

Alibaba Cloud

AnalyticDB for MySQL
Data integration

Document Version: 20201218

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 <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1.Synchronize data	05
2.Import Data	06
2.1. Big data	06
2.1.1. Use DataWorks to import data from EMR	06
2.1.1.1. Configure an HDFS connection	06
2.1.1.2. Configure a MaxCompute connection	07
2.1.1.3. Configure an AnalyticDB for MySQL connection	07
2.1.1.4. Configure the source connection and destination con..	07
2.1.2. Import MaxCompute data to AnalyticDB for MySQL	07
2.1.3. Flink ADB Connector	10
2.2. Message queues	11
2.2.1. Use Logstash to write Apache Kafka data to AnalyticD...	11
2.2.1.1. Overview	11
2.2.1.2. Use Logstash to write Apache Kafka data to Analyt...	12
2.3. Log data	14
2.3.1. Use Logstash to collect log data in real time	14
2.3.2. Use Log Service to transfer log data to AnalyticDB for...	16
3.Submit import or export tasks asynchronously	19
4.Export Data	21
4.1. Export data from AnalyticDB for MySQL to MySQL	21
4.2. Export data from AnalyticDB for MySQL to OSS	23
4.3. Use external tables to export data to MaxCompute partitio...	25

1.Synchronize data

To synchronize data in different scenarios, you can use the following data synchronization solutions provided by AnalyticDB for MySQL:

Data source	Reference
Databases	<ul style="list-style-type: none">• Use DTS to synchronize data from ApsaraDB RDS MySQL• Use external tables to synchronize data. For more information, see Export data from AnalyticDB for MySQL to MySQL and Import MySQL data to AnalyticDB for MySQL.• Use DataWorks to synchronize data from ApsaraDB RDS for MySQL
OSS	<ul style="list-style-type: none">• Use external tables to synchronize data. For more information, see Import data from OSS to AnalyticDB for MySQL and Export data from AnalyticDB for MySQL to OSS.• Use DataWorks to synchronize data from OSS
Big data	<ul style="list-style-type: none">• Use DataWorks to synchronize data from Hadoop• Use DataWorks to synchronize data from EMR
Local data	<ul style="list-style-type: none">• Use the LOAD DATA statement to import local data• Use Kettle to synchronize local data to AnalyticDB for MySQL

2. Inport Data

2.1. Big data

2.1.1. Use DataWorks to import data from EMR

2.1.1.1. Configure an HDFS connection

This topic describes how to configure an HDFS connection in DataWorks.

Procedure

1. Log on to the [DataWorks console](#). In the Workspaces section, find the target workspace and click **Data Integration**.
2. On the **Data Integration** page, click **Connection** in the left-side navigation pane. On the page that appears, click **Add Connection**.
3. In the **Add Connection** dialog box, click **HDFS**.
4. In the **Add HDFS Connection** dialog box, configure the parameters as prompted.

Add HDFS Connection
✕

* Connection Name :

Description :

* Environment : Development Production

* DefaultFS : ?

Extension Parameters : ?

Resource Group Name	Type	Status	Test Time	Actions
Public Resource Group		Not Tested		Test Connection..

Attention : If the test fails, the possible reason is:

1. The database is not started, please confirm that it has started normally.
2. DataWorks cannot access the network where the database is located. Please make sure the network is connected with Aliyun.
3. DataWorks is prohibited by the network firewall where the database is located. Please add [whitelist](#) .
4. The database domain name cannot be resolved correctly. Please confirm that the domain name can

Previous Complete

Parameter	Description
Connection Name	The name of the connection. The name can contain letters, digits, and underscores (_). It cannot start with a digit or underscore (_).

Parameter	Description
Description	The description of the connection. The description can be up to 80 characters in length.
DefaultFS	The address of the NameNode of the HDFS, in the <code>hdfs://ServerIP:Port</code> format.

5. Click **Test Connection**.
6. After the connection passes the connectivity test, click **Complete**.

Description of connectivity test

- If the user-created data source is hosted on an ECS instance in a classic network, network connectivity is not guaranteed when nodes are deployed in the default resource group. We recommend that you use custom resource groups.
- Connectivity testing is not supported for data sources in VPCs. You can click **Complete** without testing the connectivity.

2.1.1.2. Configure a MaxCompute connection

This topic describes how to configure a MaxCompute connection.

For more information, see [Configure a MaxCompute connection](#).

2.1.1.3. Configure an AnalyticDB for MySQL connection

To configure an AnalyticDB for MySQL connection, you can follow the steps for using DataWorks.

For more information, see [Configure an AnalyticDB for MySQL connection](#).

