ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

DataHub

Developer Guide

Product Version: v3.16.2

Document Version: 20220819

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.SDK overview

This topic describes the SDKs that are provided by DataHub.

Package name	Description
aliyun-sdk-datahub	The SDK is used to perform basic operations in DataHub. The SDK package provides API operations on basic objects in DataHub, including projects, topics, and shards.

DataHub provides SDKs for Java, Python, Go, and C++. The following sections describe how to use DataHub SDKs for Java and Python.

2.Prerequisites

2.1. Overview

Before you use MaxCompute SDKs, you must prepare the required environment.

- Create an Alibaba Cloud account that is authorized to access MaxCompute and obtain an AccessKey pair that consists of an AccessKey ID and an AccessKey secret.
- Obtain the endpoint of MaxCompute.

2.2. Obtain an AccessKey pair

2.2.1. Log on to the Apsara Uni-manager Management Console

This topic describes how to log on to the Apsara Uni-manager Management Console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
   - It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
     a. On the Bind Virtual MFA Device page, bind an MFA device.

b. Enter the account and password again as in Step 2 and click **Log On**.

c. Enter a six-digit MFA verification code and click **Authenticate**.

- You have enabled MFA and bound an MFA device.

  Enter a six-digit MFA authentication code and click **Authenticate**.

> **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Operations Console User Guide*.

2.2.2. Obtain the AccessKey pair of an organization

This topic describes how to obtain the AccessKey pair of an organization.

Prerequisites

Only the operation administrators and level-1 organization administrators can obtain the AccessKey pair of an organization.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the Organizations navigation tree, find the organization that you want to add and click the icon on the right of an organization.
5. Select **AccessKey** from the drop-down list.
6. In the message that appears, view the AccessKey pair of the organization.

> **Note** An AccessKey pair is automatically allocated to a level-1 organization. Subordinate organizations of the level-1 organization use the AccessKey pair of the level-1 organization.

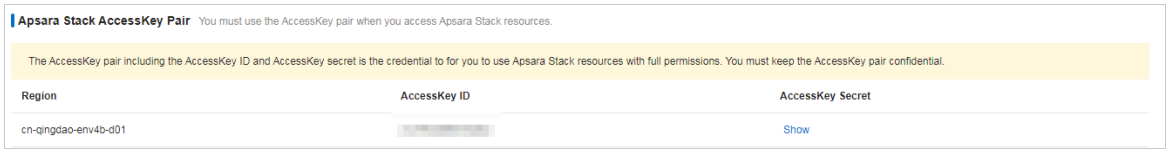
2.2.3. Obtain the AccessKey pair of an Alibaba Cloud account

To secure cloud resources, the system must verify the identity of visitors and ensure that they have the relevant permissions. You must obtain the AccessKey ID and AccessKey secret of your Alibaba Cloud account to access cloud resources. This topic describes how to obtain the AccessKey pair of an Alibaba Cloud account.

Procedure

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the upper-right corner of the homepage, move the pointer over the profile picture, and click **User Information**.

3. In the **Apsara Stack AccessKey Pair** section, you can view your AccessKey pair.



> **Note** The AccessKey pair consists of an AccessKey ID and an AccessKey secret. AccessKey pairs allow you to access Apsara Stack resources with full permissions for your account. You must keep your AccessKey pair confidential.

2.3. Obtain an endpoint

This topic describes how to obtain the endpoint of a DataHub machine in the Apsara Uni-manager Operations Console.

Context

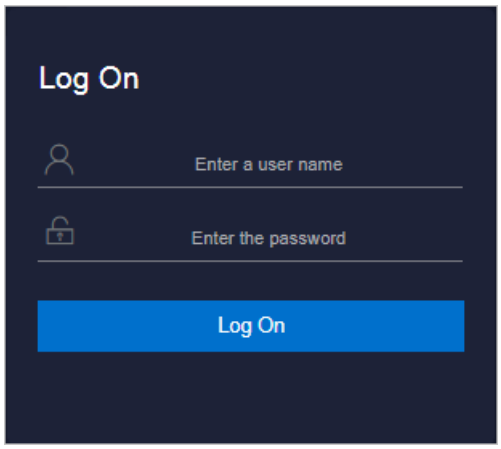
- The URL, username, and password that are used to log on to the Apsara Uni-manager Operations Console are obtained from the deployment engineers or administrators.

  The URL of the Apsara Uni-manager Operations Console is in the following format: *ops*.asconsole.*intranet-domain-id*.com.

- A browser is available. We recommend that you use the Google Chrome browser.

Procedure

1. Open your browser. In the address bar, enter the URL (*ops*.asconsole.*intranet-domain-id*.com). Then, press the Enter key.



> **Note** You can select a language from the drop-down list in the upper-right corner of the page.

