Data Lake Insight

Best Practice

Issue	01
Date	2023-03-31

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Overview

This document gives you best practices for data migration and analysis, helping you better use DLI for large-scale data analysis and processing.

Data Migration

You can use **Cloud Data Migration Service** (CDM) to easily migrate data from other cloud services or service platforms to DLI. You can refer to the following best practices:

- **Migrating Data from Hive to DLI**
- **Migrating Data from MRS Kafka to DLI**
- **Migrating Data from Elasticsearch to DLI**
- **Migrating Data from RDS to DLI**
- **Migrating Data from GaussDB(DWS) to DLI**

Data Analysis

DLI is widely used to analyze massive amounts of log data and in extract, transform, and load (ETL) processes, giving you great insight into data of a wide range of industries. You can refer to the following best practices of data analysis:

- **Analyzing Driving Behavior Data**
- **Converting Data Format from CSV to Parquet**
- **Analyzing E-commerce BI Reports**
- **Analyzing DLI Billing Data**

2 Data Migration

2.1 Overview

This section describes how you can migrate data to DLI in an efficient way. You can use **Cloud Data Migration Service** (CDM) to migrate data from other cloud services or platforms to DLI.

DLI is a serverless data processing and analysis service. It processes streaming data and batch data and supports interactive analysis. Its high-scalability framework supports the convergence of batch and streaming data analysis, and provides real-time, efficient, and diversified compute resources for TB-to EB-level data processing.

Best Practices of Data Migration

- You can migrate Hive data to DLI. For details, see **Migrating Data from Hive to DLI**.
- You can migrate Kafka data to DLI. For details, see **Migrating Data from MRS Kafka to DLI**.
- You can migrate Elasticsearch data to DLI. For details, see **Migrating Data from Elasticsearch to DLI**.
- You can migrate RDS data to DLI. For details, see **Migrating Data from RDS to DLI**.
- You can migrate GaussDB(DWS) data to DLI. For details, see **Migrating Data from GaussDB(DWS) to DLI**.

Data Type Mapping

If you migrate data from other cloud services or platforms to DLI, data types need to be converted and source and destination data must be mapped by type. **Table 2-1** lists the mapping relationships.

Table 2-1 Data type mapping

MySQL	Hive	GaussDB(DWS)	Oracle	PostgreSQL	Hologres	DLI Spark
CHAR	CHAR	CHAR	CHAR	CHAR	CHAR	CHAR
VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR	VARCHAR/ STRING
DECIMAL	DECIMAL	NUMERIC	NUMERIC	NUMERIC	DECIMAL	DECIMAL
INT	INT	INTEGER	NUMBER	INTEGER	INTEGER	INT
BIGINT	BIGINT	BIGINT	NUMBER	BIGINT	BIGINT	BIGINT/ LONG
TINYINT	TINYINT	SMALLINT	NUMBER	SMALLINT	SMALLINT	TINYINT
SMALLINT	SMALLINT	SMALLINT	NUMBER	SMALLINT	SMALLINT	SMALLINT/SHORT
BINARY	BINARY	BYTEA	RAW	BYTEA	BYTEA	BINARY
VARBINARY	BINARY	BYTEA	RAW	BYTEA	BYTEA	BINARY
FLOAT	FLOAT	FLOAT4	FLOAT	DOUBLE	FLOAT4	FLOAT
DOUBLE	DOUBLE	FLOAT8	FLOAT	REAL/ DOUBLE	FLOAT8	DOUBLE
DATE	DATE	TIMESTAMP	DATE	DATE	DATE	DATE
TIME	Not supported (use String instead)	TIME	DATE	TIME	TIME	Not supported (use String instead)
DATETIME	TIMESTAMP	TIMESTAMP	TIME	TIME	TIMESTAMP	TIMESTAMP
TINYINT	TINYINT	BOOLEAN	Not supported	TINYINT	BOOLEAN	BOOLEAN
Not supported (use TEXT instead)	Not supported (use String instead)	Not supported (use TEXT instead)	Not supported (use VARCHAR instead)	Not supported (use TEXT instead)	Not supported (use TEXT instead)	ARRAY

MySQL	Hive	GaussDB(DWS)	Oracle	PostgreSQL	Hologres	DLI Spark
Not supported (use TEXT instead)	Not supported (use String instead)	Not supported (use TEXT instead)	Not supported (use VARCHAR instead)	Not supported (use TEXT instead)	Not supported (use TEXT instead)	MAP
Not supported (use TEXT instead)	Not supported (use String instead)	Not supported (use TEXT instead)	Not supported (use VARCHAR instead)	Not supported (use TEXT instead)	Not supported (use TEXT instead)	STRUCT

NOTE

If a service does not support a standard data type, you can use the recommended data type.

2.2 Migrating Data from Hive to DLI

This section describes how to use the CDM data synchronization function to migrate data from MRS Hive to DLI. Data of other MRS Hadoop components can be bidirectionally synchronized between CDM and DLI.

Prerequisites

- You have created a DLI SQL queue.

  For details about how to create a DLI queue, see **Creating a DLI Queue**.

  CAUTION

  When you create a queue, set its **Type** to **For SQL**.

- You have created an MRS security cluster that contains the Hive component.

  For details about how to create an MRS cluster, see **Purchasing a Custom Cluster**.

  - In this example, the MRS cluster and component versions are as follows:
    - Cluster version: MRS 3.1.0
    - Hive version: 3.1.0
    - Hadoop version: 3.1.1
  - In this example, Kerberos authentication is enabled when the MRS cluster is created.
- You have created a CDM cluster.

  For details about how to create an MRS cluster, see **Creating a CDM Cluster**.

NOTE

- If the destination data source is an on-premises database, you need the Internet or Direct Connect. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- If the data source is MRS or GaussDB(DWS) on a cloud, the network must meet the following requirements:

  i. If the CDM cluster and the cloud service are in different regions, a public network or a dedicated connection is required for enabling communication between the CDM cluster and the cloud service. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules.

  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster are the same as those of the MRS cluster.

Step 1: Prepare Data

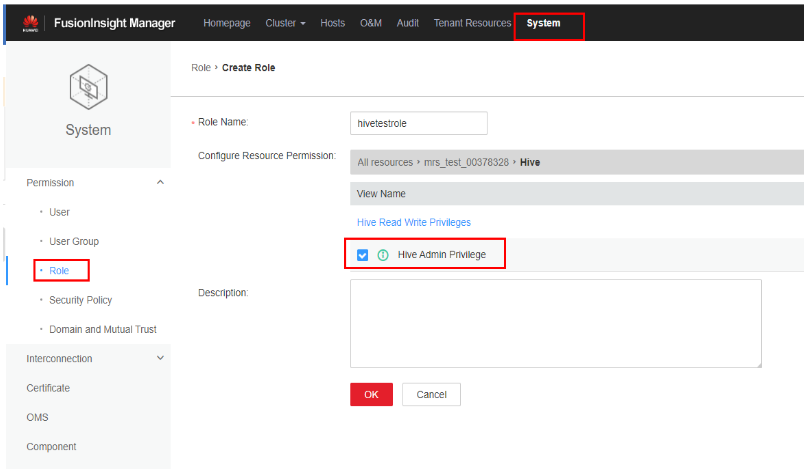
- Create a Hive table in the MRS cluster and insert data in the table.

  a. Log in to MRS Manager by referring to **Accessing FusionInsight Manager**.

  b. On MRS Manager, click **System** in the top navigation pane. On the page displayed, choose **Permission** > **Role** from the left navigation pane. On the displayed page, configure the following parameters:

     - **Role Name**: Enter a role name, for example, **hivetestrole**.
     - **Configure Resource Permission**: Select the MRS cluster name and then **Hive**. Select **Hive Admin Privilege**.

Figure 2-1 Creating a Hive role



* Role Name: hivetestrole

Configure Resource Permission: All resources › mrs_test_00378328 › Hive

View Name

Hive Read Write Privileges

☑ Hive Admin Privilege

Description:

c. On the MRS Manager console, click **System** in the top navigation pane. On the displayed page, choose **Permission** > **User** from the left navigation pane. On the displayed page, set the following parameters:

   i. **Username**: Enter a username. In this example, enter **hivetestusr**.

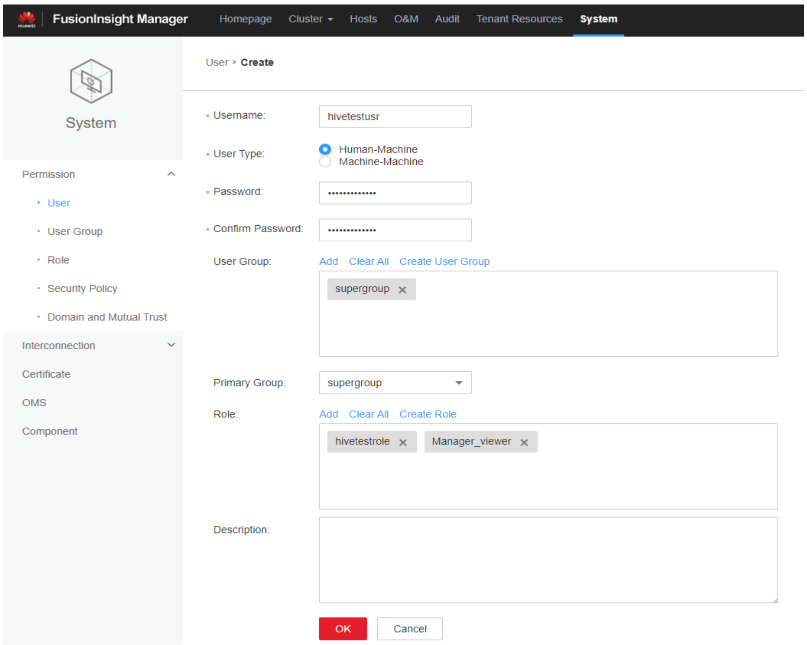
   ii. **User Type**: Select **Human-Machine**.

   iii. **Password** and **Confirm Password**: Enter the password of the current user and enter it again.

   iv. **User Group** and **Primary Group**: Select **supergroup**.

   v. **Role**: Select the role created in **b** and the **Manager_viewer** role.

Figure 2-2 Creating a Hive User



User › Create

* Username: hivetestusr
* User Type: ● Human-Machine ○ Machine-Machine
* Password: ••••••••••••••
* Confirm Password: ••••••••••••••

User Group: Add Clear All Create User Group
supergroup ×

Primary Group: supergroup

Role: Add Clear All Create Role
hivetestrole × Manager_viewer ×

Description:

OK Cancel

d. Download and install the Hive client by referring to **Installing an MRS Client**. For example, the Hive client is installed in the **/opt/hiveclient** directory on the active MRS node.

e. Go to the client installation directory as user **root**.

   For example, run the **cd /opt/hiveclient** command.

f. Run the following command to set environment variables:

   source bigdata_env

g. Run the following command to authenticate the user created in **c** as Kerberos authentication has been enabled for the current cluster:

   kinit *<Username in* ***c****>*

   Example: **kinit hivetestusr**

h. Run the following command to connect to Hive:

   beeline

i. Create a table and insert data into it.

   Run the following statement to create a table:

```
create table user_info(id string,name string,gender string,age int,addr string);
```

   Run the following statements to insert data into the table:

```
insert into table user_info(id,name,gender,age,addr) values("12005000201", "A", "Male", 19,
"City A");
insert into table user_info(id,name,gender,age,addr) values ("12005000202","B","male",20,"City
B");
insert into table user_info(id,name,gender,age,addr) values ("12005000202","B","male",20,"City
B");
```

NOTE

In the preceding example, data is migrated by creating a table and inserting data. To migrate an existing Hive database, run the following commands to obtain Hive database and table information:

- Run the following command on the Hive client to obtain database information:

  show databases

- Switch to the Hive database from which data needs to be migrated.

  use *Hive database name*

- Run the following command to display information about all tables in the database:

  show tables

- Run the following command to query the creation statement of the Hive table:

  show create table *table name*

  The queried table creation statements must be processed to comply with the DLI table creation syntax before being executed.

- Create a database and table on DLI.

  a. Log in to the DLI management console and click **SQL Editor**. On the displayed page, set **Engine** to **spark** and **Queue** to the created SQL queue.

     Enter the following statement in the editing window to create a database, for example, the migrated DLI database **testdb**:

```
create database testdb;
```

  b. Create a table in the database.

     NOTE

     You need to edit the table creation statement obtained by running **show create table** *hive table name* in MRS Hive to ensure the statement complies with the table creation syntax of DLI.

```
create table user_info(id string,name string,gender string,age int,addr string);
```

Step 2: Migrate Data

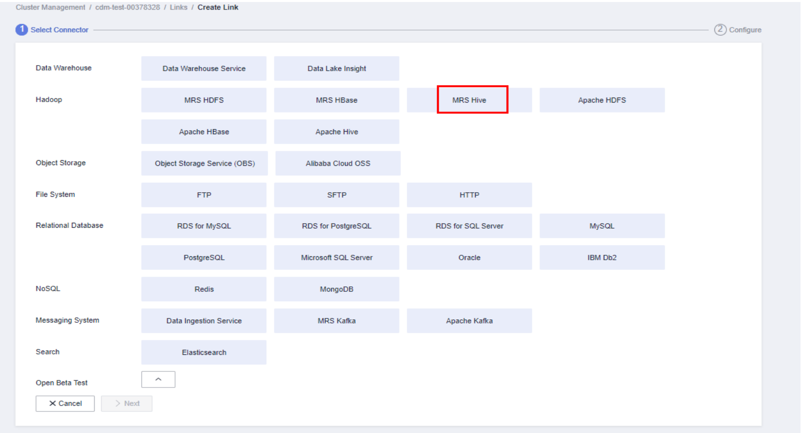
1. Create a CDM connection to MRS Hive.

   a. Create a connection to link CDM to the data source MRS Hive.

      i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

      ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **MRS Hive** and click **Next**.