2.1.1.4. Configure the source connection and destination connection for a synchronization task

To synchronize data from EMR to AnalyticDB for MySQL, you can follow the steps for configuring synchronization tasks.

For more information, see [Use DataWorks to synchronize data from ApsaraDB RDS for MySQL](#).

2.1.2. Import MaxCompute data to AnalyticDB for MySQL

This topic describes how to import MaxCompute data to AnalyticDB for MySQL.

Prerequisites

- Source data is prepared. For more information, see [Prepare](#) and [Quick Start](#).

The following table is created in MaxCompute through the table creation statement. For more information, see [Create a table](#). If you have created source data, skip this step.

```
CREATE TABLE IF NOT EXISTS odps_nopart_import_test
(
  uid STRING,
  other STRING
)
LIFECYCLE 3;
```

The data within a file named `odps_nopart_import_test.txt` is imported to a table named `odps_nopart_import_test`. For more information, see [Create tables and import data](#).

- The preparations for using AnalyticDB for MySQL are completed. For example, a cluster is created and a whitelist is configured. An account and a database are created. For more information, see [Quick start of AnalyticDB for MySQL](#).

Procedure


1. Connect to an AnalyticDB for MySQL cluster and the target database. For more information, see [Connect to a cluster](#).

The data in the `odps_nopart_import_test` table is imported to the `test_adb` database in AnalyticDB for MySQL in this example.

2. Execute the CREATE TABLE statement to create an external mapping table named `odps_nopart_import_test_external_table` in the `test_adb` database. For more information, see [CREATE TABLE](#).

```
CREATE TABLE IF NOT EXISTS odps_nopart_import_test_external_table
(
  uid string,
  other string
)
ENGINE='ODPS'
TABLE_PROPERTIES='{
  "endpoint":"http://service.cn.maxcompute.aliyun-inc.com/api",
  "accessid":"L*****FsE",
  "accesskey":"CcwF*****iWjv",
  "project_name":"odps_project1",
  "table_name":"odps_nopart_import_test"
}'
```

Parameter	Description
ENGINE='ODPS'	Specifies the table as an external table whose data engine is MaxCompute.
TABLE_PROPERTIES	Specifies the connection information used by the AnalyticDB for MySQL database to access the data in MaxCompute.

Parameter	Description
endpoint	<p>The endpoint of MaxCompute.</p> <p> Note AnalyticDB for MySQL can access MaxCompute only through the VPC endpoint.</p> <p>For more information, see Configure endpoints.</p>
accessid	<p>The AccessKey ID used to access the source table in MaxCompute.</p> <p>For more information about how to obtain your <code>accessid</code> and <code>accesskey</code>, see Obtain the AccessKey pair of an account.</p>
accesskey	The AccessKey secret used to access the source table in MaxCompute.
project_name	The name of the project in MaxCompute.
table_name	The name of the source table in MaxCompute.

- Execute the CREATE TABLE statement to create a destination table named `adb_nopart_import_test` in the `adb_demo` database to store data imported from MaxCompute. For more information, see [CREATE TABLE](#).

```
CREATE TABLE IF NOT EXISTS adb_nopart_import_test
(
  uid string,
  other string
)
DISTRIBUTE BY HASH(uid);
```

- Execute the INSERT statement to import data from MaxCompute to AnalyticDB for MySQL.
 - Execute the `INSERT INTO` statement to import data.

```
insert into adb_nopart_import_test
select * from odps_nopart_import_test_external_table
```

```
select * from adb_nopart_import_test
```

```
+-----+-----+
| uid   | other |
+-----+-----+
| 4     | other4 |
| 6     | other6 |
| 5     | other5 |
| 2     | other2 |
| 1     | other1 |
| 3     | other3 |
| 7     | other7 |
```

- Execute the `INSERT OVERWRITE INTO` statement to import data asynchronously.

```
submit job insert overwrite into adb_nopart_import_test
```

```
select * from odps_nopart_import_test_external_table
```

```
+-----+
| job_id          |
+-----+
| 2020112122202917203100908203303000321 |
```

For more information about asynchronous execution, see [Submit import or export tasks asynchronously](#).

2.1.3. Flink ADB Connector

This topic describes how to write the data of Apache Flink to AnalyticDB for MySQL.