2. Enter your username and password.

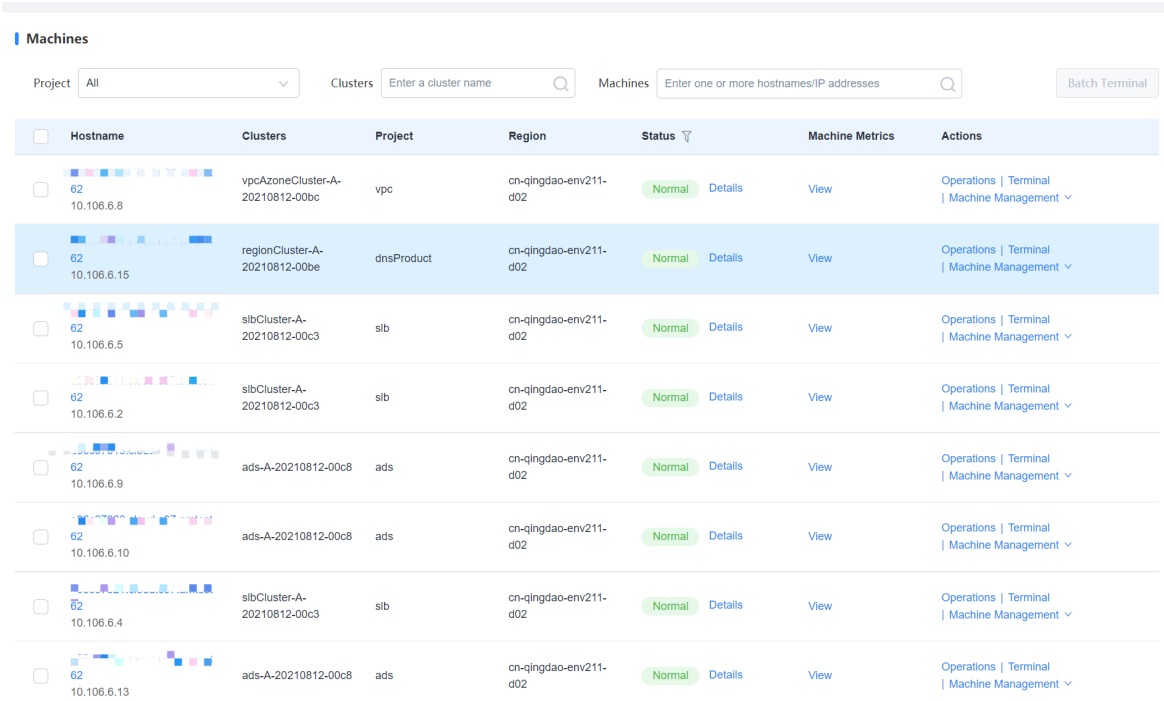
> **Note** To obtain the username and password that are used to log on to the Apsara Uni-manager Operations Console, contact the deployment engineers or administrators.

If you log on to the Apsara Uni-manager Operations Console for the first time, you must change the password of your username.

To ensure the security of your account, make sure that the password meet the following requirements:

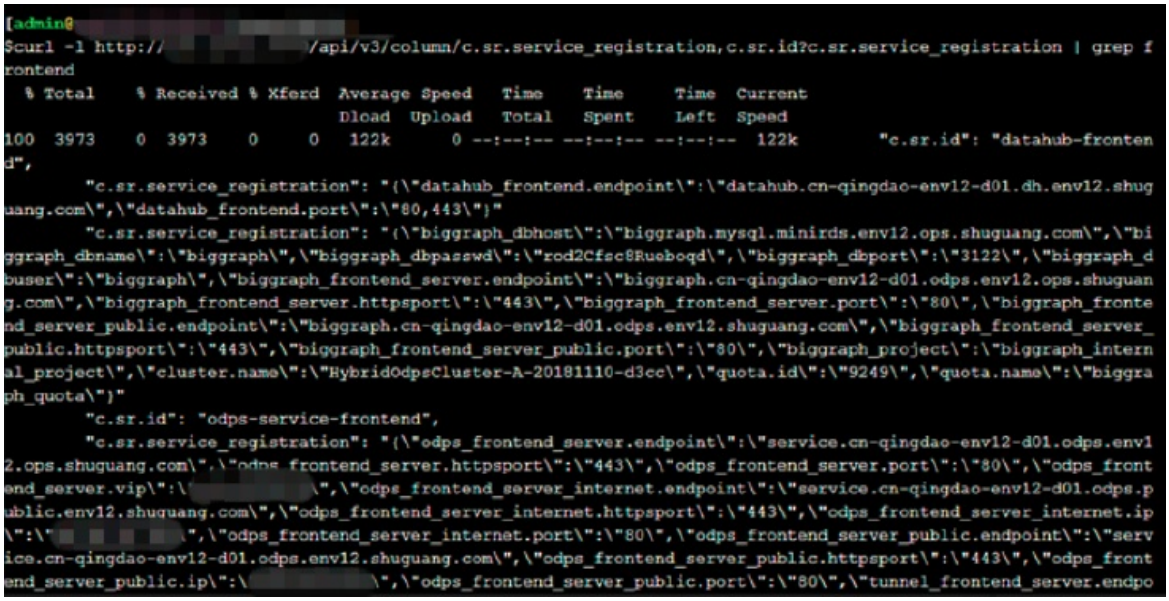
- The password contains uppercase or lowercase letters.
- The password contains digits.
- The password contains special characters such as exclamation points (!), at signs (@), number signs (#), dollar signs ($), and percent signs (%).
- The password is 10 to 20 characters in length.