Figure 2-3 Selecting the MRS Hive connector



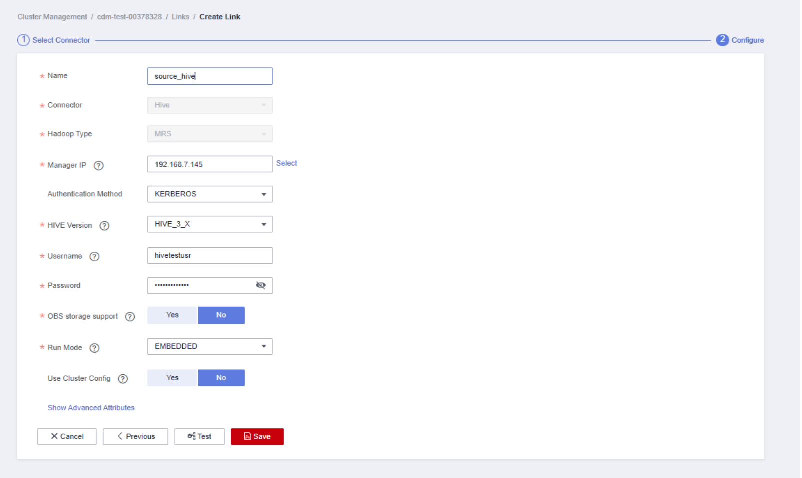
iii. Configure the connection. The following table describes the required parameters.

Table 2-2 MRS Hive connection configurations

Parameter	Value
Name	Name of the MRS Hive data source, for example, **source_hive**
Manager IP	Click **Select** next to the text box and select the MRS Hive cluster. The Manager IP address is automatically specified.
Authentication Method	Set this parameter to **KERBEROS** if Kerberos authentication is enabled for the MRS cluster. Set this parameter to **SIMPLE** if the MRS cluster is a common cluster. In this example, set this parameter to **KERBEROS**.
Hive Version	Set this parameter to the Hive version you have selected during MRS cluster creation. If the current Hive version is 3.1.0, set this parameter to **HIVE_3_X**.
Username	Name of the MRS Hive user created on **c**
Password	Password of the MRS Hive user

Retain default values for other parameters.

Figure 2-4 Configuring the connection to MRS Hive



* Name: source_hive

* Connector: Hive

* Hadoop Type: MRS

* Manager IP: 192.168.7.145

Authentication Method: KERBEROS

* HIVE Version: HIVE_3_X

* Username: hivetestusr

* Password: •••••••••••••

* OBS storage support: Yes [✓ No]

* Run Mode: EMBEDDED

Use Cluster Config: Yes [✓ No]

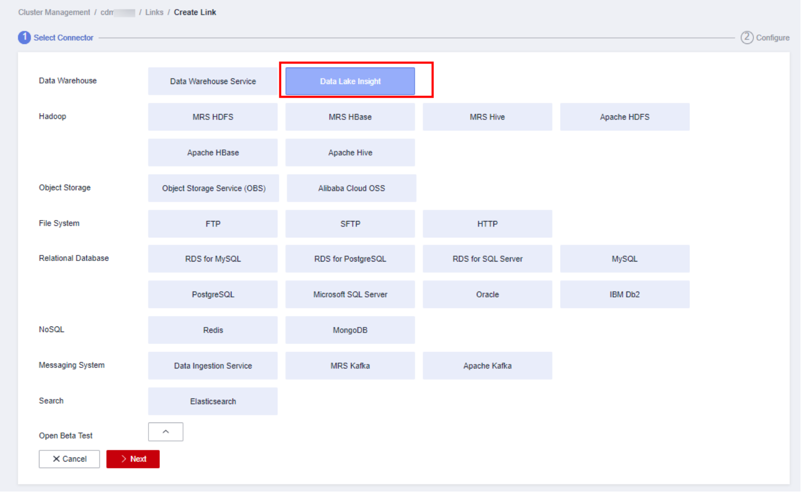
iv. Click **Save** to complete the configuration.

b. Create a connection to link CDM to DLI.

i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

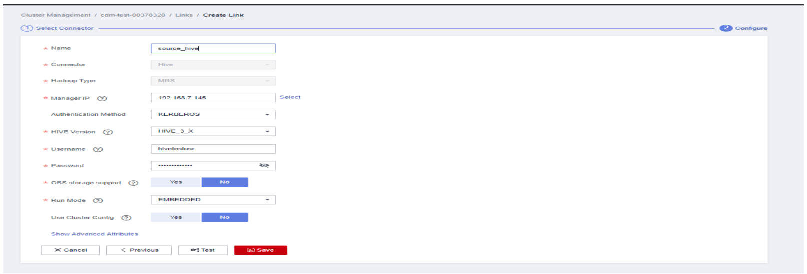
Figure 2-5 Selecting the DLI connector



Data Warehouse	Data Warehouse Service	Data Lake Insight		
Hadoop	MRS HDFS	MRS HBase	MRS Hive	Apache HDFS
	Apache HBase	Apache Hive		
Object Storage	Object Storage Service (OBS)	Alibaba Cloud OSS		
File System	FTP	SFTP	HTTP	
Relational Database	RDS for MySQL	RDS for PostgreSQL	RDS for SQL Server	MySQL
	PostgreSQL	Microsoft SQL Server	Oracle	IBM Db2
NoSQL	Redis	MongoDB		
Messaging System	Data Ingestion Service	MRS Kafka	Apache Kafka	
Search	Elasticsearch			

iii. Configure the connection parameters.

Figure 2-6 Configuring connection parameters

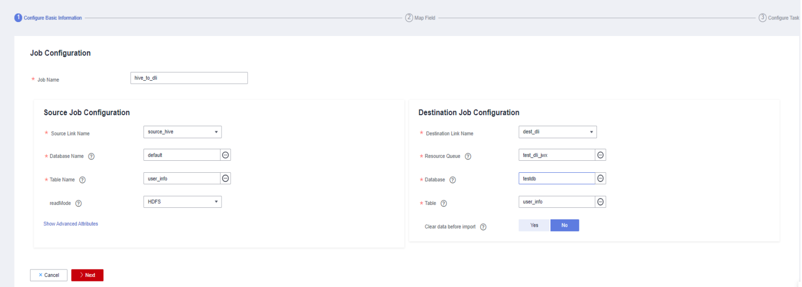


* Name	source_hive
* Connector	Hive
* Hadoop Type	MRS
* Manager IP	192.168.7.145
Authentication Method	KERBEROS
* HIVE Version	HIVE_3_X
* Username	hivetestusr
* Password	••••••••••••••
* OBS storage support	Yes [✓ No]
* Run Mode	EMBEDDED
Use Cluster Config	Yes [✓ No]

After the configuration is complete, click **Save**.

2. Create a CDM migration job.
   a. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.
   b. On the **Job Management** page, choose the **Table/File Migration** tab and click **Create Job**.
   c. On the **Create Job** page, specify job information.

Figure 2-7 Configuring the CDM job



Job Configuration

* Job Name	hive_to_dli

Source Job Configuration

* Source Link Name	source_hive
* Database Name	default
* Table Name	user_info
readMode	HDFS

Destination Job Configuration

* Destination Link Name	dest_dli
* Resource Queue	test_dli_jwx
* Database	testdb
* Table	user_info
Clear data before import	Yes [✓ No]

i. **Job Name**: Name of the data migration job, for example, **hive_to_dli**

ii. Set parameters required for **Source Job Configuration**.

Table 2-3 Source job configuration parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Database Name	Select the name of the MRS Hive database you want to migrate to DLI. For example, the **default** database.
Table Name	Select the name of the Hive table. In this example, a database created on DLI and the **user_info** table are selected.

Parameter	Value
readMode	In this example, **HDFS** is selected. Two read modes are available: HDFS and JDBC. By default, the HDFS mode is used. If you do not need to use the WHERE condition to filter data or add new fields on the field mapping page, select the HDFS mode. The HDFS mode shows good performance, but in this mode, you cannot use the WHERE condition to filter data or add new fields on the field mapping page. The JDBC mode allows you to use the WHERE condition to filter data or add new fields on the field mapping page.

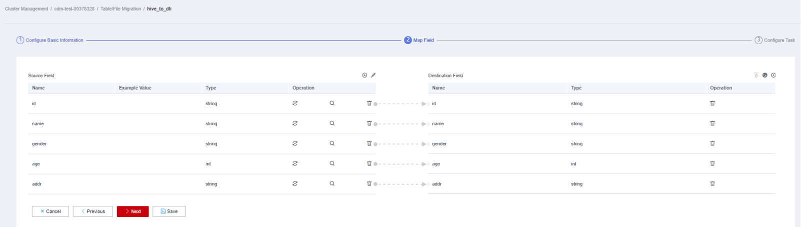
iii. Set parameters required for **Destination Job Configuration**.

Table 2-4 Destination job configuration parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection created in **1.b**.
Resource Queue	Select a created DLI SQL queue.
Database	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table	Select the name of a table in the database. In this example, table **user_info** created in **Create a database and table on DLI** is created.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set this parameter to **No**. If this parameter is set to **Yes**, data in the destination table will be cleared before the task is started.

3. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.
   - If the field mapping is incorrect, you can drag the fields to adjust the mapping.
   - If the type is automatically created at the migration destination, you need to configure the type and name of each field.
   - CDM allows for field conversion during migration.

Figure 2-8 Field mapping


Cluster Management / cdm-test-00378328 / Table/File Migration / hive_to_dli
① Configure Basic Information
② Map Field
③ Configure Task
× Cancel
< Previous
> Next
Save

Source Field

Name	Example Value	Type	Operation
id		string	
name		string	
gender		string	
age		int	
addr		string	

Destination Field

Name	Type	Operation
id	string	
name	string	
gender	string	
age	int	
addr	string	

4. Click **Next** and set task parameters. Generally, retain the default values of all parameters.

   In this step, you can configure the following optional functions:

   – **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.

   – **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.

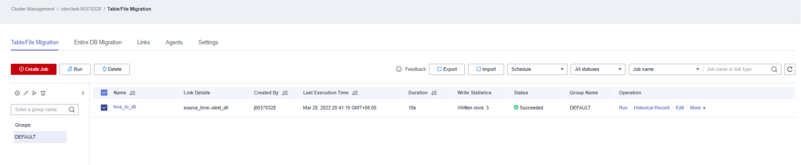
   – **Scheduled Execution**: Retain the default value **No**.

   – **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.

   – **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No** so that dirty data is not recorded.

5. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 2-9 Job progress and execution result



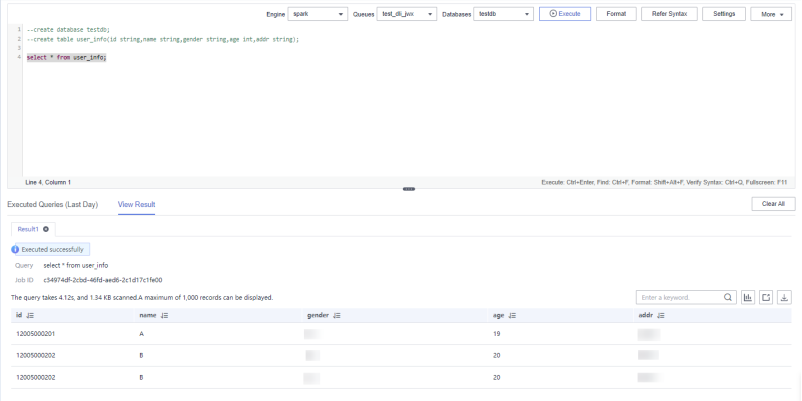
Name	Link Details	Created By	Last Execution Time	Duration	Write Statistics	Status	Group Name	Operation
hive_to_dli	source_hive--dest_dli	j00378328	Mar 28, 2022 20:41:10 GMT+08:00	18s	Written rows: 3	Succeeded	DEFAULT	Run | Historical Record | Edit | More

Step 3: Query Results

After the migration job is complete, log in to the DLI management console and click **SQL Editor**. In the displayed page, set **Engine** to **spark**, **Queue** to the created SQL queue, and **Database** to the database created in a. Execute the following query statement and check whether the Hive table data has been migrated to the **user_info** table:

```
select * from user_info;
```

Figure 2-10 Querying migrated data



Engine: spark
Queues: test_dli_jwx
Databases: testdb

```
--create database testdb;
--create table user_info(id string,name string,gender string,age int,addr string);

select * from user_info;
```

Line 4, Column 1

Execute: Ctrl+Enter, Find: Ctrl+F, Format: Shift+Alt+F, Verify Syntax: Ctrl+Q, Fullscreen: F11

Executed successfully

Query select * from user_info

Job ID c34974df-2cbd-46fd-aed6-2c1d17c1fe00

The query takes 4.12s, and 1.34 KB scanned.A maximum of 1,000 records can be displayed.

id	name	gender	age	addr
12005000201	A	[redacted]	19	[redacted]
12005000202	B	[redacted]	20	[redacted]
12005000202	B	[redacted]	20	[redacted]

2.3 Migrating Data from MRS Kafka to DLI

This section describes how to use the CDM data synchronization function to migrate data from MRS Kafka to DLI.

Prerequisites

- You have created a DLI SQL queue. For details about how to create a DLI queue, see **Creating a Queue**.

  ⚠ CAUTION

  When you create a queue, set its **Type** to **For SQL**.

- You have created an MRS security cluster that contains the Kafka component.
  - In this example, the version of the MRS cluster is 3.1.0.
  - You have enabled Kerberos authentication for the MRS cluster.
- You have created a CDM cluster.

NOTE

- If the destination data source is an on-premises database, you need the Internet or Direct Connect. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- If the data source is MRS or GaussDB(DWS), the network must meet the following requirements:

  i. If the CDM cluster and the cloud service are in different regions, a public network or a dedicated connection is required for enabling communication between the CDM cluster and the cloud service. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules.