Procedure

- Download [flink-jdbc-adb-compatibility_2.11-1.10.10.jar](#).
- Go to the `/home/admin/flink-1.10.1/lib` directory of each Apache Flink node and replace the `flink-jdbc_2.11-1.10.1.jar` file with `flink-jdbc-adb-compatibility_2.11-1.10.10.jar`.
- Restart all the Apache Flink nodes.
- Create an AnalyticDB for MySQL table in Apache Flink.

```
CREATE TABLE table_test (  
  col1 VARCHAR,  
  col2 BIGINT,  
  col3 BIGINT  
) WITH (  
  'connector.type' = 'jdbc', -- Use a JDBC connector.  
  'connector.dialect' = 'adb ', -- Use the connector dialect.  
  'connector.url' = 'jdbc:mysql://ip:port/db_name', -- jdbc url  
  'connector.table' = 'table_name', -- The name of the table.  
  'connector.username' = 'xxx', -- The user name of the account that owns the destination AnalyticDB for  
  MySQL cluster.  
  'Connector.password' = 'xxx', -- The password of the account that owns the destination AnalyticDB for  
  MySQL cluster.  
);
```

For more information about Apache Flink, see [Flink](#).

2.2. Message queues

2.2.1. Use Logstash to write Apache Kafka data to AnalyticDB for MySQL

2.2.1.1. Overview

Logstash is an open source, server-side data processing pipeline. It can collect data from multiple data sources at the same time, convert the data, and then write the data to specified storage. AnalyticDB for MySQL is fully compatible with MySQL. You can write data from data sources supported by Logstash input plug-ins into AnalyticDB for MySQL. This topic describes how to use the logstash-input-kafka plug-in to write Apache Kafka data into AnalyticDB for MySQL.

Logstash plug-ins

- **Input plug-ins that are used to collect data of various types and in different sizes from disparate sources**

In common business scenarios, data is stored in a variety of formats across multiple systems in a centralized or scattered manner. Logstash supports multiple data input modes to collect data from disparate data sources at the same time. Logstash can collect data from logs, metrics, web applications, data storage, and Amazon services in a continuous streaming manner.

- **Filter plug-ins that are used to parse and convert data in real time**

Logstash uses filters to parse all types of events and identify defined fields to construct schemas. Then, it converts the schemas into common data types and transmits data to destination repositories, enabling you to analyze and make the most out of data in an easy and efficient manner.

- Use Grok to parse unstructured data into structured data.
- Parse geographic information from IP addresses.

- Anonymize personally identifiable information (PII) to completely exclude sensitive fields.
- Simplify overall processing without being affected by data sources, formats, or architectures.
- **Output plug-ins that are used to export data**

In addition to AnalyticDB for MySQL, Logstash provides several data output destinations.

Example

Apache Kafka is a distributed service that can publish and subscribe to logs with high throughput. It provides high availability, high performance, distributed architecture, high scalability, and durability. Apache Kafka is widely used in major companies. It can be integrated with Logstash, eliminating the need for repeated construction.

For more information, see [Use Logstash to write Apache Kafka data to AnalyticDB for MySQL](#).

2.2.1.2. Use Logstash to write Apache Kafka data to AnalyticDB for MySQL

This topic describes how to use Logstash to write Apache Kafka data to AnalyticDB for MySQL.

Procedure

1. Run the following commands to install and update plug-ins:

```
$bin/plugin install
$bin/plugin update
```

Logstash 1.5 and later versions have been integrated with Apache Kafka and have changed the directories and names of all plug-ins. The URL for publishing plug-ins is [logstash-plugins](#).

2. Configure plug-ins.
 - **Input configuration examples**

The following example shows how to use a Kafka consumer.

```
input {
  kafka {
    zk_connect => "localhost:2181"
    group_id => "logstash"
    topic_id => "test"
    codec => plain
    reset_beginning => false # boolean (optional), default: false
    consumer_threads => 5 # number (optional), default: 1
    decorate_events => true # boolean (optional), default: false
  }
}
```


Parameters:

- `group_id` : specifies the consumer group with a unique group ID. Consumption from different consumer groups is isolated from each other.
- `topic_id` : specifies the topic. A consumer first subscribes to a topic and then consumes data in the topic.
- `reset_beginning` : specifies the position from which Logstash starts to consume data. By default, Logstash starts to consume data from the previous offset. If no data has been consumed, Logstash will start to consume data from the start offset.

To import the original data, you must set `reset_beginning` to `true`, which works in the same way as the `cat` command. This way, Logstash will consume data from the start offset. When Logstash consumes the last row, the system sets `reset_beginning` to `tail -F` and Logstash will continue to monitor the corresponding data without terminating.

- `decorate_events` : specifies that the intrinsic information is provided when a message is published. This information includes the size of the consumed message, topic source, and consumer group information.
- `rebalance_max_retries` : specifies the number of retries that can be performed to register the owner registry node of a partition. A rebalance is performed when a new consumer joins a consumer group, after which some partitions are moved to the new consumer. If the new consumer obtains consumption permissions on a specific partition, the new consumer will register the partition owner registry node in ZooKeeper. Retries are performed until the original consumer releases the node.
- `consumer_timeout_ms` : specifies that an exception is thrown if no message arrives within the specified timeout period. Typically, you do not need to modify this parameter.