3. Click **Log On** to go to the Apsara Uni-manager Operations Console.
4. In the top navigation bar of the Apsara Uni-manager Operations Console, click **O&M**. In the left-side navigation pane, choose **Product Management > Products**. In the **Apsara Stack O&M** section, click **Apsara Infrastructure Management Framework**.
5. In the left-side navigation pane, choose **Operations > Machine Operations**. The **Machine Operations** page appears.
6. Find the machine to which you want to log on, and click **Terminal** in the Actions column.

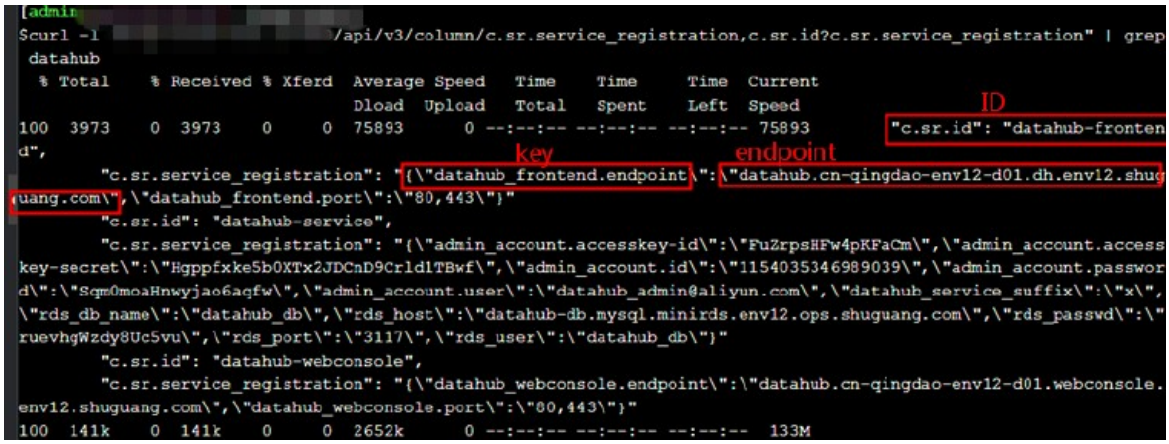


7. Run the following command on the machine to obtain JSON data:

```
curl -l "http://127.0.0.1:7070/api/v3/column/c.sr.service_registration,c.sr.id?c.sr.service_registration"
```



8. Find the endpoint of *DataHub* based on the service ID (*datahub-frontend*) and the key value (*datahub_frontend.endpoint*) of *DataHub*.



3.DataHub SDK for Java

3.1. Overview

You do not need to download the DataHub SDK for Java. You only need to configure the SDK as a Maven dependency in a pom.xml file. Add the following dependency configuration to the pom.xml file:

```
<dependency>
    <groupId>com.aliyun.datahub</groupId>
    <artifactId>aliyun-sdk-datahub</artifactId>
    <version>2.9.2-public</version>
</dependency>
```

Note If you use DataHub V3.8.1 or earlier, use the preceding dependency configuration, where the SDK version is 2.9.2.

```
<dependency>
    <groupId>com.aliyun.datahub</groupId>
    <artifactId>aliyun-sdk-datahub</artifactId>
    <version>2.13.1-public</version>
</dependency>
```

Note If you use DataHub V3.8.1 or later, use the preceding dependency configuration, where the SDK version is 2.13.1.

3.2. Preparations

To access DataHub, you must have an account that is authorized to access DataHub, the AccessKey for the account, and an endpoint of DataHub. If you are using Apsara Stack V2, obtain the endpoint from the CMDB system. If you are using Apsara Stack V3, obtain the endpoint from the Apsara Infrastructure Management Framework console.

The sample code is shown as follows:

```
String accessId = "Your AccessId";
String accessKey = "Your AccessKey";
String endpoint = "http://XXXXXX";
// Specify the endpoint you have obtained.
AliyunAccount account = new AliyunAccount(accessId, accessKey);
DatahubConfiguration conf = new DatahubConfiguration(account, endpoint);
```

The following example shows how to initialize a DataHub client:

```
DatahubClient client = new DatahubClient(conf);
```

Note After the DataHub client is initialized, you can call the DatahubClient operation to access DataHub. You can use this client to perform all operations on DataHub because the client is thread-safe.

3.3. Ingest tuple data

3.3.1. Create a tuple topic

The sample code is shown as follows:

```
RecordSchema schema = new RecordSchema();
schema.addField(new Field("a", FieldType.STRING));
schema.addField(new Field("b", FieldType.BIGINT));
int shardCount = 5;
int lifeCycle = 3;
String topicName = "topic_example";
String topicDesc = "topic_example_desc";
client.createTopic(projectName, topicName, shardCount, lifeCycle, RecordType.TUPLE, schema,
topicDesc);
// Wait until the shards are in Active status.
client.waitForShardReady(projectName, topicName);
```

Note In the sample code, lifeCycle indicates the time-to-live of each record in the topic.