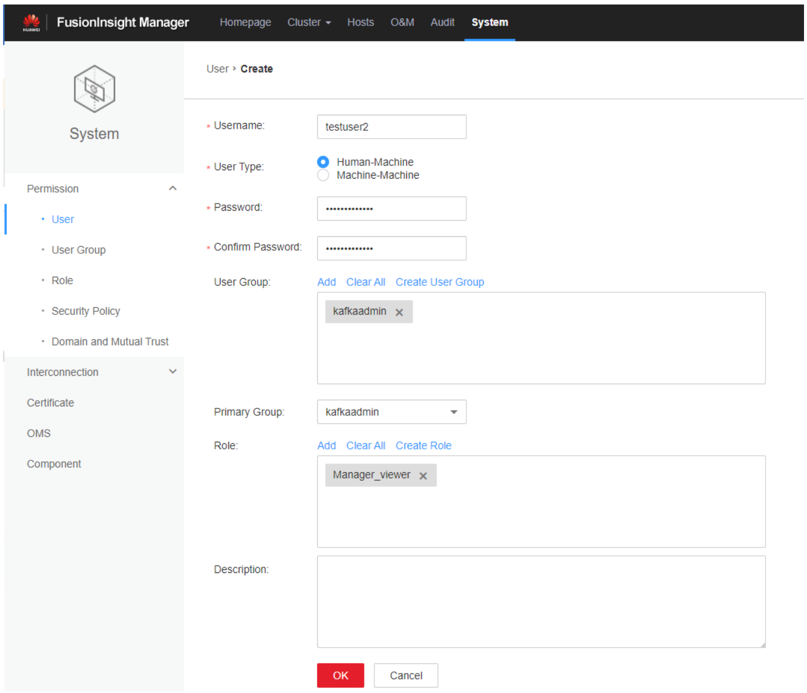
  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster are the same as those of the MRS cluster.

Step 1: Prepare Data

- Create a Kafka topic for the MRS cluster and send messages to the topic.
  a. Log in to MRS Manager by referring to **Accessing FusionInsight Manager**.
  b. On MRS Manager, click **System** in the top navigation pane. On the page displayed, choose **Permission** > **User** from the left navigation pane. On the displayed page, configure the following parameters:
     i. **Username**: Enter a username. In this example, enter **testuser2**.
     ii. **User Type**: Select **Human-Machine**.
     iii. **Password** and **Confirm Password**: Enter the password of the current user and enter it again.
     iv. **User Group** and **Primary Group**: Select **kafkaadmin**.
     v. **Role**: Select **Manager_viewer**.

Figure 2-11 Creating a Kafka user



* Username: testuser2

* User Type: ● Human-Machine ○ Machine-Machine

* Password: ••••••••••••

* Confirm Password: ••••••••••••

User Group: kafkaadmin ×

Primary Group: kafkaadmin

Role: Manager_viewer ×

Description:

c. On the MRS Manager console, choose **Cluster** > *Name of the desired cluster* > **Service** > **ZooKeeper** > **Instance**. On the displayed page, obtain the IP address of the ZooKeeper instance.

d. On the MRS Manager console, choose **Cluster** > *Name of the desired cluster* > **Service** > **Kafka** > **Instance**. On the displayed page, obtain the IP address of the Kafka instance.

e. Download and install the Kafka client by referring to **Installing an MRS Client**. For example, the Kafka client is installed in the **/opt/kafkaclient** directory on the active MRS node.

f. Go to the client installation directory as user **root**.

Example command: **cd /opt/kafkaclient**

g. Run the following command to set environment variables:

source bigdata_env

h. Run the following command to authenticate the user created in **b** since Kerberos authentication has been enabled for the cluster:

kinit <*Username in* ***b***>

Example command: **kinit testuser2**

i. Run the following command to create a Kafka topic named **kafkatopic**:

```
kafka-topics.sh --create --zookeeper IP address 1 of the node where the ZooKeeper role is:2181,IP address 2 of the node where the ZooKeeper role is:2181,IP address 3 of the node where the ZooKeeper role is:2181/kafka --replication-factor 1 --partitions 1 --topic kafkatopic
```

In this command, IP address of the node where the ZooKeeper role is deployed is that of the ZooKeeper instance obtained in **c**.

j. Run the following command to send a test message to **kafkatopic**:

```
kafka-console-producer.sh --broker-list IP address 1 of the node where the Kafka role is::21007;IP address 2 of the node where the Kafka role is::21007;IP address 3 of the node where the Kafka role is::21007 --topic kafkatopic --producer.config /opt/kafkaclient/Kafka/kafka/config/producer.properties
```

In this command, IP address of the node where the Kafka role is deployed in that of the Kafka instance obtained in **d**.

The content of the test message is as follows:

```
{"PageViews":5, "UserID":"4324182021466249494", "Duration":146,"Sign":-1}
```

- Create a database and table on DLI.
  a. Log in to the DLI management console and click **SQL Editor**. On the displayed page, set **Engine** to **spark** and **Queue** to the created SQL queue.

     Enter the following statement in the editing window to create a database. The following example creates the migrated DLI database **testdb**.

     ```
     create database testdb;
     ```

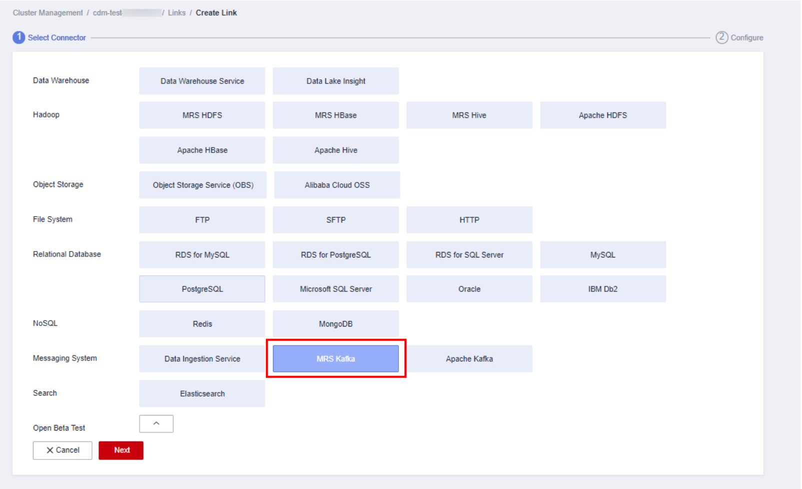
  b. Create a table in the database.

     ```
     CREATE TABLE testdlitable(value STRING);
     ```

Step 2: Migrate Data

1. Create a CDM connection to MRS Kafka.
   a. Create a connection to link CDM to the data source MRS Kafka.
      i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.
      ii. On the **Job Management** page, click the **Links** tab and click **Create Link**. On the displayed page, select **MRS Kafka** and click **Next**.

Figure 2-12 Selecting the MRS Kafka connector



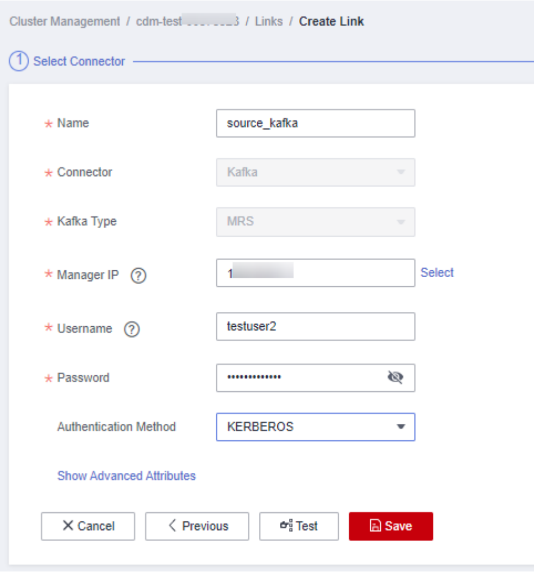
      iii. Configure the connection. The following table describes the required parameters.

Table 2-5 MRS Kafka connection configurations

Parameter	Value
Name	Name of the MRS Kafka data source, for example, **source_kafka**.

Parameter	Value
Manager IP	Manager IP address of the cluster. The value is automatically specified after you click **Select** next to the text box and select the MRS Kafka cluster.
Username	Name of the MRS Kafka user created in **b**.
Password	Password of the MRS Kafka user.
Authentication Method	**KERBEROS** if Kerberos authentication is enabled for the MRS cluster; **SIMPLE** if the MRS cluster is a common cluster In this example, set this parameter to **KERBEROS**.

Figure 2-13 Configuring the MRS Kafka connection



Cluster Management / cdm-test[redacted] / Links / Create Link

1 Select Connector

* Name: source_kafka

* Connector: Kafka

* Kafka Type: MRS

* Manager IP: 1[redacted]

* Username: testuser2

* Password: ••••••••••••••

Authentication Method: KERBEROS

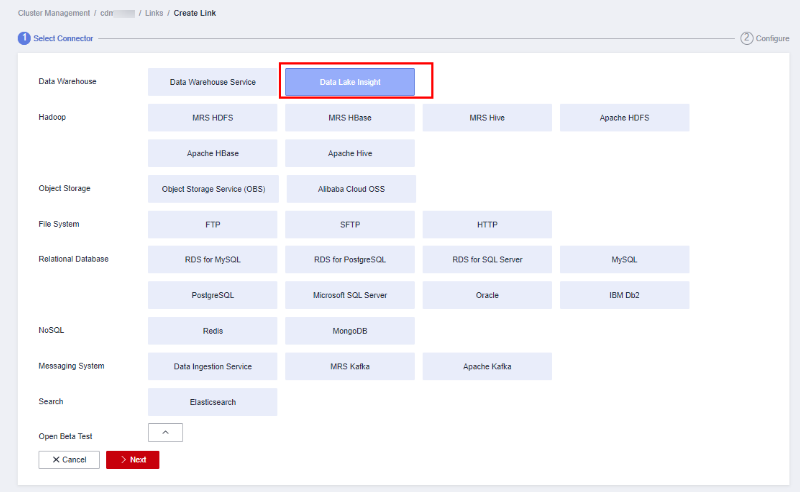
iv. Click **Save** to complete the configuration.

b. Create a connection to link CDM to DLI.

i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 2-14 Selecting the DLI connector



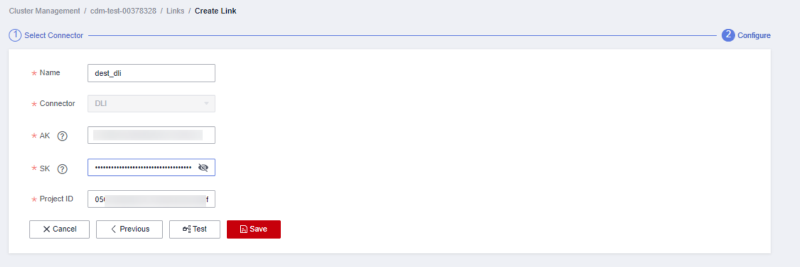
Cluster Management / cdm[redacted] / Links / Create Link

1 Select Connector — 2 Configure

Data Warehouse	Data Warehouse Service	Data Lake Insight		
Hadoop	MRS HDFS	MRS HBase	MRS Hive	Apache HDFS
	Apache HBase	Apache Hive		
Object Storage	Object Storage Service (OBS)	Alibaba Cloud OSS		
File System	FTP	SFTP	HTTP	
Relational Database	RDS for MySQL	RDS for PostgreSQL	RDS for SQL Server	MySQL
	PostgreSQL	Microsoft SQL Server	Oracle	IBM Db2
NoSQL	Redis	MongoDB		
Messaging System	Data Ingestion Service	MRS Kafka	Apache Kafka	
Search	Elasticsearch			
Open Beta Test	^			

iii. Configure the connection parameters.

Figure 2-15 Configuring connection parameters



Cluster Management / cdm-test-00378328 / Links / Create Link

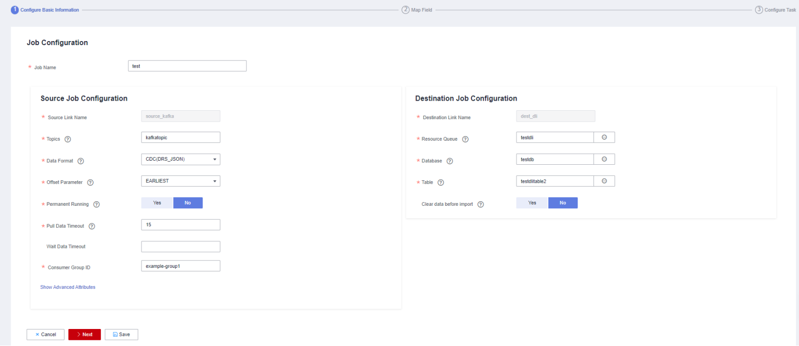
1 Select Connector — 2 Configure

* Name: dest_dli
* Connector: DLI
* AK: [redacted]
* SK: ••••••••••••••••••••••••••••••••
* Project ID: 05[redacted]f

iv. After the configuration is complete, click **Save**.

2. Create a CDM migration job.
   a. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster and click **Job Management** in the **Operation** column.
   b. On the **Job Management** page, choose the **Table/File Migration** tab and click **Create Job**.
   c. On the **Create Job** page, specify job information.

Figure 2-16 Configuring the CDM job



1 Configure Basic Information — 2 Map Field — 3 Configure Task

Job Configuration

* Job Name: test

Source Job Configuration

* Source Link Name: source_kafka
* Topics: kafkatopic
* Data Format: CDC(DRS_JSON)
* Offset Parameter: EARLIEST
* Permanent Running: Yes [✓ No]
* Pull Data Timeout: 15
Wait Data Timeout:
* Consumer Group ID: example-group1

Show Advanced Attributes

Destination Job Configuration

* Destination Link Name: dest_dli
* Resource Queue: testdli
* Database: testdb
* Table: testdlitable2
Clear data before import: Yes [✓ No]

i. **Job Name**: Name of the data migration job, for example, **test**

ii. Set parameters required for **Source Job Configuration**.

