management page.
- You can also modify the configurations of the task. For information about configuration parameters, see Create a log shipping task.

5.Best practices

5.1. Connect to a data warehouse

The LogShipper feature of Log Service ships logs to storage services such as Object Storage Service (OSS), Table Store, and MaxCompute. It cooperates with E-MapReduce (Spark and Hive) and MaxCompute for offline computing.

Data warehousing (offline computing)

Data warehousing (offline computing) is the supplement to real-time computing, but they are used for different purposes.

Mode	Advantage	Disadvantage	Application scope
Real-time computing	Fast	Simple computing	Mainly used for incremental computation in monitoring and real-time analysis
Offline computing (data warehousing)	Accurate and powerful	Relatively slow	Mainly used for full computation in Business Intelligence (BI), and data statistics and comparison

To satisfy the current data analysis requirements, you need to perform both real-time computing and data warehousing (offline computing) on the same set of data. For example, you need to perform the following operations when processing access logs:

- Use Realtime Compute to display the data, including the current PV, UV, and operator information, on the dashboard in real time.
- Conduct detailed analysis of the full data every night to obtain the growth, year-on-year or month-on-month growth, and top ranking data.

In the world of Internet, there are two classic data processing architectures:

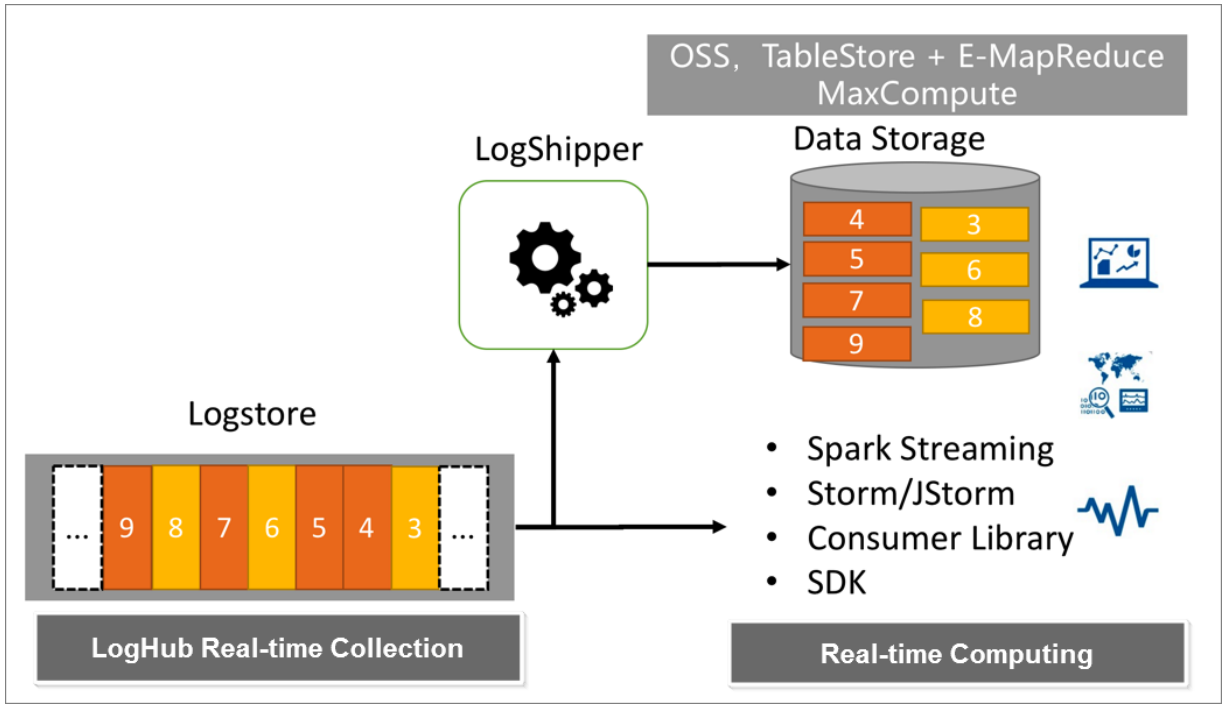
- Lamdba architecture: When data comes in, the architecture can stream and at the same time save the data to the data warehouse. However, when you initiate a query, the results are returned from real-time computing and offline computing based on query conditions and complexities.
- Kappa architecture: Kafka-based architecture. The offline computing feature is weakened. All data is stored in Kafka, and all queries are fulfilled with real-time computing.

Log Service provides an architecture that is more similar to the Lamdba architecture.

LogHub and LogShipper for both real-time and offline computing

Create a Logstore first, and configure LogShipper in the console to enable data warehouse connection. Currently, the following data warehouses are supported:

- OSS: large-scale object storage service
- TableStore: NoSQL data storage service
- MaxCompute: data computing service



LogShipper provides the following features:

- Quasi real-time: connects to a data warehouse in minutes.
- Enormous data volume: does not need to worry about concurrency.
- Retry on error: performs automatic retries or API-based manual retries in case of faults.
- Task API: uses APIs to acquire the log shipping status for different time periods.
- Automatic compression: supports data compression to reduce the storage bandwidth.

Typical scenarios

Scenario 1: Log auditing

A is responsible for maintaining a forum and part of his job is to conduct audits and offline analysis of all access logs on the forum.

- Department G wants A to capture user visits over the past 180 days, and to provide the access logs within a given period of time on demand.
- The operations team must prepare an access log report on a quarterly basis.

A uses Log Service to collect log data from the servers and enables the LogShipper feature, allowing Log Service to automatically collect, ship, and compress logs. When an audit is required, the logs within the time period can be authorized to a third party. To conduct offline analysis, use E-MapReduce to run a 30-minute offline task. In this way, two jobs are done at minimal cost. In addition, Data Lake Analytics (DLA) can be used to analyze the log data shipped to OSS.

Scenario 2: Real-time and offline analysis of log data

As an open-source software enthusiast, B prefers to use Spark for data analysis. His requirements are as follows:

- Collect logs from the mobile client by using APIs.
- Use Spark Streaming to conduct real-time log analysis and collect statistics on online user visits.
- Use Hive to conduct offline analysis in T+1 mode.
- Grant downstream agencies access to the log data for analysis in other dimensions.

With a combination of Log Service, OSS, E-MapReduce or DLA, and Resource Access Management (RAM), you can fulfill such requirements.