3.3.2. Obtain the list of shards

The sample code is shown as follows:

```
ListShardResult listShardResult = client.listShard(projectName, topicName);
```

3.3.3. Write data into DataHub

The sample code is shown as follows:

```
List<RecordEntry> recordEntries = new ArrayList<RecordEntry>();
// Write into DataHub starting from any active shard returned by listShardResult. In this e
xample, start from the first active shard.
String shardId = listShardResult.getShards().get(0).getShardId();
RecordEntry entry = new RecordEntry(schema);
entry.setString(0, "Test");
entry.setBigint(1, 5L);
entry.setShardId(shardId);
recordEntries.add(entry);
PutRecordsResult result = client.putRecords(projectName, topicName, recordEntries);
if (result.getFailedRecordCount() ! = 0) {
    List<ErrorEntry> errors = result.getFailedRecordError();
    // Process records failed to be written into DataHub.
    for (ErrorEntry e : result.getFailedRecordError()) {
       if (! e.getErrorcode().equals("MalformedRecord")) { // Retry the operation except th
at the MalformedRecord, NoSuchProject, and NoSuchTopic errors occur.}
}
```

Note A request body size cannot exceed 4 MB. For more information, see Data ingestion methods.

3.3.4. Consume data

You can use this SDK to obtain the cursor that points to the first record to be consumed. The cursor type can be **OLDEST**, **LATEST**, or **SYSTEM_TIME**. OLDEST: the cursor that points to the earliest record in DataHub. LATEST: the cursor that points to the latest record in DataHub. SYSTEM_TIME: the cursor that points to the first record ingested after the specified time.

The sample code is shown as follows:

```
GetCursorResult cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
// If you want to obtain the cursor that points to the earliest record ingested during the last 24 hours, set CursorType to SYSTEM_TIME and set SYSTEM_TIME to System.currentTimeMillis() - 24 * 3600 * 1000 /* ms */.
int limit = 100;
String cursor = cursorRs.getCursor();
while (true) {
    try {
        GetBlobRecordsResult recordRs = client.getBlobRecords(projectName, topicName, shardId, cursor, limit);
        List<BlobRecordEntry> recordEntries = recordRs.getRecords();
        if (recordEntries.size() == 0) {
            // Data has not been updated. Try again later.
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        // The cursor that points to the next record to be consumed is obtained.
        cursor = recordRs.getNextCursor();
    } catch (InvalidCursorException ex) {
        // The cursor is invalid or has expired. Specify another record as the beginning of consumption.
        cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
        cursor = cursorRs.getCursor();
    } catch (DatahubClientException ex) {
        // An error occurred while obtaining the cursor. Try again.
        System.out.printf(ex.getMessage());
        ex.printStackTrace();
    }
}
```

3.4. Ingest blob data

3.4.1. Create a blob topic

The sample code is shown as follows:

```
int shardCount = 5;
int lifeCycle = 3;
String topicName = "topic_example";
String topicDesc = "topic_example_desc";
client.createTopic(projectName, topicName, shardCount, lifeCycle, RecordType.BLOB, topicDesc);
// Wait until the shards are in Active status.
client.waitForShardReady(projectName, topicName);
```

3.4.2. Obtain the list of shards

The sample code is shown as follows:

```
ListShardResult listShardResult = client.listShard(projectName, topicName);
```

3.4.3. Write data into DataHub

The sample code is shown as follows:

```
List<BlobRecordEntry> recordEntries = new ArrayList<BlobRecordEntry>();
// Write into DataHub starting from any active shard returned by listShardResult. In this e
xample, start from the first active shard.
String shardId = listShardResult.getShards().get(0).getShardId();
String data = String.valueOf(System.currentTimeMillis());
BlobRecordEntry entry = new BlobRecordEntry();
entry.setData(data.getBytes());
entry.setShardId(shardId);
recordEntries.add(entry);
PutBlobRecordsResult result = client.putBlobRecords(projectName, topicName, recordEntries);
if (result.getFailedRecordCount() ! = 0) {
    List<ErrorEntry> errors = result.getFailedRecordError();
    // Retry the operation except that the MalformedRecord, NoSuchProject, and NoSuchTopic
errors occur.
}
```

> **Note** A request body size cannot exceed 4 MB.

3.4.4. Consume data

The sample code is shown as follows:

```
GetCursorResult cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
// If you want to obtain the cursor that points to the earliest record ingested during the last 24 hours, set CursorType to SYSTEM_TIME and set SYSTEM_TIME to System.currentTimeMillis() - 24 * 3600 * 1000 /* ms */.
int size = 100;
String endCursor = cursorRes.GetCursor();
while (true) {
    try {
        GetBlobRecordsResult recordRs = client.getBlobRecords(projectName, topicName, shardId, cursor, limit);
        List<BlobRecordEntry> recordEntries = recordRs.getRecords();
        if(records.size() == 0) {
            // Data has not been updated. Try again later.
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        // The cursor that points to the next record to be consumed is obtained.
        cursor = recordRs.getNextCursor();
    } catch (InvalidCursorException ex) {
        // The cursor is invalid or has expired. Specify another record as the beginning of consumption.
        cursorRs = client.getCursor(projectName, topicName, shardId, GetCursorRequest.CursorType.OLDEST);
        cursor = cursorRs.getCursor();
    } catch (DatahubClientException ex) {
        // An error occurred while obtaining the cursor. Try again.
        System.out.println(response.getMessage());
        e.printStackTrace();
    }
}
```

Note Before being consumed, blob records that have been base64 encoded during ingestion must be decoded.