Table 2-6 Source job configuration parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Topics	Name of the topics you want to migrate to DLI. You can select one or more topics. Example: **kafkatopic**.
Data Format	Select the message format as needed. In this example, **CDC (DRS_JSON)** is selected, indicating that the source data will be parsed in DRS_JSON format.
Offset Parameter	Initial offset when data is pulled from Kafka. In this example, select **EARLIEST**. Available values are as follows: ● **Latest**: Maximum offset, indicating that the latest data will be extracted ● **Earliest**: Minimum offset, indicating that the earliest data will be extracted ● **Submitted**: Data that has been submitted ● **Time Range**: Data within a specified time range
Permanent Running	Whether a job runs permanently. In this example, set this parameter to **No**.
Pull Data Timeout	Maximum minutes allowed for a continuous data pulling. In this example, set this parameter to **15**.
Wait Data Timeout	(Optional) Maximum seconds allowed for waiting data reading. In this example, leave this parameter empty.
Consumer Group ID	Consumer group ID. The default Kafka message group ID **example-group1** is used.

iii. Set parameters required for **Destination Job Configuration**.

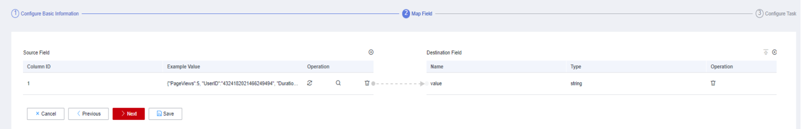
Table 2-7 Destination job configuration parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection created in **1.b**.

Parameter	Value
Resource Queue	Select a created DLI SQL queue.
Database	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table	Select the name of a table in the database. In this example, table **testdlitable** created in **Create a database and table on DLI** is selected.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set this parameter to **No**. If this parameter is set to **Yes**, data in the destination table will be cleared before the task is started.

3. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.
   - If the field mapping is incorrect, you can drag the fields to adjust the mapping.
   - If the type is automatically created at the migration destination, you need to configure the type and name of each field.
   - CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 2-17 Field mapping



Source Field

Column ID	Example Value	Operation
1	{"PageViews":5, "UserID":"4324182021466249494", "Duratio...	

Destination Field

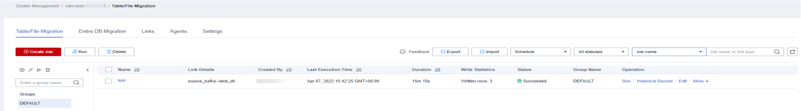
Name	Type	Operation
value	string	

4. Click **Next** and set task parameters. Generally, retain the default values of all parameters.

   In this step, you can configure the following optional functions:
   - **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
   - **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
   - **Scheduled Execution**: Retain the default value **No**.
   - **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
   - **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. In this example, retain the default value **No** so that dirty data is not recorded.

5. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 2-18 Job progress and execution result



Name	Link Details	Created By	Last Execution Time	Duration	Write Statistics	Status	Group Name	Operation
test	source_kafka--dest_dli	[redacted]	Apr 07, 2022 15:42:25 GMT+08:00	15m 15s	Written rows: 3	Succeeded	DEFAULT	Run | Historical Record | Edit | More

Step 3: Query Results

After the migration job is complete, log in to the DLI management console and click **SQL Editor**. In the displayed page, set **Engine** to **spark**, **Queue** to the created SQL queue, and **Database** to the database created in **a**. Execute the following query statement and check whether the Kafka table data has been migrated to the **testdlitable** table:

```
select * from testdlitable;
```

2.4 Migrating Data from Elasticsearch to DLI

This section describes how to use the CDM data synchronization function to migrate data from a CSS Elasticsearch cluster to DLI. Data in a self-built Elasticsearch cluster can also be bidirectionally synchronized between CDM and DLI.

Prerequisites

- You have created a DLI SQL queue.

  ⚠ CAUTION

  When you create a queue, set its **Type** to **For SQL**.

- You have created a CSS Elasticsearch cluster.

  In this example, the version of the created CSS cluster is 7.6.2, and security mode is disabled for the cluster.

- You have created a CDM cluster.

NOTE

- If the destination data source is an on-premises database, you need the Internet or Direct Connect. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- If the data source is CSS on a cloud, the network must meet the following requirements:

  i. If the CDM cluster and the cloud service are in different regions, a public network or a dedicated connection is required for enabling communication between the CDM cluster and the cloud service. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules.

  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster are the same as those of the CSS cluster.

Step 1: Prepare Data

- Create an index for the CSS cluster and import data.
  a. Log in to the CSS management console and choose **Clusters** > **Elasticsearch** from the navigation pane on the left.
  b. On the **Clusters** page, click **Access Kibana** in the **Operation** column of the created CSS cluster.
  c. In the navigation pane of Kibana, choose **Dev Tools**. The **Console** page is displayed.
  d. On the displayed **Console** page, run the following command to create index **my_test**:

```
PUT /my_test
{
  "settings": {
    "number_of_shards": 1
  },
  "mappings": {
          "properties": {
        "productName": {
          "type": "text",
          "analyzer": "ik_smart"
        },
        "size": {
          "type": "keyword"
        }
      }
    }
  }
```

  e. Run the following command to import data to the **my_test** index:

```
POST /my_test/_doc/_bulk
{"index":{}}
{"productName":"2017 Autumn New Shirts for Women", "size":"L"}
{"index":{}}
```

```
{"productName":"2017 Autumn New Shirts for Women", "size":"M"}
{"index":{}}
{"productName":"2017 Autumn New Shirts for Women", "size":"S"}
{"index":{}}
{"productName":"2018 Spring New Jeans for Women","size":"M"}
{"index":{}}
{"productName":"2018 Spring New Jeans for Women","size":"S"}
{"index":{}}
{"productName":"2017 Spring Casual Pants for Women","size":"L"}
{"index":{}}
{"productName":"2017 Spring Casual Pants for Women","size":"S"}
```

If **errors** is **false** in the command output, the data is imported.

- Create a database and table on DLI.

  a. Log in to the DLI management console and click **SQL Editor**. On the displayed page, set **Engine** to **spark** and **Queue** to the created SQL queue.

     Enter the following statement in the editing window to create a database, for example, the migrated DLI database **testdb**:

     ```
     create database testdb;
     ```

  b. Create a table in the database.

     ```
     create table tablecss(size string, productname string);
     ```

Step 2: Migrate Data

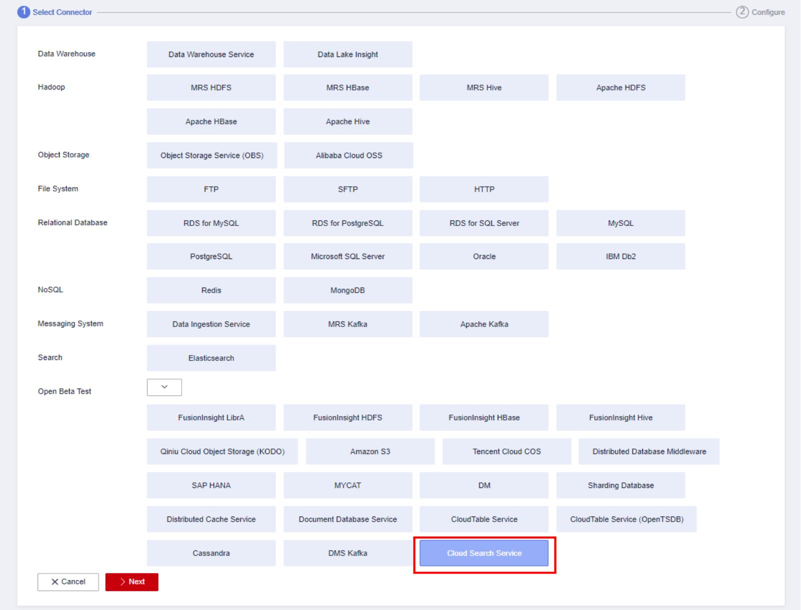
1. Create a CDM connection to MRS Hive.

   a. Create a connection to link CDM to the data source CSS.

      i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

      ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Cloud Search Service** and click **Next**.

Figure 2-19 Selecting the CSS connector

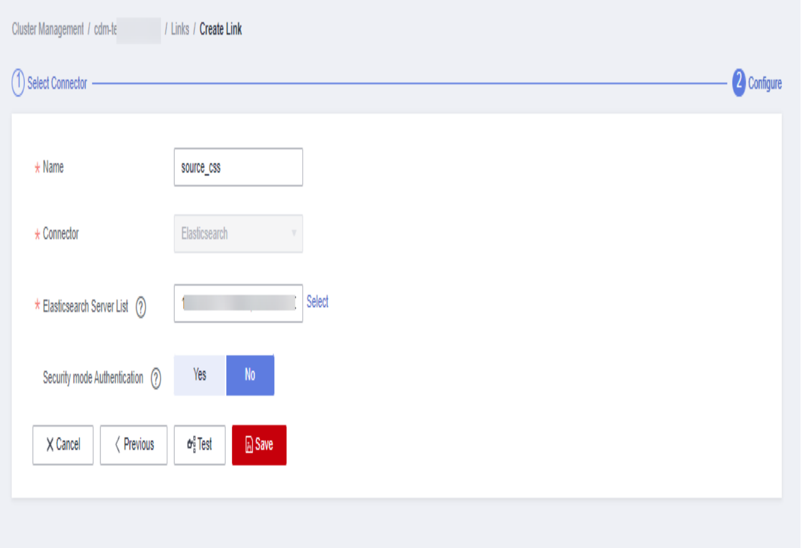


iii. Configure the connection. The following table describes the required parameters.

Table 2-8 CSS data source configuration

Parameter	Value.
Name	Name of the CSS data source, for example, **source_css**.
Elasticsearch Server List	Click **Select** next to the text box and select the CSS cluster. The Elasticsearch server list is automatically displayed.
Security mode Authentication	If you have enabled the security mode for the CSS cluster, set this parameter to **Yes**. Otherwise, set this parameter to **No**. In this example, set this parameter to **No**.

Figure 2-20 Configuring the CSS connection



Cluster Management / cdm-te[redacted] / Links / Create Link

1 Select Connector — 2 Configure

* Name: source_css

* Connector: Elasticsearch

* Elasticsearch Server List: [redacted] Select

Security mode Authentication: Yes [✓ No]

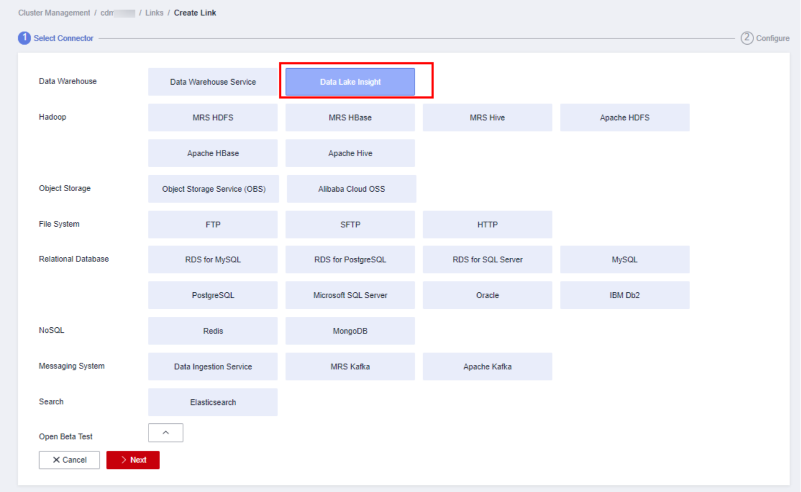
iv. Click **Save** to complete the configuration.

b. Create a connection to link CDM to DLI.

i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 2-21 Selecting the DLI connector



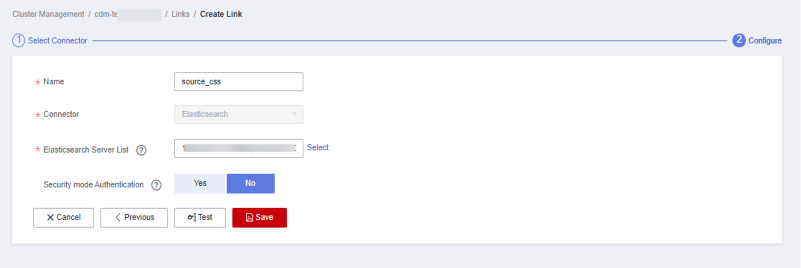
Cluster Management / cdm[redacted] / Links / **Create Link**

① Select Connector ② Configure

Data Warehouse	Data Warehouse Service	Data Lake Insight		
Hadoop	MRS HDFS	MRS HBase	MRS Hive	Apache HDFS
	Apache HBase	Apache Hive		
Object Storage	Object Storage Service (OBS)	Alibaba Cloud OSS		
File System	FTP	SFTP	HTTP	
Relational Database	RDS for MySQL	RDS for PostgreSQL	RDS for SQL Server	MySQL
	PostgreSQL	Microsoft SQL Server	Oracle	IBM Db2
NoSQL	Redis	MongoDB		
Messaging System	Data Ingestion Service	MRS Kafka	Apache Kafka	
Search	Elasticsearch			
Open Beta Test	^			

iii. Configure the connection parameters.

Figure 2-22 Configuring connection parameters



Cluster Management / cdm-te[redacted] / Links / **Create Link**

① Select Connector ② Configure

* Name: source_css

* Connector: Elasticsearch

* Elasticsearch Server List: 1[redacted] Select

Security mode Authentication: Yes [✓ No]

iv. After the configuration is complete, click **Save**.