For more information about Input parameters, visit [logstash-kafka](#).

 **Note** If multiple consumers need to consume messages of the same topic in parallel, divide the topic into multiple partitions and set the same `group_id` and `topic_id` values for two or more consumers. This ensures that messages are consumed in sequence.

◦ Output configuration examples

```
output {
  jdbc {
    driver_class => "com.mysql.jdbc.Driver"
    connection_string => "jdbc:mysql://HOSTNAME/DATABASE? user=USER&password=PASSWORD"
    statement => ["INSERT INTO log (host, timestamp, message) VALUES(?, ?, ?)", "host", "@timestamp", "message"]
  }
}
```

Parameters:

- `connection_string` : specifies the connection string of AnalyticDB for MySQL.
- `statement` : specifies the declared arrays in the INSERT statement.

For more information about Output parameters, visit [logstash-kafka](#).

3. Run the `bin/logstash -f config/xxx.conf` command in the installation directory of Logstash to start the task and write Kafka data to AnalyticDB for MySQL.

2.3. Log data

2.3.1. Use Logstash to collect log data in real time

Logstash is an open source data collection engine with real-time pipelining capabilities. It is first used to write log data into Elastic Stack. As the open source community develops, Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice.

For example, Java Database Connectivity (JDBC) is allowed to access AnalyticDB for MySQL. You can use a Logstash plug-in named `logstash-output-jdbc` to import log data to AnalyticDB for MySQL for subsequent analysis. However, JDBC writes single records. If you use JDBC to write large amounts of log data to AnalyticDB for MySQL, the write performance is low and the CPU utilization is high. To manage this issue, AnalyticDB for MySQL has optimized a Logstash plug-in named `logstash-output-analyticdb` based on JDBC. This plug-in is dedicated to write log data to AnalyticDB for MySQL in aggregate.

The `logstash-output-analyticdb` plug-in provides five times the write speed of the `logstash-output-jdbc` plug-in at a lower CPU utilization.

Install Logstash

The following section describes how to install the `logstash-output-analyticdb` plug-in. For information about how to install Logstash, visit [Installing Logstash](#).

1. Run the following command to go to the root directory of Logstash: `cd logstash` .
2. Run the following command to install the `logstash-output-analyticdb` plug-in: `bin/logstash-plugin install logstash-output-analyticdb` .
3. Run the following command to create a directory named `vendor/jar/jdbc` under the root directory of Logstash: `mkdir -p vendor/jar/jdbc` .
4. Run the following command to copy the installation package of JDBC to `vendor/jar/jdbc` : `cd vendor/jar/jdbc; wget http://central.maven.org/maven2/mysql/mysql-connector-java/5.1.36/mysql-connector-java-5.1.36.jar` .

After you complete the preceding steps, the `logstash-output-analyticdb` plug-in is installed.

Configure Logstash

Create a configuration file named `logstash-analyticdb.conf` under the config directory. You can also customize the file name. `logstash-analyticdb.conf` contains the following content:

```

input
{
  stdin {}
}
output {
  analyticdb {
    driver_class => "com.mysql.jdbc.Driver"
    connection_string => "jdbc:mysql://HOSTNAME:PORT/DATABASE? user=USER&password=PASSWORD"
    statement => [ "INSERT INTO log (host, timestamp, message) VALUES(?, ?, ?)", "host", "@timestamp", "message" ]
    commit_size => 4194304
  }
}

```

- `connection_string` : the JDBC connection string used to connect to AnalyticDB for MySQL.
- `statement` : the declared arrays in the INSERT statement.

Other parameters:

- `max_flush_exceptions` : specifies the maximum number of retries that can be attempted if an exception occurs during data write. Default value: 100.
- `skip_exception` : specifies whether to skip exceptions. The default value is FALSE, indicating that the maximum number of retries specified by the `max_flush_exceptions` parameter are attempted if an exception occurs. If all of these retries fail, an exception will be thrown to terminate the synchronization task. If this parameter is set to TRUE and the maximum number of retries fail, the exception is skipped and simply written to a log.
- `flush_size` : the maximum number of data records that can buffer simultaneously. This parameter is used in combination with the `commit_size` parameter.
- `commit_size` : the maximum amount of data that can buffer simultaneously. This parameter is used in combination with the `flush_size` parameter. Data write tasks are submitted when upper limits are reached.