3.5. Data ingestion methods

3.5.1. Ingest data based on shard IDs

By using this method, you can write data into a specific shard and ensure that data is written into each shard in a specified order.

The sample code is shown as follows:

```
// Create a client.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Specify records to be ingested by DataHub.
RecordSchema schema = client.getTopic("projectName", "topicName").getRecordSchema();
List<RecordEntry> recordEntries = new ArrayList<~>();
RecordEntry entry = new RecordEntry(schema);
for (int i=0; i<entry.getFieldCount(); i++) {
    entry.setBigint(i, 1);
}
// Write into DataHub starting from any active shard returned by listShardResult. In this e
xample, start from the first active shard.
String shardId = listShardResult.getShards().get(0).getShardId();
entry.setShardId(shardId);
recordEntries.add(entry);
// Write data into DataHub.
PutRecordsResult result = client.putRecords("projectName", "topicName", recordEntries);
```

3.5.2. Ingest data based on hash values

Specify a 128-bit hash value produced by MD5 message-digest algorithm. Use the hash value to map associated records to a shard based on the hash key range of the shards.

You cannot specify the ingestion order of records by using this method.

The sample code is shown as follows:

```
// Create a client.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Specify records to be ingested by DataHub.
RecordSchema schema = client.getTopic("projectName", "topicName").getRecordSchema();
List<RecordEntry> recordEntries = new ArrayList<~>();
RecordEntry entry = new RecordEntry(schema);
for (int i=0; i<entry.getFieldCount(); i++) {
    entry.setBigint(i, 1);
}
entry.setHashKey("7FFFFFFFFFFFFFFD7FFFFFFFFFFFFFFD");
recordEntries.add(entry);
// Write the data into DataHub.
PutRecordsResult result = client.putRecords("projectName", "topicName", recordEntries);
```

3.5.3. Ingest data based on partition keys

Specify a partition key as a string. DataHub creates an MD5 hash from the string and maps the records to shards based on the hash key range of the shards.

You do not need to provide hash values by using this method.

The sample code is shown as follows:

```
// Create a client.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Specify records to be ingested by DataHub.
RecordSchema schema = client.getTopic("projectName", "topicName").getRecordSchema();
List<RecordEntry> recordEntries = new ArrayList<~>();
RecordEntry entry = new RecordEntry(schema);
for (int i=0; i<entry.getFieldCount(); i++) {
    entry.setBigint(i, 1);
}
entry.setPartitionKey("TestPartitionKey");
recordEntries.add(entry);
// Write the data into DataHub.
PutRecordsResult result = client.putRecords("projectName", "topicName", recordEntries);
```

3.6. Create a DataConnector

3.6.1. Create a DataConnector for MaxCompute

The sample code is shown as follows:

```
public void createDataConnector () {
    // Create a DataConnector for MaxCompute
    // Configure the destination MaxCompute table.
    String odpsProject = "datahub_test";
    String odpsTable = "test_table";
    String odpsEndpoint = "http://service-all.ext.odps.example.com/api";
    String tunnelEndpoint = "http://dt-all.ext.odps.example.com";
    OdpsDesc odpsDesc = new OdpsDesc();
    odpsDesc.setProject(odpsProject);
    odpsDesc.setTable(odpsTable);
    odpsDesc.setOdpsEndpoint(odpsEndpoint);
    odpsDesc.setTunnelEndpoint(tunnelEndpoint);
    odpsDesc.setAccessId(accessId);
    odpsDesc.setAccessKey(accessKey);
    odpsDesc.setPartitionMode(OdpsDesc.PartitionMode.USER_DEFINE);
    // Select some or all columns in the topic to be archived to MaxCompute. The MaxCompute
table must have the same columns in the same order as selected.
    List<String> jobIds = new ArrayList<String>();
    columnFields.add("f1");
    // By default, the partition mode is set to UserDefine.
    // If you set the partition mode to SYSTEM_TIME or EVENT_TIME, you must complete the fo
llowing configuration.
    // The configuration begins.
    int timeRange = 15;  // The partitioning interval, in minutes. Minimum value: 15.
    odpsDesc.setPartitionMode(OdpsDesc.PartitionMode.SYSTEM_TIME);
    odpsDesc.setTimeRange(timeRange);
    Map<String, String> partitionConfig = new LinkedHashMap<String, String>();
    // The partition key must be in the %Y%m%d%H%M format.
    partitionConfig.put("pt", "%Y%m%d");
    partitionConfig.put("ct", "%H%M");
    odpsDesc.setPartitionConfig(partitionConfig);
    // The configuration ends.
    client.createDataConnector(projectName, topicName, ConnectorType.SINK_ODPS, columnField
s, odpsDesc);
    // You can obtain the DataConnector status periodically, such as every 15 minutes, to c
heck whether all shards are being archived properly to the destination platform.
    String shard = "0";
    GetDataConnectorShardStatusResult getDataConnectorShardStatusResult =
        client.getDataConnectorShardStatus(projectName, topicName, ConnectorType.SINK_ODPS,
shard);
    System.out.println(getDataConnectorShardStatusResult.getCurSequence());
    System.out.println(getDataConnectorShardStatusResult.getLastErrorMessage());
}
```