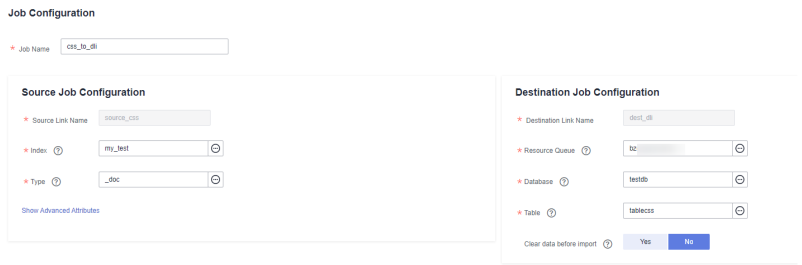
2. Create a CDM migration job.

   a. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

   b. On the **Job Management** page, choose the **Table/File Migration** tab and click **Create Job**.

   c. On the **Create Job** page, specify job information.

Figure 2-23 Configuring the CDM job



Job Configuration

* Job Name: css_to_dli

Source Job Configuration

* Source Link Name: source_css

* Index: my_test

* Type: _doc

Show Advanced Attributes

Destination Job Configuration

* Destination Link Name: dest_dli

* Resource Queue: bz[redacted]

* Database: testdb

* Table: tablecss

Clear data before import: Yes [✓ No]

i. **Job Name**: Name of the data migration job, for example, **css_to_dli**

ii. Set parameters required for **Source Job Configuration**.

Table 2-9 Source job configuration parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Index	Select the Elasticsearch index created for the CSS cluster. In this example, the **my_test index** created in **Create an index for the CSS cluster and import data** is used. The index can contain only lowercase letters.
Type	Elasticsearch type, which is similar to the table name of a relational database. The type name can contain only lowercase letters. Example: **_doc**.

For details about other parameters, see **From Elasticsearch or CSS**.

iii. Set parameters required for **Destination Job Configuration**.

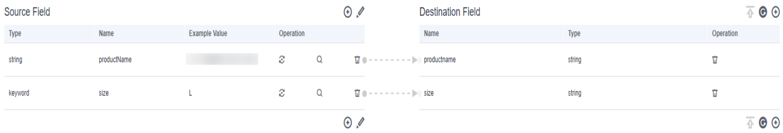
Table 2-10 Destination job configuration parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection created in **1.b**.
Resource Queue	Select a created DLI SQL queue.
Database	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table	Select the name of a table in the database. In this example, table **tablecss** created in **Create a database and table on DLI** is created.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set this parameter to **No**. If this parameter is set to **Yes**, data in the destination table will be cleared before the task is started.

3. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.
   - If the field mapping is incorrect, you can drag the fields to adjust the mapping.
   - If the type is automatically created at the migration destination, you need to configure the type and name of each field.

- CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 2-24 Field mapping



Source Field

Type	Name	Example Value	Operation
string	productName	[redacted]	
keyword	size	L	

Destination Field

Name	Type	Operation
productname	string	
size	string	

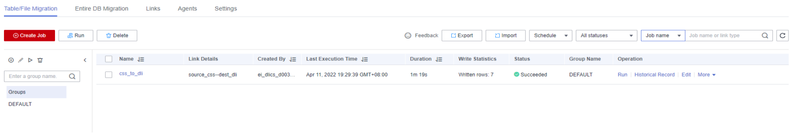
4. Click **Next** and set task parameters. Generally, retain the default values of all parameters.

   In this step, you can configure the following optional functions:

   - **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
   - **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
   - **Scheduled Execution**: Retain the default value **No**.
   - **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
   - **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No** so that dirty data is not recorded.

5. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 2-25 Job progress and execution result



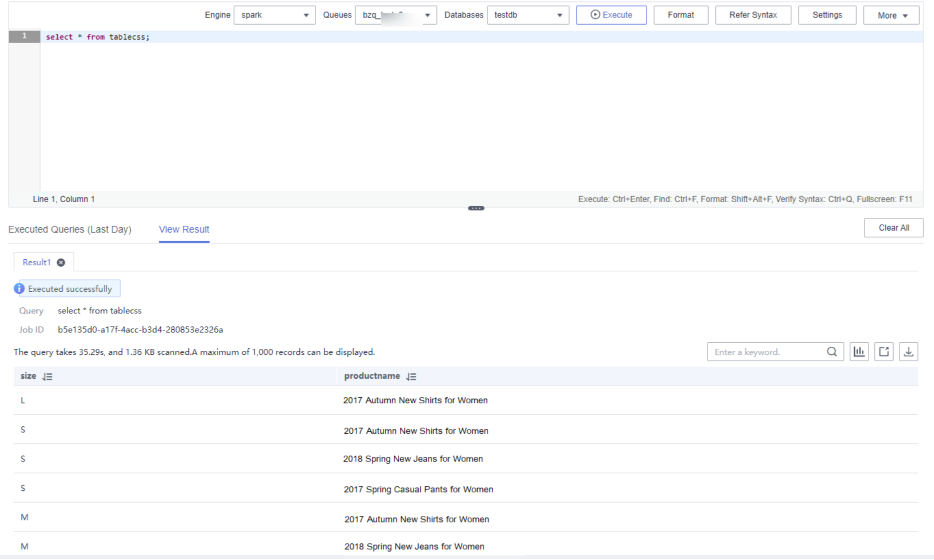
Name	Link Details	Created By	Last Execution Time	Duration	Write Statistics	Status	Group Name	Operation
css_to_dli	source_css--dest_dli	ei_dlics_d003...	Apr 11, 2022 19:29:39 GMT+08:00	1m 19s	Written rows: 7	Succeeded	DEFAULT	Run | Historical Record | Edit | More

Step 3: Query Results

After the migration job is complete, log in to the DLI management console and click **SQL Editor**. In the displayed page, set **Engine** to **spark**, **Queue** to the created SQL queue, and **Database** to the database created in **a**. Execute the following query statement and check whether the CSS table data has been migrated to the **tablecss** table:

```
select * from tablecss;
```

Figure 2-26 Querying migrated data



Engine spark Queues bzq_[redacted] Databases testdb

```
select * from tablecss;
```

Line 1, Column 1

Execute: Ctrl+Enter, Find: Ctrl+F, Format: Shift+Alt+F, Verify Syntax: Ctrl+Q, Fullscreen: F11

Executed successfully

Query select * from tablecss

Job ID b5e135d0-a17f-4acc-b3d4-280853e2326a

The query takes 35.29s, and 1.36 KB scanned.A maximum of 1,000 records can be displayed.

size	productname
L	2017 Autumn New Shirts for Women
S	2017 Autumn New Shirts for Women
S	2018 Spring New Jeans for Women
S	2017 Spring Casual Pants for Women
M	2017 Autumn New Shirts for Women
M	2018 Spring New Jeans for Women

2.5 Migrating Data from RDS to DLI

This section describes how to use the CDM data synchronization function to migrate data from an RDS DB instance to DLI. Data in other relational databases can also be bidirectionally synchronized between CDM and DLI.

Prerequisites

- You have created a DLI SQL queue. For details about how to create a DLI queue, see **Creating a Queue**.

  CAUTION

  When you create a queue, set its **Type** to **For SQL**.

- You have created an RDS for MySQL DB instance.
  - In this example, the RDS DB engine is MySQL.
  - In this example, the DB engine version is 5.7.
- You have created a CDM cluster.

NOTE

- If the destination data source is an on-premises database, you need the Internet or Direct Connect. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- If the data source is RDS or MRS on a cloud, the network must meet the following requirements:

  i. If the CDM cluster and the cloud service are in different regions, a public network or a dedicated connection is required for enabling communication between the CDM cluster and the cloud service. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules.

  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster are the same as those of the RDS for MySQL DB instance.

Step 1: Prepare Data

- Create databases and tables on the RDS for MySQL DB instance.
  a. Log in to the RDS console. On the displayed page, locate the target DB instance and choose **More** > **Log In** in the **Operation** column.
  b. On the displayed login page, enter the correct username and password and click **Log In**.
  c. On the **Databases** page, click **Create Database**. In the displayed dialog box, enter **testrdsdb** as the database name and retain default values of rest parameters. Then, click **OK**.
  d. In the **Operation** column of row where the created database locates, click **SQL Window** and enter the following statement to create a table:

```
CREATE TABLE tabletest (
    `id` VARCHAR(32) NOT NULL,
    `name` VARCHAR(32) NOT NULL,
    PRIMARY KEY (`id`)
)    ENGINE = InnoDB
    DEFAULT CHARACTER SET = utf8mb4;
```

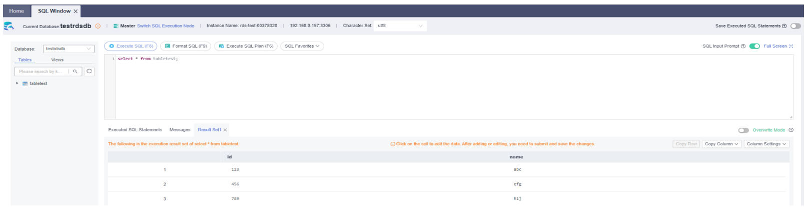
  e. Run the following statements to insert data to the created table:

```
insert into tabletest VALUES ('123','abc');
insert into tabletest VALUES ('456','efg');
insert into tabletest VALUES ('789','hij');
```

  f. Run the following statement to query table data:

```
select * from tabletest;
```

Figure 2-27 Querying table data



Current Database: testrdsdb | Master | Instance Name: rds-test-00378328 | 192.168.0.157:3306 | Character Set: utf8

Database: testrdsdb

```
select * from tabletest;
```

The following is the execution result set of select * from tabletest.

Click on the cell to edit the data. After adding or editing, you need to submit and save the changes.

	id	name
1	123	abc
2	456	efg
3	789	hij

- Create a database and table on DLI.

  a. Log in to the DLI management console and click **SQL Editor**. On the displayed page, set **Engine** to **spark** and **Queue** to the created SQL queue.

     Enter the following statement in the editing window to create a database, for example, the migrated DLI database **testdb**:

     ```
     create database testdb;
     ```

  b. In **SQL Editor**, select **testdb** for **Database** and run the following table creation statement to create a table in the database.

     ```
     create table tabletest(id string,name string);
     ```

Step 2: Migrate Data

1. Create a CDM connection to MRS Hive.

   a. Create a connection to the RDS database.

      i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

      ii. If this is your first time crating a connection to RDS for MySQL, upload the MySQL driver. Choose the **Links** tab and click **Driver Management**. The **Driver Management** page is displayed.

      iii. Download the MySQL driver to your local PC and decompress the driver package to obtain the JAR file.

      For example, download the **mysql-connector-java-5.1.48.zip** package and decompress it to obtain the driver file **mysql-connector-java-5.1.48.jar**.

      iv. Return to the **Driver Management** page. Locate the **MYSQL** driver and click **Upload** in the **Operation** column. In the **Import Driver File** dialog box, click **Select File** to upload the driver file obtained in **1.a.iii**.

      v. On the **Driver Management** page, click **Back** to return to the **Links** tab. Click **Create Link**, select **RDS for MySQL**, and click **Next**.

      vi. Configure the connection. The following table describes the required parameters.

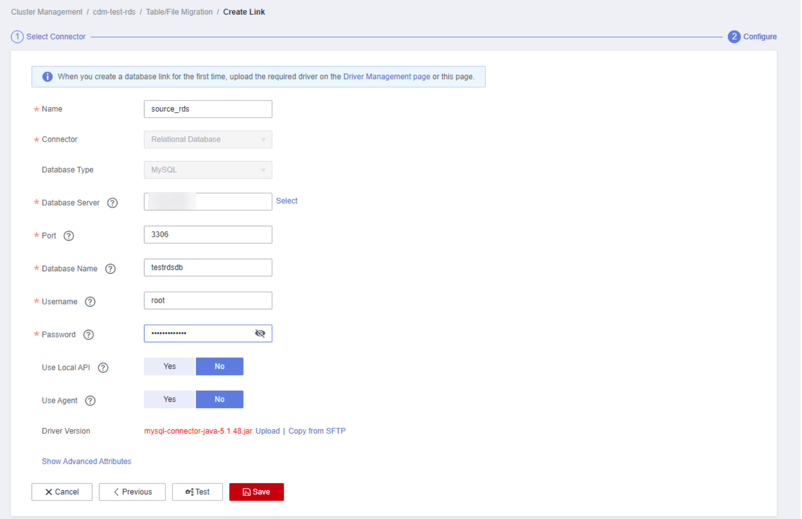
Table 2-11 Connection parameters

Parameter	Value
Name	Name of the RDS data source, for example, **source_rds**

Parameter	Value
Database Server	Click **Select** next to the text box and click the name of the created RDS DB instance. The database server address is automatically entered.
Port	Port number of the RDS DB instance. The value is automatically entered after you select the database server.
Database Name	Name of the RDS DB instance you want to migrate. The **testrdsdb** database created in **c** is used in this example.
Username	Username used for accessing the database. This account must have the permissions required to read and write data tables and metadata. In this example, the default user **root** for creating the RDS for MySQL DB instance is used.
Password	Password of the user.

For other parameters, retain the default values.. Click **Save** to complete the configuration.

Figure 2-28 Configuring the connection to the RDS for MySQL DB instance



Cluster Management / cdm-test-rds / Table/File Migration / Create Link

1 Select Connector — 2 Configure

When you create a database link for the first time, upload the required driver on the Driver Management page or this page.