The preceding configuration file is provided for reference. You must configure the `logstash-analyticdb.conf` file. For more information about configurations related to AnalyticDB for MySQL, visit [README](#). For more information about configurations and rules of Logstash, see the Logstash documentation.

After you configure the preceding parameters, the configuration is complete.

Start a task

Run the following command in the installation directory of Logstash to start a task: `bin/logstash -f config/logstash-analyticdb.conf` .

Precautions

Before you write data into AnalyticDB for MySQL, we recommend that you run the following command to upgrade Logstash to the latest version:

```
bin/logstash-plugin update logstash-output-analyticdb
```

2.3.2. Use Log Service to transfer log data to AnalyticDB for MySQL

This topic describes how to use Log Service to transfer log data to AnalyticDB for MySQL.

Context


Logs are a special type of data that plays an important role in processing historical data, diagnosing problems, and tracing system activities. Logs are essential data sources for data analysts, developers, and O&M personnel.

Alibaba Cloud allows you to use [Log Service](#) to synchronize logs such as NGINX access logs, Log4j logs, Apache access logs, and structured text to AnalyticDB for MySQL in real time. This allows you to use AnalyticDB for MySQL to perform real-time log analytics.

Prerequisites

Before you use Log Service, complete the following steps:

1. When you use Log Service for the first time, you must use your Alibaba Cloud account to log on to the [Log Service](#) product page and click **Get it Free**. The system automatically redirects you to the buy page. Read and select the I agree with Log Service Agreement of Service check box and click **Enable Now**. The [Log Service console](#) appears.

 **Note** If you have activated Log Service, you can start to create a project.

2. Create a project and a Logstore. For more information, see [Get started](#).

In this example, a project named adb-test and a Logstore named adb-source are created.

Before you use AnalyticDB for MySQL, complete the following steps:

1. [Create an AnalyticDB for MySQL cluster](#).
2. [Create database accounts](#).
3. [Create databases](#).
4. If you need to use a public endpoint to connect to AnalyticDB for MySQL clusters, you must first [apply for a public endpoint](#).
5. Execute the **CREATE TABLE** statement to create tables to store the log data transferred to AnalyticDB for MySQL.

Create a LogShipper task


You can use the Export function in the Data Transformation module of Log Service to transfer the collected logs in Logstores to AnalyticDB for MySQL.

1. Log on to the [Log Service console](#) and click the target project.

- On the **Logstores** tab, find the target Logstore and choose > > **Data Transformation** > **Export** > **AnalyticDB**.

If you use Log Service to transfer log data to AnalyticDB for MySQL for the first time, you must grant access permissions on Log Service to AnalyticDB for MySQL.

- Click **+** to the right of AnalyticDB. The system prompts you to complete authorization. Click **permissions**.
 - On the Cloud Resource Access Authorization page, click **Confirm Authorization Policy** and assign the RAM role **AliyunAnalyticDBAccessingLogRole** to AnalyticDB for MySQL.
- After you complete authorization, click **+** to the right of AnalyticDB. You can click **Ship** to directly transfer log data. You can also click **Go to Data Transformation** to **process** log data before the data is transferred.

 **Note** In this example, the log data is transferred directly.

- On the **LogHub - Data Shipper** page, configure the parameters as prompted.

LogHub - Data Shipper parameters

Parameter	Description
Shipper Name	The name of the LogShipper task. We recommend that you specify an identifiable name to facilitate subsequent management.
Shipper Description	The description of the LogShipper task. We recommend that you specify a meaningful description to facilitate subsequent management.
Cluster Version	The version of the AnalyticDB for MySQL cluster. Select 3.0 . For more information about how to use Log Service to transfer log data to AnalyticDB for MySQL V2.0, see Use Log Service to synchronize ECS log data to AnalyticDB for MySQL .
Cluster Name	The name of the destination AnalyticDB for MySQL cluster.
Database Name	The name of the database in the destination AnalyticDB for MySQL cluster.
Table Name	The name of the table in the destination AnalyticDB for MySQL cluster.
Account Name	The name of the database account used to connect to the destination AnalyticDB for MySQL cluster. You can use either of the following accounts: <ul style="list-style-type: none"> Privileged accounts Standard accounts
Account Password	The password of the database account used to connect to the destination AnalyticDB for MySQL cluster.
Field Mapping	Log Service automatically extracts all the log fields of the last ten minutes and maps these fields to the destination fields in AnalyticDB for MySQL.

Parameter	Description
Delivery Start Time	The start time of the LogShipper task. After log data is written into Log Service, Log Service can transfer the data to AnalyticDB for MySQL in real time.
Filter Dirty Data	Specifies whether to filter dirty data. <ul style="list-style-type: none"> ◦ Disable: automatically interrupts the LogShipper task if dirty data is found. ◦ Enable: filters dirty data. <p>Dirty data refers to data whose data type fails to be converted or whose required fields are empty.</p>

5. After you configure the preceding parameters, click **OK**. Log Service will transfer log data to AnalyticDB for MySQL tables in real time at the specified delivery time.