3.6.2. Create a DataConnector for AnalyticDB or ApsaraDB RDS for MySQL

Sample code:

```
public void createADSDataConnector () {
    // Create a DataConnector for AnalyticDB or ApsaraDB RDS for MySQL.
    // Configure the destination table.
    String dbHost = "127.0.0.1";
    int dbPort = 3306;
    String dbName = "db";
    String user = "123";
    String password = "123";
    String tableName = "table";
    DatabaseDesc desc = new DatabaseDesc();
    desc.setHost(dbHost);
    desc.setPort(dbPort);
    desc.setDatabase(dbName);
    desc.setUser(user);
    desc.setPassword(password);
    desc.setTable(tableName);
    // The maximum size of records that can be written to DataHub at a time. Unit: Byte.
    desc.setMaxCommitSize(512L);
    // Specify whether to ignore errors and continue the process.
    desc.setIgnore(true);
    // Select several or all columns in the topic to be synchronized to AnalyticDB or Apsar
aDB RDS for MySQL. The destination table must have the columns in the same order as selecte
d.
    List<String> columnFields = new ArrayList<String>();
    columnFields.add("f1");
    client.createDataConnector(projectName, topicName, ConnectorType.SINK_ADS, columnFields
, desc);
    // Alternative: client.createDataConnector(projectName, topicName, ConnectorType.SINK_M
YSQL, columnFields, odpsDesc);
    // You can obtain the DataConnector status periodically, such as every 15 minutes, to c
heck whether all shards are being properly synchronized to the destination table.
    String shard = "0";
    GetDataConnectorShardStatusResult getDataConnectorShardStatusResult =
        client.getDataConnectorShardStatus(projectName, topicName, ConnectorType.SINK_ADS,
shard);
    System.out.println(getDataConnectorShardStatusResult.getCurSequence());
    System.out.println(getDataConnectorShardStatusResult.getLastErrorMessage());
}
```

Note You cannot create DataConnectors for services that reside in virtual private clouds (VPCs) by using the SDK. You must create this type of DataConnectors in the DataHub console.

4.DataHub SDK for Python

4.1. Install DataHub SDK for Python

Run the following commands to install DataHub SDK for Python.

Quick installation

```
$ sudo pip install pydatahub
```

Install by using source code

```
$ git clone https://github.com/aliyun/aliyun-datahub-sdk-python.git
$ cd aliyun-datahub-sdk-python
$ sudo python setup.py install
```

Verify the installation

```
$ python -c "from datahub import DataHub"
```

4.2. Preparations

To access DataHub, you must have an account that is authorized to access DataHub, the AccessKey for the account, and an endpoint of DataHub. If you are using Apsara Stack V2, obtain the endpoint from the CMDB system. If you are using Apsara Stack V3, obtain the endpoint from the Apsara Infrastructure Management Framework console.

The following example shows how to initialize DataHub configurations:

```
import sys
import traceback
from datahub import DataHub
from datahub.utils import Configer
from datahub.models import Topic, RecordType, FieldType, RecordSchema, BlobRecord, TupleRec
ord, CursorType
from datahub.errors import DatahubException, ObjectAlreadyExistException
access_id = ***your access id***
access_key = ***your access key***
endpoint = ***your datahub server endpoint***
dh = DataHub(access_id, access_key, endpoint)
```

4.3. Create topics

Create a tuple topic

You must specify a schema for a tuple topic. The following data types are supported:

Tuple data types

Type	Description	Value range
Bigint	An 8-byte signed integer. Note Do not use the minimum value (-9223372036854775808) because this is a system reserved value.	-9223372036854775807 to 9223372036854775807
String	A string. Only UTF-8 encoding is supported.	The size of a string must not exceed 1 MB.
Boolean	One of two possible values.	Valid values: True and False, true and false, or 0 and 1.
Double	An 8-byte double-precision floating point.	-1.0 10^{308} to 1.0 10^{308}
TimeStamp	A timestamp.	It is accurate to milliseconds.