* Name: source_rds

* Connector: Relational Database

Database Type: MySQL

* Database Server: [redacted] Select

* Port: 3306

* Database Name: testrdsdb

* Username: root

* Password: •••••••••••••

Use Local API: Yes [✓ No]

Use Agent: Yes [✓ No]

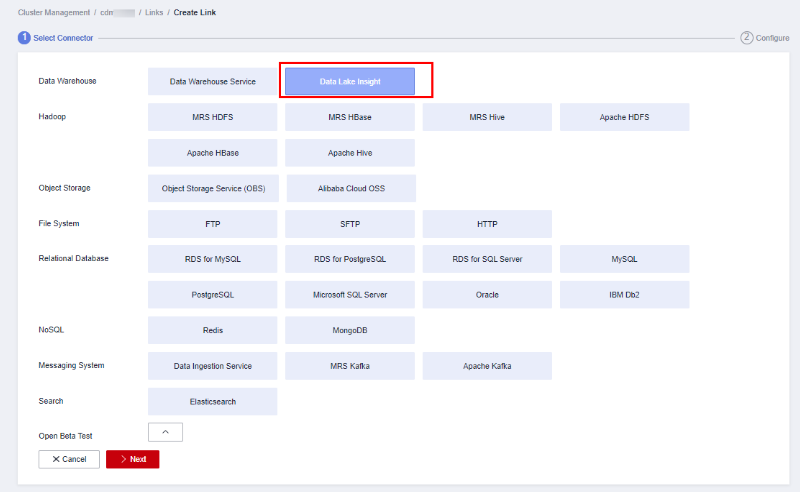
Driver Version: mysql-connector-java-5.1.48.jar Upload | Copy from SFTP

b. Create a connection to the DLI.

i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

Figure 2-29 Selecting the DLI connector



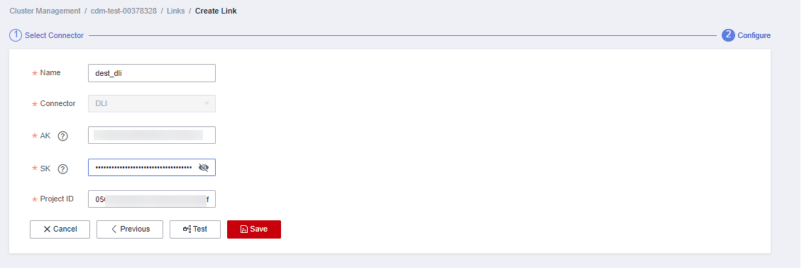
Cluster Management / cdm[redacted] / Links / Create Link

1 Select Connector — 2 Configure

Data Warehouse	Data Warehouse Service	Data Lake Insight		
Hadoop	MRS HDFS	MRS HBase	MRS Hive	Apache HDFS
	Apache HBase	Apache Hive		
Object Storage	Object Storage Service (OBS)	Alibaba Cloud OSS		
File System	FTP	SFTP	HTTP	
Relational Database	RDS for MySQL	RDS for PostgreSQL	RDS for SQL Server	MySQL
	PostgreSQL	Microsoft SQL Server	Oracle	IBM Db2
NoSQL	Redis	MongoDB		
Messaging System	Data Ingestion Service	MRS Kafka	Apache Kafka	
Search	Elasticsearch			
Open Beta Test	^			

i. Create a connection to link CDM to DLI.

Figure 2-30 Selecting the DLI connector



Cluster Management / cdm-test-00378328 / Links / Create Link

1 Select Connector — 2 Configure

* Name: dest_dli

* Connector: DLI

* AK ?: [redacted]

* SK ?: ••••••••••••••••••••••••••••••••••••

* Project ID: 05[redacted]f

After the configuration is complete, click **Save**.

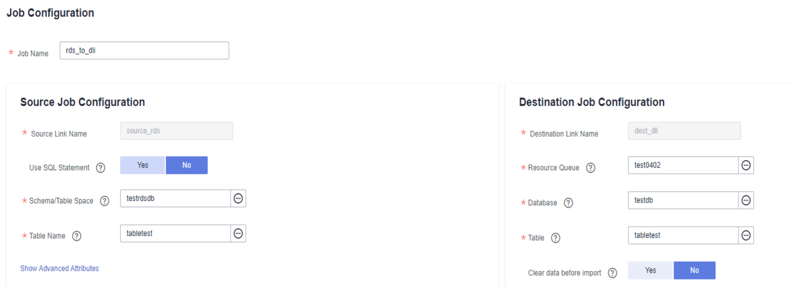
2. Create a CDM migration job.

   a. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

   b. On the **Job Management** page, choose the **Table/File Migration** tab and click **Create Job**.

   c. On the **Create Job** page, specify job information.

Figure 2-31 Configuring the migration job



Job Configuration

* Job Name: rds_to_dli

Source Job Configuration

* Source Link Name: source_rds

Use SQL Statement ?: Yes [✓ No]

* Schema/Table Space ?: testrdsdb

* Table Name ?: tabletest

Show Advanced Attributes

Destination Job Configuration

* Destination Link Name: dest_dli

* Resource Queue ?: test0402

* Database ?: testdb

* Table ?: tabletest

Clear data before import ?: Yes [✓ No]

i. **Job Name**: Name of the data migration job, for example, **rds_to_dli**

ii. Set parameters required for **Source Job Configuration**.

Table 2-12 Source job configuration parameters

Parameter	Value
Source Link Name	Select the name of the data source created in **1.a**.
Use SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement. In this example, set this parameter to **No**.
Schema/Table Space	Select the name of the RDS for MySQL DB instance you want to migrate to DLI. For example, the **testrdsdb** database.
Table Name	Name of the table you want to migrate. In this example, use **tabletest** created in **d**.

For details about parameter settings, see **From a Common Relational Database**.

iii. Set parameters required for **Destination Job Configuration**.

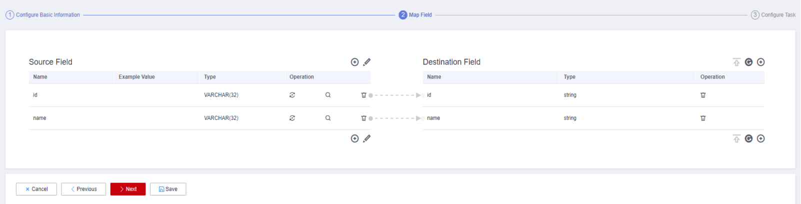
Table 2-13 Destination job configuration parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection.
Resource Queue	Select a created DLI SQL queue.
Database	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table	Select the name of a table in the database. In this example, table **tabletest** created in **Create a database and table on DLI** is created.
Clear data before import	Whether to clear data in the destination table before data import. In this example, set this parameter to **No**. If this parameter is set to **Yes**, data in the destination table will be cleared before the task is started.

iv. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.

- If the field mapping is incorrect, you can drag the fields to adjust the mapping.
- If the type is automatically created at the migration destination, you need to configure the type and name of each field.
- CDM allows for field conversion during migration. For details, see **Field Conversion**.

Figure 2-32 Field mapping



Source Field

Name	Example Value	Type	Operation
id		VARCHAR(32)	
name		VARCHAR(32)	

Destination Field

Name	Type	Operation
id	string	
name	string	

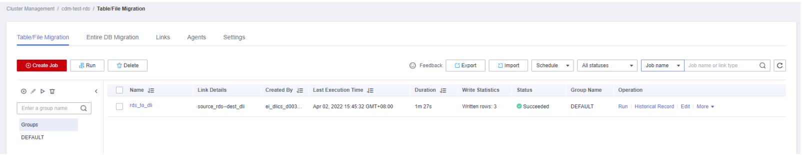
v. Click **Next** and set task parameters. Generally, retain the default values of all parameters.

In this step, you can configure the following optional functions:

- **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
- **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
- **Scheduled Execution**: Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No** so that dirty data is not recorded.

vi. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 2-33 Job progress and execution result



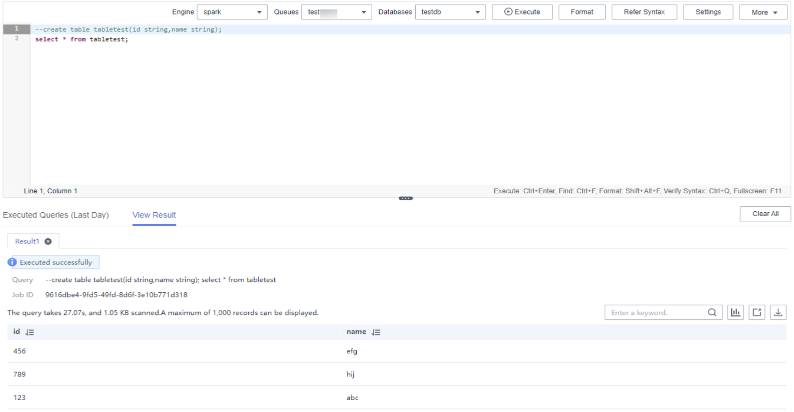
Name	Link Details	Created By	Last Execution Time	Duration	Write Statistics	Status	Group Name	Operation
rds_to_dli	source_rds--dest_dli	ei_dlics_d003...	Apr 02, 2022 15:45:32 GMT+08:00	1m 27s	Written rows: 3	Succeeded	DEFAULT	Run | Historical Record | Edit | More

Step 3: Query Results

After the migration job is complete, log in to the DLI management console and click **SQL Editor**. In the displayed page, set **Engine** to **spark**, **Queue** to the created SQL queue, and **Database** to the database created in **Create a database and table on DLI**. Execute the following query statement and check whether the table data has been migrated to the **tabletest** table:

```
select * from tabletest;
```

Figure 2-34 Querying data in the table



```
--create table tabletest(id string,name string);
select * from tabletest;
```

Executed successfully

Query --create table tabletest(id string,name string); select * from tabletest

Job ID 9616dbe4-9fd5-49fd-8d6f-3e10b771d318

The query takes 27.07s, and 1.05 KB scanned.A maximum of 1,000 records can be displayed.

id	name
456	efg
789	hij
123	abc

2.6 Migrating Data from GaussDB(DWS) to DLI

This section describes how to use the CDM data synchronization function to migrate data from GaussDB(DWS) to DLI.

Prerequisites

- You have created a DLI SQL queue.

⚠ CAUTION

When you create a queue, set its **Type** to **For SQL**.

- You have created a GaussDB(DWS) cluster.
- You have created a CDM cluster. For details about how to create a CDM cluster, see **Creating a CDM Cluster**.

NOTE

- If the destination data source is an on-premises database, you need the Internet or Direct Connect. When using the Internet, ensure that an EIP has been bound to the CDM cluster, the security group of CDM allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
- If the data source is GaussDB(DWS) or MRS on a cloud, the network must meet the following requirements:

  i. If the CDM cluster and the cloud service are in different regions, a public network or a dedicated connection is required for enabling communication between the CDM cluster and the cloud service. If the Internet is used for communication, ensure that an EIP has been bound to the CDM cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.

  ii. If the CDM cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If the CDM cluster and the cloud service are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules.

  iii. The cloud service instance and the CDM cluster belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.

In this example, the VPC, subnet, and security group of the CDM cluster are the same as those of the GaussDB(DWS) cluster.

Step 1: Prepare Data

- Create a database and table in the GaussDB(DWS) cluster.

  a. Connect to the existing GaussDB(DWS) cluster by referring to **Using the gsql CLI Client to Connect to a Cluster**.

  b. Connect to the default database **gaussdb** of a GaussDB(DWS) cluster.

```
gsql -d gaussdb -h Connection address of the GaussDB(DWS) cluster -U dbadmin -p 8000 -W
password -r
```

  - **gaussdb**: Default database of the GaussDB(DWS) cluster
  - **Connection address of the DWS cluster**: If a public network address is used for connection, set this parameter to **Public Network Address** or **Public Network Access Domain Name**. If a private network address is used for connection, set this parameter to **Private Network Address** or **Private Network Access Domain Name**. If an ELB is used for connection, set this parameter to the ELB address.
  - **dbadmin**: Default administrator username used during cluster creation
  - **-W**: Default password of the administrator

  c. Run the following command to create the **testdwsdb** database:

```
CREATE DATABASE testdwsdb;
```

  d. Run the following command to exit the **gaussdb** database and connect to **testdwsdb**:

```
\q
gsql -d testdwsdb -h Connection address of the GaussDB(DWS) cluster -U dbadmin -p 8000 -W
password -r
```

  e. Run the following commands to create a table and import data to the table.

  Run the following command to create a table:

```
CREATE TABLE table1(id int, a char(6), b varchar(6),c varchar(6)) ;
```

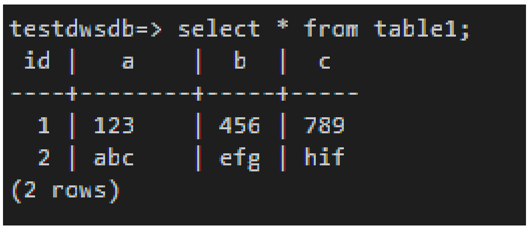
  Run the following statements to insert data into the table:

```
INSERT INTO table1 VALUES(1,'123','456','789');
INSERT INTO table1 VALUES(2,'abc','efg','hif');
```

  f. Query the table data to verify that the data is inserted.

```
select * from table1;
```

Figure 2-35 Querying data in the table



```
testdwsdb=> select * from table1;
 id |   a    |  b  |  c
----+--------+-----+-----
  1 | 123    | 456 | 789
  2 | abc    | efg | hif
(2 rows)
```

- Create a database and table on DLI.

a. Log in to the DLI management console and click **SQL Editor**. On the displayed page, set **Engine** to **spark** and **Queue** to the created SQL queue.