View log data

After logs are transferred to AnalyticDB for MySQL, you can execute the **SELECT** statement to analyze log data.

Manage a LogShipper task

- You can click the task name to view the execution status of the LogShipper task. For example, you can view the volume of data that is transferred and fails to be transferred and delivery latency.
- To modify a LogShipper task, click **Modify LogShipper**.
- To restart a LogShipper task, first click **Stop**. After the LogShipper task is stopped, click **Start** to restart the LogShipper task.

3. Submit import or export tasks asynchronously

AnalyticDB for MySQL allows you to submit import or export tasks asynchronously.

Submit a task asynchronously

- Syntax

```
submit job insert overwrite into xxx select ...
```

After you execute the preceding statement, a job ID is returned.

- Example

```
mysql> submit job insert overwrite into test select * from test_external_table;
+-----+
| job_id          |
+-----+
| 2017112122202917203100908203303000715 |
```

Query the status of an asynchronous task

- Syntax

```
show job status where job='job_id'
```

- Example

```
mysql> show job status where job='2017112122202917203100908203303000715';
+-----+-----+-----+-----+-----+-----+-----+
| job_id          | schema_name | status | fail_msg | create_time      | update_time      | definition |
+-----+-----+-----+-----+-----+-----+-----+
| 2017112122202917203100908203303000715 | test        | RUNNING | NULL     | 2017-11-21 22:20:31.0 | 2017-11-21 22:20:40.0 | insert into test select * from test |
```

- Task status

- **INIT** : The task enters the queue.
- **RUNNING** : The task is running.
- **FINISH** or **FAILED** : The task succeeds or fails.

Terminate a task

- Syntax

```
cancel job 'job_id'
```

- **Example**

```
mysql> cancel job '2017112122202917203100908203303000715';
```

- **Description**

- Unscheduled, failed, and successful tasks are removed from the queue.
- Running tasks are terminated. Imported data is rolled back.


4. Export Data

4.1. Export data from AnalyticDB for MySQL to MySQL

This topic describes how to use AnalyticDB for MySQL external tables and the INSERT INTO statement to export data from AnalyticDB for MySQL to MySQL.

Prerequisites

- An ApsaraDB RDS for MySQL instance is created, a whitelist is configured, and an account, a databases, and tables are created. For more information, see [Quick start of ApsaraDB RDS MySQL](#).

 **Note** The ApsaraDB RDS for MySQL instance and AnalyticDB for MySQL cluster must be in the same VPC. You must add the CIDR block of this VPC to the whitelist of the ApsaraDB RDS for MySQL instance. In this case, the AnalyticDB for MySQL database can use the VPC endpoint of the ApsaraDB RDS for MySQL instance to access the ApsaraDB RDS for MySQL database.

In this example, a table named courses is created in the test_adb database in ApsaraDB RDS for MySQL to store the data exported from AnalyticDB for MySQL.

```
CREATE TABLE courses (  
  id bigint NOT NULL,  
  name varchar(32) NOT NULL,  
  grade varchar(32) NOT NULL,  
  submission_date timestamp NOT NULL,  
  PRIMARY KEY (id)  
)
```


- An AnalyticDB for MySQL cluster is created, a whitelist is configured, and an account and a database are created. For more information, see [Quick start of AnalyticDB for MySQL](#).

In this example, the data in the courses table of the adb_demo database in AnalyticDB for MySQL is exported to the courses table of the test_adb database in ApsaraDB RDS for MySQL.

```
create Table courses (  
  id bigint AUTO_INCREMENT,  
  name varchar NOT NULL,  
  grade varchar DEFAULT 'Grade 3',  
  submission_date timestamp  
) DISTRIBUTE BY HASH(id)
```

```
insert into courses (name,submission_date) values("Jams",NOW());
```

Context

 **Note** The MySQL database in this topic can be an ApsaraDB RDS for MySQL database or a user-created ECS-hosted MySQL database.