The sample code is shown as follows:

```
topic = Topic(name=topic_name)
topic.project_name = project_name
topic.shard_count = 3
topic.life_cycle = 7
topic.record_type = RecordType.TUPLE
topic.record_schema = RecordSchema.from_lists(['bigint_field', 'string_field', 'double_fiel
d', 'bool_field', 'time_field'], [Fie
ldType.BIGINT, FieldType.STRING, FieldType.DOUBLE, FieldType.BOOLEAN, FieldType.TIMESTAMP])
try:
    dh.create_topic(topic)
    print "create topic success!"
    print "=======================================\n\n"
except ObjectAlreadyExistException, e:
    print "topic already exist!"
    print "=======================================\n\n"
except Exception, e:
    print traceback.format_exc()
    sys.exit(-1)
```

Create a blob topic

In a blob topic, a chunk of binary data is stored as a record. Records written into DataHub are Base64 encoded.

The sample code is shown as follows:

```
topic = Topic(name=topic_name)
topic.project_name = project_name
topic.shard_count = 3
topic.life_cycle = 7
topic.record_type = RecordType.BLOB
try:
    dh.create_topic(topic)
    print "create topic success!"
    print "=======================================\n\n"
except ObjectAlreadyExistException, e:
    print "topic already exist!"
    print "=======================================\n\n"
except Exception, e:
    print traceback.format_exc()
    sys.exit(-1)
```

4.4. Write records into DataHub and subscribe applications to records

Obtain the list of shards

You can use this SDK to obtain all shards in the topic.

```
shards = dh.list_shards(project_name, topic_name)
```

> **Note** A list is returned and each entry in the list describes a shard in terms of the ID, status, starting hash key, and ending hash key.

Write records into DataHub

You can use this SDK to write records into a topic in DataHub.

```
failed_indexs = dh.put_records(project_name, topic_name, records)
```

> **Note** The records request parameter is a list and each entry in the list represents a record. All records must be of the same data type, either tuple or blob. The subscript of the records that fail to be written into DataHub are returned.

The following sample code shows how to write tuple records into DataHub:

```
try:
    #Wait until all shards are in Active status.
    dh.wait_shards_ready(project_name, topic_name)
    print "shards all ready!!!"
    print "========================================\n\n"
    topic = dh.get_topic(topic_name, project_name)
    print "get topic suc! topic=%s" % str(topic)
    if topic.record_type ! = RecordType.TUPLE:
        print "topic type illegal!"
        sys.exit(-1)
    print "========================================\n\n"
    shards = dh.list_shards(project_name, topic_name)
    for shard in shards:
        print shard
    print "========================================\n\n"
    records = []
    record0 = TupleRecord(schema=topic.record_schema, values=[1, 'yc1', 10.01, True, 145586
9335000000])
    record0.shard_id = shards[0].shard_id
    record0.put_attribute('AK', '47')
    records.append(record0)
    record1 = TupleRecord(schema=topic.record_schema)
    record1['bigint_field'] = 2
    record1['string_field'] = 'yc2'
    record1['double_field'] = 10.02
    record1['bool_field'] = False
    record1['time_field'] = 1455869335000011
    record1.shard_id = shards[1].shard_id
    records.append(record1)
    record2 = TupleRecord(schema=topic.record_schema)
    record2['bigint_field'] = 3
    record2['string_field'] = 'yc3'
    record2['double_field'] = 10.03
    record2['bool_field'] = False
    record2['time_field'] = 1455869335000013
    record2.shard_id = shards[2].shard_id
    records.append(record2)
    failed_indexs = dh.put_records(project_name, topic_name, records)
    print "put tuple %d records, failed list: %s" %(len(records), failed_indexs)
    #If you specify the failed_indexs parameter, we recommend that you retry writing the fa
iled records into DataHub.
    print "========================================\n\n"
except DatahubException, e:
    print traceback.format_exc()
    sys.exit(-1)
else:
    sys.exit(-1)
```

Obtain the cursor of a record

You can use this SDK to obtain the cursor of a record. Valid values: **OLDEST**, **LATEST**, and **SYSTEM_TIME**.

- OLDEST: the cursor that points to the earliest record in DataHub.
- LATEST: the cursor that points to the latest record in DataHub.
- SYSTEM_TIME: the cursor that points to the first record ingested after the specified time.

```
cursor = dh.get_cursor(project_name, topic_name, CursorType.OLDEST, shard_id)
```

> **Note** If you select SYSTEM_TIME, this SDK returns the cursor that points to the record ingested at the specified time. If no record is ingested at the specified time, the first record ingested after the time is returned.

Consume records

You can use this SDK to specify the cursor that points to a record where the consumption begins and specify the upper limit of records to be consumed. If fewer records exist than the specified limit, the actual number of records are returned.

```
dh.get_records(topic, shard_id, cursor, 10)
```

The following sample code shows how to consume tuple records:

```
try:
    #Wait until all shards are in Active status.
    dh.wait_shards_ready(project_name, topic_name)
    print "shards all ready!!!"
    print "=======================================\n\n"
    topic = dh.get_topic(topic_name, project_name)
    print "get topic suc! topic=%s" % str(topic)
    if topic.record_type ! = RecordType.TUPLE:
        print "topic type illegal!"
        sys.exit(-1)
    print "=======================================\n\n"
    cursor = dh.get_cursor(project_name, topic_name, CursorType.OLDEST, '0')
    while True:
        (record_list, record_num, next_cursor) = dh.get_records(topic, '0', cursor, 10)
        for record in record_list:
            print record
        if 0 == record_num:
            time.sleep(1)
        cursor = next_cursor
except DatahubException, e:
    print traceback.format_exc()
    sys.exit(-1)
else:
    sys.exit(-1)
```

5.Scalability

5.1. Scenarios

You can increase or decrease the number of shards in a topic according to the service load.