Enter the following statement in the editing window to create a database, for example, the migrated DLI database **testdb**:

```
create database testdb;
```

b. In **SQL Editor**, select **testdb** for **Database** and run the following table creation statement to create a table in the database:

```
create table tabletest(id INT, name1 string, name2 string, name3 string);
```

Step 2: Migrate Data

1. Create a CDM connection to MRS Hive.

a. Create a connection to the GaussDB(DWS) database.

i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

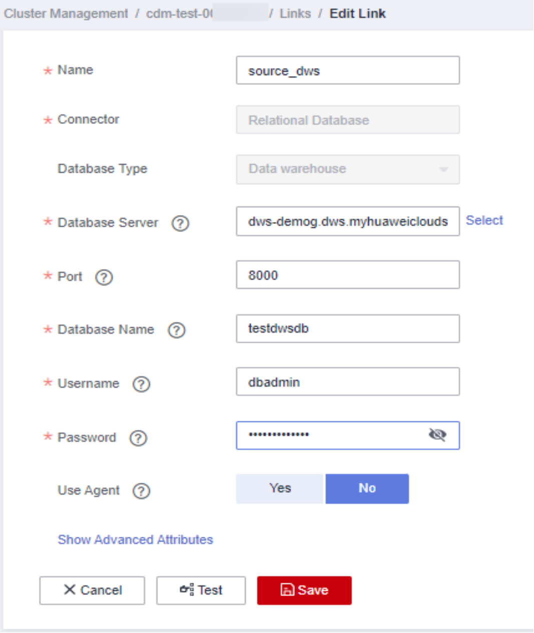
ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Warehouse Service** and click **Next**.

iii. Configure the connection. The following table describes the required parameters.

Table 2-14 GaussDB(DWS) data source configuration

Parameter	Value
Name	Name of the GaussDB(DWS) data source, for example, **source_dws**.
Database Server	Click **Select** next to the text box to select the name of the created GaussDB(DWS) cluster.
Port	Port number of the GaussDB(DWS) database. The default value is **8000**.
Database Name	Name of the GaussDB(DWS) database you want to migrate The **testdwsdb** database created in **Create a database and table in the GaussDB(DWS) cluster** is used in this example.
Username	Username used for accessing the database. This account must have the permissions required to read and write data tables and metadata. In this example, the default administrator **dbadmin** specified when you create the GaussDB(DWS) database is used.
Password	Password of the GaussDB(DWS) database user.

Figure 2-36 Configuring the GaussDB(DWS) connection



Cluster Management / cdm-test-0[redacted] / Links / **Edit Link**

* Name: source_dws

* Connector: Relational Database

Database Type: Data warehouse

* Database Server: dws-demog.dws.myhuaweiclouds

* Port: 8000

* Database Name: testdwsdb

* Username: dbadmin

* Password: ••••••••••••••

Use Agent: Yes [✓ No]

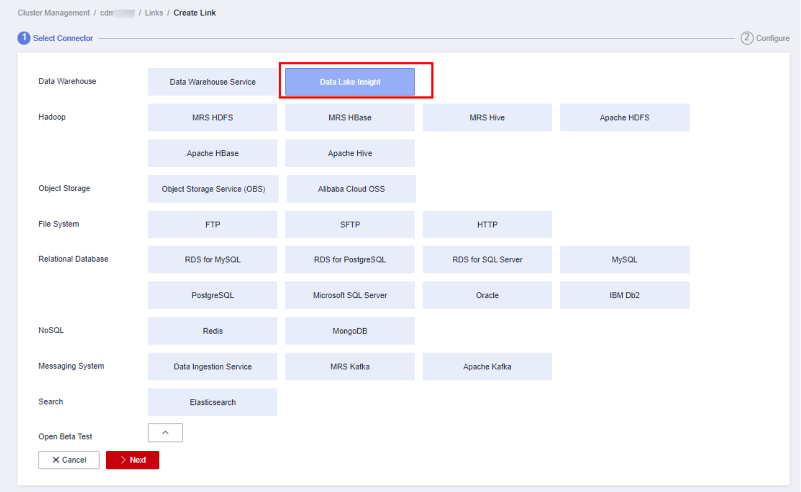
For other parameters, retain the default values. Click **Save** to complete the configuration.

b. Create a connection to the DLI.

i. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

ii. On the **Job Management** page, click the **Links** tab, and click **Create Link**. On the displayed page, select **Data Lake Insight** and click **Next**.

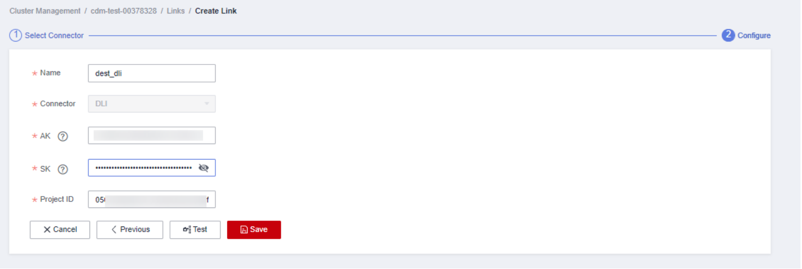
Figure 2-37 Selecting the DLI connector



Cluster Management / cdm[redacted] / Links / **Create Link**

i. Create a connection to link CDM to DLI.

Figure 2-38 Selecting the DLI connector



Cluster Management / cdm-test-00378328 / Links / **Create Link**

- * Name: dest_dli
- * Connector: DLI
- * AK: [redacted]
- * SK: ••••••••••••••••••••••••••••••••••
- * Project ID: 05[redacted]f

After the configuration is complete, click **Save**.

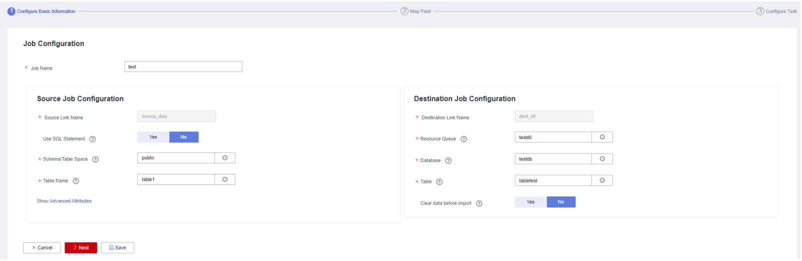
2. Create a CDM migration job.

   a. Log in to the CDM console, choose **Cluster Management**. On the displayed page, locate the created CDM cluster, and click **Job Management** in the **Operation** column.

   b. On the **Job Management** page, choose the **Table/File Migration** tab and click **Create Job**.

   c. On the **Create Job** page, specify job information.

Figure 2-39 Configuring the migration job



Job Configuration

- * Job Name: test

Source Job Configuration

- * Source Link Name: source_dws
- Use SQL Statement: Yes [✓ No]
- * Schema/Table Space: public
- * Table Name: table1

Destination Job Configuration

- * Destination Link Name: dest_dli
- * Resource Queue: testdli
- * Database: testdb
- * Table: tabletest
- Clear data before import: Yes [✓ No]

i. **Job Name**: Name of the data migration job, for example, **test**

ii. Set parameters required for **Source Job Configuration**.

Table 2-15 Source job configuration parameters

Parameter	Value
Source Link Name	Select the name of the data source created in 1.a.
Use SQL Statement	When **Use SQL Statement** is set to **Yes**, enter an SQL statement here. CDM exports data based on the SQL statement. In this example, set this parameter to **No**.

Parameter	Value
Schema/Table Space	Name of the schema or tablespace from which data will be extracted. This parameter is displayed when **Use SQL Statement** is set to **No**. Click the icon next to the text box to go to the page for selecting a schema or directly enter a schema or tablespace. In this example, no schema is created in **Create a database and table in the GaussDB(DWS) cluster**. In this case, set this parameter to the default value **public**. If the desired schema or tablespace is not displayed, check whether the login account has the permissions required to query metadata. **NOTE** The parameter value can contain wildcard characters (*), which is used to export all databases whose names start with a certain prefix or end with a certain suffix. For example: **SCHEMA*** indicates that all databases whose names starting with **SCHEMA** are exported. ***SCHEMA** indicates that all databases whose names ending with **SCHEMA** are exported. ***SCHEMA*** indicates that all databases whose names containing **SCHEMA** are exported.
Table Name	Name of the table you want to migrate. In this example, **table1** created in **Create a database and table in the GaussDB(DWS) cluster** is used.

iii. Set parameters required for **Destination Job Configuration**.

Table 2-16 Destination job configuration parameters

Parameter	Value
Destination Link Name	Select the DLI data source connection.
Resource Queue	Select a created DLI SQL queue.
Database	Select a created DLI database. In this example, database **testdb** created in **Create a database and table on DLI** is selected.
Table	Select the name of a table in the database. In this example, table **tabletest** created in **Create a database and table on DLI** is created.

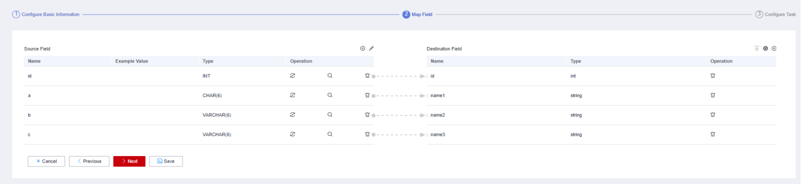
Parameter	Value
Clear data before import	Whether to clear data in the destination table before data import. In this example, set this parameter to **No**. If this parameter is set to **Yes**, data in the destination table will be cleared before the task is started.

For details about parameter settings, see **To DLI**.

iv. Click **Next**. The **Map Field** page is displayed. CDM automatically matches the source and destination fields.

- If the field mapping is incorrect, you can drag the fields to adjust the mapping.
- If the type is automatically created at the migration destination, you need to configure the type and name of each field.
- CDM allows for field conversion during migration.

Figure 2-40 Field mapping



Source Field

Name	Example Value	Type	Operation
id		INT	
a		CHAR(6)	
b		VARCHAR(6)	
c		VARCHAR(6)	

Destination Field

Name	Type	Operation
id	int	
name1	string	
name2	string	
name3	string	

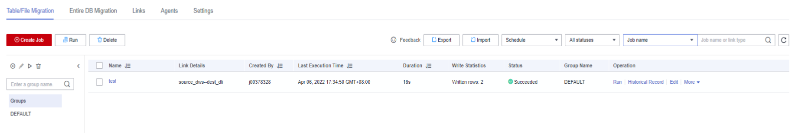
v. Click **Next** and set task parameters. Generally, retain the default values of all parameters.

In this step, you can configure the following optional functions:

- **Retry Upon Failure**: If the job fails to be executed, you can determine whether to automatically retry. Retain the default value **Never**.
- **Group**: Select the group to which the job belongs. The default group is **DEFAULT**. On the **Job Management** page, jobs can be displayed, started, or exported by group.
- **Scheduled Execution**: Retain the default value **No**.
- **Concurrent Extractors**: Enter the number of extractors to be concurrently executed. Retain the default value **1**.
- **Write Dirty Data**: Specify this parameter if data that fails to be processed or filtered out during job execution needs to be written to OBS. Before writing dirty data, create an OBS link. You can view the data on OBS later. Retain the default value **No** so that dirty data is not recorded.

vi. Click **Save and Run**. On the **Job Management** page, you can view the job execution progress and result.

Figure 2-41 Job progress and execution result



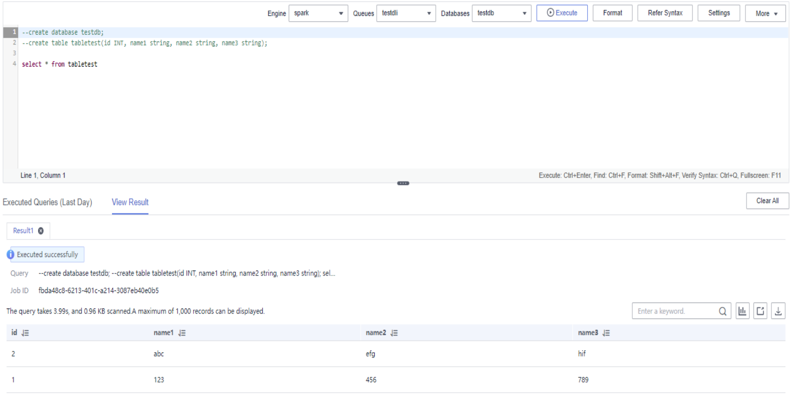
Name	Link Details	Created By	Last Execution Time	Duration	Write Statistics	Status	Group Name	Operation
test	source_dws--dest_dli	j00378328	Apr 06, 2022 17:34:50 GMT+08:00	16s	Written rows: 2	Succeeded	DEFAULT	Run | Historical Record | Edit | More

Step 3: Query Results

After the migration job is complete, log in to the DLI management console and click **SQL Editor**. In the displayed page, set **Engine** to **spark**, **Queue** to the created SQL queue, and **Database** to the database created in **Create a database and table on DLI**. Execute the following query statement and check whether the table data has been migrated to the **tabletest** table:

```
select * from tabletest;
```

Figure 2-42 Querying data in the table



```
--create database testdb;
--create table tabletest(id INT, name1 string, name2 string, name3 string);

select * from tabletest
```

Line 1, Column 1

Execute: Ctrl+Enter, Find: Ctrl+F, Format: Shift+Alt+F, Verify Syntax: Ctrl+Q, Fullscreen: F11

Executed successfully

Query --create database testdb; --create table tabletest(id INT, name1 string, name2 string, name3 string); sel...