Procedure

1. Connect to an AnalyticDB for MySQL cluster and the target database. For more information, see [Connect to a cluster](#).
2. Use the `CREATE TABLE` statement to create an external mapping table named `courses_external_table` in the AnalyticDB for MySQL database named `adb_demo`.

```
CREATE TABLE IF NOT EXISTS courses_external_table(
  id bigint NOT NULL,
  name varchar(32) NOT NULL,
  grade varchar(32) NOT NULL,
  submission_date timestamp NOT NULL,
  PRIMARY KEY (id)
)
ENGINE='mysql'
TABLE_PROPERTIES='{
"url":"jdbc:mysql://mysql-vpc-address:3306/test_adb",
"tablename":"courses",
"username":"mysql-user-name",
"password":"mysql-user-password"
}'
```

Parameter	Description
<code>ENGINE='mysql'</code>	Specifies the table as an external table whose data engine is MySQL.
<code>TABLE_PROPERTIES</code>	Specifies the connection information used by the AnalyticDB for MySQL database to access the data in the ApsaraDB RDS for MySQL database.

Parameter	Description
url	<p>The VPC endpoint of the ApsaraDB RDS for MySQL instance.</p> <p>The endpoint is in the "jdbc:mysql://mysql-vpc-address:3306/rds-database-name" format.</p> <p>Note AnalyticDB for MySQL can access ApsaraDB RDS for MySQL only through the VPC endpoint.</p> <p>You can log on to the ApsaraDB RDS for MySQL console and view the internal endpoint of the ApsaraDB RDS for MySQL instance on the Basic Information page.</p>
tablename	The source table in the ApsaraDB RDS for MySQL database, which is the courses table in this example.
username	The account used to connect to the test_adb database in ApsaraDB RDS for MySQL. For more information, see Create databases and accounts for an ApsaraDB RDS for MySQL instance .
password	The password of the account used to connect to the ApsaraDB RDS for MySQL database.

- Execute the INSERT statement to export the data in AnalyticDB for MySQL to the ApsaraDB RDS for MySQL.


```
insert into courses_external_table
select * from courses;
```

4.2. Export data from AnalyticDB for MySQL to OSS

This topic describes how to use AnalyticDB for MySQL external tables and the INSERT INTO statement to export data from AnalyticDB for MySQL to Object Storage Service (OSS).


Prerequisites

- You have created a directory for storing AnalyticDB for MySQL data in OSS by completing the following steps:
 - [Activate OSS](#).

 **Note** Ensure that OSS and AnalyticDB for MySQL are in the same region.

- ii. **Create buckets.**
- iii. **Create folders.**

For example, create the `adb_data/` folder in the directory `bucket-name.oss-cn-hangzhou.aliyuncs.com/` in OSS. The data exported from AnalyticDB for MySQL is stored in this folder.


 **Note** When you export data from AnalyticDB for MySQL, the number of objects to be exported is automatically determined based on the transmission rate of concurrent tasks.

- You have created an AnalyticDB for MySQL cluster, configured a whitelist, and create accounts and databases for the cluster. For more information, see [Quick start of AnalyticDB for MySQL](#).

Procedure

1. **Connect to AnalyticDB for MySQL** to access the target database.


In this example, the data in the `adb_oss_import_test` table of the `adb_demo` database of AnalyticDB for MySQL is exported to the directory `bucket-name.oss-cn-hangzhou.aliyuncs.com/adb_data/` in OSS.

 **Note** For more information about the `adb_oss_import_test` table, see [Import data from OSS to AnalyticDB for MySQL](#).

2. Execute the CREATE TABLE statement to create an external mapping table named `oss_export_test_external_table` in the `adb_demo` database.

```
CREATE TABLE IF NOT EXISTS oss_export_test_external_table (
  uid string,
  other string
)
ENGINE='OSS'
TABLE_PROPERTIES='{
  "endpoint":"oss-cn-hangzhou-internal.aliyuncs.com",
  "url":"oss://bucket-name/adb_data/",
  "accessid":"LTAIF****5FsE",
  "accesskey":"Ccw****iWjv",
  "delimiter":";",
}'
```

Parameter	Description
ENGINE='OSS'	Specifies the table as an external table whose storage engine is OSS.

Parameter	Description
TABLE_PROPERTIES	Specifies the connection information used by AnalyticDB for MySQL to access and write data to OSS.
endpoint	<p>The endpoint of the OSS bucket.</p> <div style="background-color: #e0f2f1; padding: 5px; border: 1px solid #ccc;"> <p> Note AnalyticDB for MySQL can access OSS only from ECS instances over the VPCs.</p> </div> <p>You can log on to the OSS console, click the target bucket, and view the Endpoint on the Overview page.</p>
url	The URL of the folder in OSS. The URL must end with a forward slash (/).
accessid	The AccessKey ID used to access the directory <code>bucket-name.oss-cn-hangzhou.aliyuncs.com/adb_data/</code> in OSS.
accesskey	The AccessKey secret used to access the directory <code>bucket-name.oss-cn-hangzhou.aliyuncs.com/adb_data/</code> in OSS.
delimiter	The column delimiter of OSS objects.