For example, if the topic throughput cannot handle a surge in the service load during Double 11, you can split existing shards to up to 256 to increase the throughput to 256 MB/s.

As the service load decreases after Double 11, you can reduce the number of shards as needed by performing the merge operation.

5.2. Obtain shard details

You can call the ListShard operation to obtain details about all shards.

The sample response is shown as follows:

```
{
    "ShardId": "string",
    "State": "string",
    "ClosedTime": uint64,
    "BeginHashKey": "string",
    "EndHashKey": "string",
    "ParentShardIds": [string,string,],
    "LeftShardId": "string",
    "RightShardId": "string"
}
```

5.3. Split a shard

You can split a shard by using an SDK or in the console. You can specify the shard by setting a shard ID and a 128-bit hash key value. After the shard is split into two child shards, the IDs and key values of the child shards are returned. The status of the parent shard is changed to Deactivated.

For example, before the split operation, the parent shard details are shown as follows:

```
ShardId:0 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:FFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFF
```

Split the shard by using an SDK:

```
String shardId = "0";
SplitShardRequest req = new SplitShardRequest(projectName, topicName, shardId, "AAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA");
SplitShardResult resp = client.splitShard(req);
```

After the split operation, the parent shard turns into the following three shards:

```
ShardId:0 Status:Deactivated BeginHashKey:00000000000000000000000000000000 EndHashKey:FFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFF
ShardId:1 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA
ShardId:2 Status:ACTIVE BeginHashKey:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA EndHashKey:FFFFFFFFFF
FFFFFFFFFFFFFFFFFFFF
```

5.4. Merge shards

You can merge two adjacent shards by using an SDK or in the console. Shards are considered adjacent if the union of the hash key ranges for the two shards forms a contiguous set with no gaps. After the two shards are merged, the ID and key value of the new shard are returned. The statuses of the two parent shards change to Deactivated.

For example, before the merge operation, the details of the two parent shards are as follows:

```
ShardId:0 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:7FFFFFFFFF
FFFFFF7FFFFFFFFFFFFFFF
ShardId:1 Status:ACTIVE BeginHashKey:7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF EndHashKey:FFFFFFFFFF
FFFFFFFFFFFFFFFFFFFF
```

Merge the two shards by using an SDK:

```
String shardId = "0";
String adjacentShardId = "1";
MergeShardRequest req = new MergeShardRequest(projectName, topicName, shardId, adjacentShar
dId);
MergeShardResult resp = client.mergeShard(req);
```

After the merge operation, the parent shards turn into the following three shards:

```
ShardId:0 Status:Deactivated BeginHashKey:00000000000000000000000000000000 EndHashKey:7FFFF
FFFFFFFFFFF7FFFFFFFFFFFFFFF
ShardId:1 Status:Deactivated BeginHashKey:7FFFFFFFFFFFFFFF7FFFFFFFFFFFFFFF EndHashKey:FFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFF
ShardId:2 Status:ACTIVE BeginHashKey:00000000000000000000000000000000 EndHashKey:FFFFFFFFFF
FFFFFFFFFFFFFFFFFFFF
```

Note

- After merge or split operations are performed, the statuses of the parent shards change to Deactivated. New data cannot be written into deactivated shards, whereas existing data in deactivated shards can still be consumed. A deactivated shard cannot be split into two shards or merged with any other shards. When the time-to-live of the records in a deactivated shard expires, the shard is deleted.
- If a DataConnector is configured for a deactivated shard, the DataConnector will be stopped after all records in the shard are copied to the destination platform. The DataConnector is automatically deleted after the shard is deleted.
- The new shard can only be used after its status is changed to Active. It requires less than five seconds for the status to change to Active.

6.Insert a column to a topic schema

6.1. Insert a column

This section describes how to insert columns to a topic schema.

The sample code is shown as follows:

```
// Specify a DataHub account.
Account account = new AliyunAccount("your access id", "your access key");
DatahubConfiguration conf = new DatahubConfiguration(account, "datahub endpoint");
DatahubClient client = new DatahubClient(conf);
// Insert a bigint column to the topic schema.
client.appendField(new AppendFieldRequest(projectName, topicName, new Field("test", FieldTy
pe.BIGINT)));
// If you have created a DataConnector for MaxCompute, call the following operation.
client.appendDataConnectorField(new AppendDataConnectorFieldRequest(projectName, topicName,
ConnectorType.SINK_ODPS, "test"));
}
```

> **Note**
>
> - DataHub supports inserting columns only. Modifying or deleting columns is not supported.
> - The operation can only be performed on a DataConnector for MaxCompute: **Insert a column in MaxCompute > Insert a corresponding column in DataHub > Modify the corresponding DataConnector configuration.**