- Execute the INSERT INTO statement in AnalyticDB for MySQL to export the data to OSS.

```
insert into oss_export_test_external_table select * from adb_oss_import_test
```

After the INSERT INTO statement is executed, you can log on to the OSS console to view the synchronized data in the `adb_data` folder.

4.3. Use external tables to export data to MaxCompute partitioned tables

This topic describes how to use external tables to export AnalyticDB for MySQL data to MaxCompute partitioned tables.

Prerequisites

- A destination table is created. For more information, see [Preparations](#) and [Quick Start](#).

The following table is created in MaxCompute by executing the table creation statement. For more information, see [Create a table](#). If you have created the destination table, skip this step.


- Create a table that has level-1 partitions

```
CREATE TABLE IF NOT EXISTS odps_table
(
  uid STRING,
  other STRING,
  ds STRING
)
PARTITIONED BY (ds STRING)
LIFECYCLE 3;
```

- Create a table that has level-2 partitions

```
CREATE TABLE IF NOT EXISTS odps_table
(
  uid STRING,
  other STRING,
  ds STRING
)
PARTITIONED BY (ds STRING,other STRING)
LIFECYCLE 3;
```

- An AnalyticDB for MySQL cluster is created, a whitelist is configured, and an account and a database are created. For more information, see [Quick start of AnalyticDB for MySQL](#).


 **Note** To export data to a MaxCompute partitioned table, you must specify a partition to which the data is written. Data can be exported to multiple partitions only by executing multiple SQL statements. MaxCompute supports a maximum of six levels of partitions. You can reference the preceding operations to create tables that have partitions of other levels.

Export data to a MaxCompute table that has level-1 partitions

Data from the `adb_table` table in AnalyticDB for MySQL is exported to the `odps_table` table that has level-1 partitions in MaxCompute in the following example:


1. Connect to an AnalyticDB for MySQL cluster and log on to the source database. For more information, see [Connect to a cluster](#).
2. Create an external table that maps to the destination table in MaxCompute.

```
CREATE TABLE odps_external_table
(
  uid string,
  other string,
  ds string
)
ENGINE='ODPS'
TABLE_PROPERTIES='{
  "endpoint":"http://service.odps.aliyun-inc.com/api",
  "accessid":"xxx",
  "accesskey":"xxx",
  "project_name":"xxx",
  "table_name":"odps_table",
  "partition_column":"ds"
}'
```

Parameter	Description
ENGINE='ODPS'	The engine used for the external table, which is MaxCompute.
TABLE_PROPERTIES	The connection information used by AnalyticDB for MySQL to access and write data to MaxCompute.
endpoint	<p>The endpoint of MaxCompute.</p> <div style="border: 1px solid #add8e6; padding: 5px; margin: 5px 0;"> <p> Note AnalyticDB for MySQL can access MaxCompute only by using the VPC endpoint.</p> </div> <p>For more information, see Configure endpoints.</p>
accessid	The AccessKey ID used to access the destination table in MaxCompute.
accesskey	The AccessKey secret used to access the destination table in MaxCompute.
project_name	The name of the project to which the destination table belongs in MaxCompute.
table_name	The name of the destination table in MaxCompute.
partition_column	The partition field.

3. Export data from the AnalyticDB for MySQL external table to the MaxCompute table that has level-

1 partitions.

 **Note** `adb_table_column` does not contain the partition fields.

```
insert [overwrite] into odps_external_table partition(ds='20200401')
select [adb_table_column, ...] from adb_table [where ...]
```


Export data to a MaxCompute table that has level-2 partitions

Data is exported from the `adb_table` table in AnalyticDB for MySQL to the `odps_table` table that has level-2 partitions in MaxCompute in the following example:

1. Connect to an AnalyticDB for MySQL cluster and log on to the source database. For more information, see [Connect to a cluster](#).
2. Create an external table that maps to the destination table in MaxCompute.

```
CREATE TABLE odps_external_table
(
  uid string,
  other string,
  ds string
)
ENGINE='ODPS'
TABLE_PROPERTIES='{
  "endpoint":"http://service.odps.aliyun-inc.com/api",
  "accessid":"xxx",
  "accesskey":"xxx",
  "project_name":"xxx",
  "table_name":"odps_table",
  "partition_column":"ds,other"
}'
```

3. Export data from the AnalyticDB for MySQL external table to the MaxCompute table that has level-2 partitions.

 **Note** `adb_table_column` does not contain the partition fields.

```
insert [overwrite] into odps_external_table partition(ds='20200401',other='hangzhou')
select [adb_table_column, ...] from adb_table [where ...]
```