Job ID fbda48c8-6213-401c-a214-3087eb40e0b5

The query takes 3.99s, and 0.96 KB scanned.A maximum of 1,000 records can be displayed.

id	name1	name2	name3
2	abc	efg	hif
1	123	456	789

3 Data Analysis

3.1 Analyzing Driving Behavior Data

Application Scenarios

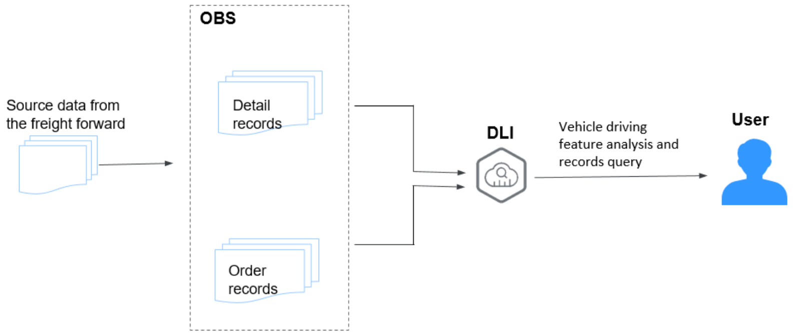
Cloud computing and big data provide companies with data analysis and mining capabilities required in the Internet of Vehicle (IoV) field, helping companies or department of motor vehicles manage and analyze vehicle and driving behavior data quickly and scientifically.

Solution Architecture

DLI can query the records of vehicle driving features based on the detail records and freight order data regularly reported by the freight forwarder.

Data Types describes the data types used by DLI to record the data.

Figure 3-1 Solution Overview


OBS
Source data from the freight forward
Detail records
Order records
DLI
Vehicle driving feature analysis and records query
User

Process

To use DLI to analyze driving behavior data, perform the following steps:

Step 1: Uploading Data. Upload the data to OBS.

Step 2: Analyzing Data. Use DLI to query the data.

Example Code

Download the **data package** for sample data and detailed SQL statements.

Solution Advantages

- Free of data migration: DLI can interconnect with multiple data sources. You only need to create SQL tables and map data sources.
- Easy to use: You can use standard SQL statements to compile metric analysis logic without paying attention to the complex distributed computing platform.
- Pay-per-use: Log analysis is scheduled periodically based on time-critical requirements. There is a long idle period between every two scheduling operations. DLI uses the pay-per-use billing mode, which effectively reduces your costs.

Resource Planning and Costs

Table 3-1 Resource planning and costs

Resource	Description	Cost
OBS	You need to create an OBS bucket and upload data to OBS for data analysis using DLI.	You will be charged for using the following OBS resources: ● **Storage Fee** for storing static website files in OBS. ● **Request Fee** for accessing static website files stored in OBS. ● **Traffic Fee** for using a custom domain name to access OBS over the public network. The actual fee depends on the size of the stored file, the number of user access requests, and the traffic volume. Estimate the fee based on your service requirements.
DLI	Before creating a SQL job, you need to purchase a queue. When using queue resources, you are billed based on the CUH of the queue.	For example, if you purchase a pay-per-use queue, you will be billed based on the number of CUHs used by the queue. Usage is billed by the hour. For example, 58 minutes of usage will be rounded to the hour. CUH pay-per-use billing = Unit price x Number of CUs x Number of hours.

Data Types

- Detail records

  Detail records include the regularly reported location records and data of alarms triggered by abnormal driving behavior.

Table 3-2 Detail records

Field	Data Type	Description
driverID	string	Driver ID
carNumber	string	License plate number
latitude	double	Latitude
longitude	double	Longitude
speed	int	Speed
direction	int	Direction
siteName	string	Site name
time	timestamp	Report time of the records
isRapidlySpeedup	int	Whether the vehicle rapidly speeds up. **1** indicates that the vehicle suddenly speeds up, and **0** indicates that the vehicle does not.
isRapidlySlowdown	int	Whether the vehicle suddenly slows down.
isNeutralSlide	int	Whether the vehicle is coasting.
isNeutralSlideFinished	int	Whether vehicle coasting has stopped.
neutralSlideTime	bigint	Time length of vehicle coasting.
isOverspeed	int	Whether the vehicle is speeding.
isOverspeedFinished	int	Whether the vehicle stops speeding.
overspeedTime	bigint	Duration of the vehicle speeding
isFatigueDriving	int	Whether fatigue driving occurs.

Field	Data Type	Description
isHthrottleStop	int	Whether the driver revs the engine in neutral.
isOilLeak	int	Abnormal oil consumption

- Order data

  Order data refers to the records of freight orders.

Table 3-3 Order data

Field	Data Type	Description
orderNumber	string	Order ID
driverID	string	Driver ID
carNumber	string	License plate number
customerID	string	Customer ID
sourceCity	string	Departure
targetCity	string	Destination
expectArriveTime	timestamp	Expected delivery time
time	timestamp	Time when a record is generated.
action	string	Event type, including creating an order, dispatching goods, delivering packages, and signing orders.

Step 1: Uploading Data

Upload the data to OBS for data analysis using DLI.

1. Download OBS Browser+. For details about the download address, see **Object Storage Service Tool Guide**.
2. Install OBS Browser+. For details about the installation procedure, see **Object Storage Service Tool Guide**.
3. Log in to OBS Browser+. OBS Browser+ supports two login modes: AK login (using access keys) or authorization code login. For details about the login procedure, see **Object Storage Service Tool Guide**.
4. Upload data using the OBS browser+.

   Start the OBS Browser+, click **Create Bucket** on the homepage. Select a region and enter a bucket name (for example, **DLI-demo**). After the bucket is created, return to the bucket list and click **DLI-demo**. OBS Browser+ supports upload by dragging. You can drag one or more files or folders from a local path to the object list of a bucket or a parallel file system on OBS Browser+. You can even drag a file or folder directly to a specified folder on OBS Browser+.

Obtain the test data by downloading the **Best_Practice_01.zip** file and decompressing it. Perform the following operations:

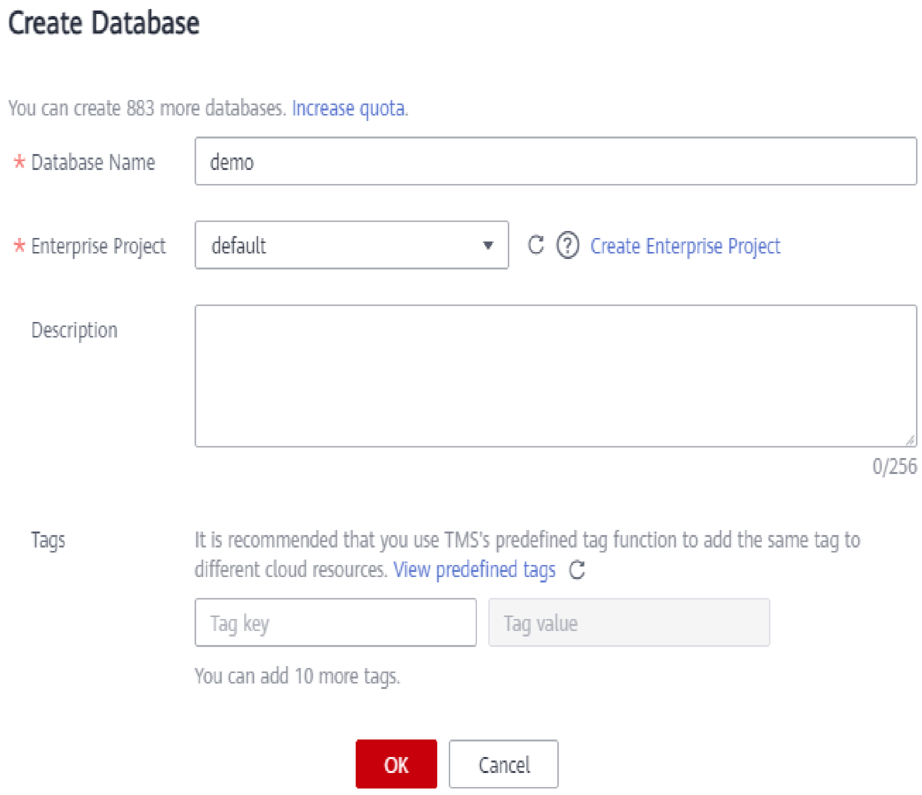
- Detail records: Upload the **detail-records** folder in the **Data** directory to the root directory of the OBS bucket.
- Order data: Upload the **order-records** folder in the **Data** directory to the root directory of the OBS bucket.

Step 2: Analyzing Data

Use DLI to query the data for analysis.

1. Creating a Database and a Table
   a. On the homepage of the management console, choose **Service List** > **Analytics** > **Data Lake Insight**.
   b. On the DLI console, click **SQL Editor**.
   c. In the left pane of the SQL Editor, select the **Databases** tab and click ⊕ to create the **demo** database.

Figure 3-2 Creating a database



Create Database

You can create 883 more databases. Increase quota.

* Database Name: demo

* Enterprise Project: default

Description:

0/256

Tags: It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

Tag key | Tag value

You can add 10 more tags.

NOTE

Database Name cannot be set to **default** because **default** is the built-in database.

d. Choose the **demo** database, and enter the following SQL statement in the editing box:

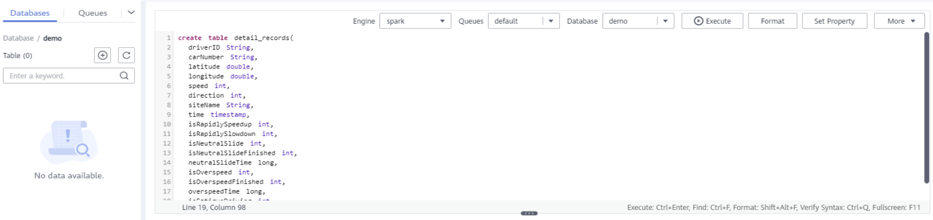
```
create table detail_records(
  driverID String,
  carNumber String,
  latitude double,
  longitude double,
  speed int,
  direction int,
  siteName String,
  time timestamp,
  isRapidlySpeedup int,
  isRapidlySlowdown int,
  isNeutralSlide int,
  isNeutralSlideFinished int,
  neutralSlideTime long,
  isOverspeed int,
  isOverspeedFinished int,
  overspeedTime long,
  isFatigueDriving int,
  isHthrottleStop int,
  isOilLeak int) USING CSV OPTIONS (PATH 'obs://dli-demo/detail-records/');
```

NOTE

Replace the file path in the preceding statement with the actual OBS path where the detail records are stored.

e. Click **Execute** to create the **detail_records** table. See **Figure 3-3**.

Figure 3-3 Creating the **detail_records** table



```
create table detail_records(
  driverID String,
  carNumber String,
  latitude double,
  longitude double,
  speed int,
  direction int,
  siteName String,
  time timestamp,
  isRapidlySpeedup int,
  isRapidlySlowdown int,
  isNeutralSlide int,
  isNeutralSlideFinished int,
  neutralSlideTime long,
  isOverspeed int,
  isOverspeedFinished int,
  overspeedTime long,
```

f. Run the following SQL statements to create the **event_records** table in the **demo** database. The operation is similar to **1.d** and **1.e**.

```
create table event_records(
  driverID String,
  carNumber String,
  latitude double,
  longitude double,
  speed int,
  direction int,
  siteName String,
  time timestamp,
  isRapidlySpeedup int,
  isRapidlySlowdown int,
  isNeutralSlide int,
  isNeutralSlideFinished int,
  neutralSlideTime long,
  isOverspeed int,
  isOverspeedFinished int,
  overspeedTime long,
  isFatigueDriving int,
  isHthrottleStop int,
  isOilLeak int)
```

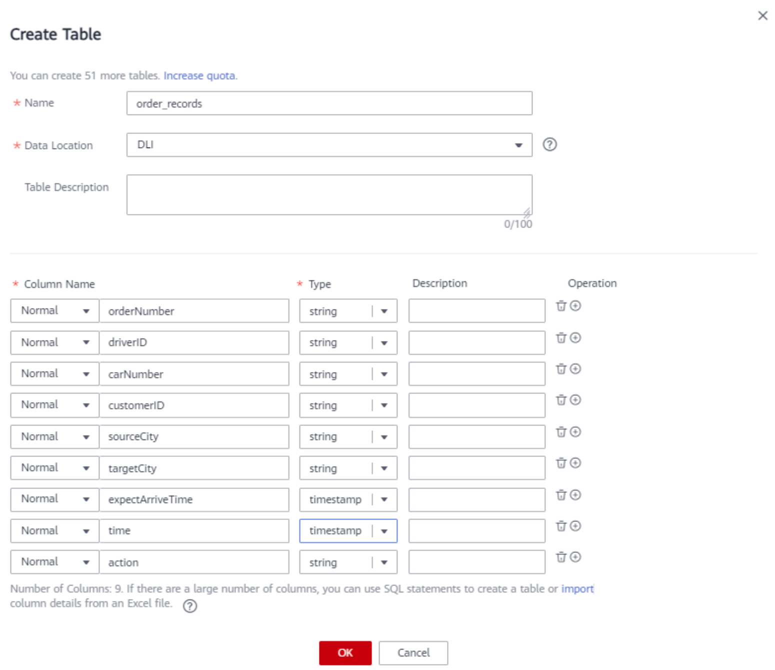
g. Run the following SQL statements to extract the alarm and event data from the detail records and insert it into the **event_records** table.

```
insert into table event_records
(select *
from detail_records
where isRapidlySpeedup > 0
OR isRapidlySlowdown > 0
OR isNeutralSlide > 0
OR isNeutralSlideFinished > 0
OR isOverspeed > 0
OR isOverspeedFinished > 0
OR isFatigueDriving > 0
OR isHthrottleStop > 0
OR isOilLeak > 0)
```

h. Use another method to create the **order_records** table.

On the left of the SQL job editor, click the **Databases** tab and click the demo database. Click the plus icon (+) on the right of **Table** to create a table, and set **Data Location** to **DLI**. Set the column types according to **Order data**.

Figure 3-4 Creating the **order_records** table



Create Table

You can create 51 more tables. Increase quota.

* Name: order_records

* Data Location: DLI

Table Description: 0/100

* Column Name		* Type	Description	Operation
Normal	orderNumber	string		
Normal	driverID	string		
Normal	carNumber	string		
Normal	customerID	string		
Normal	sourceCity	string		
Normal	targetCity	string		
Normal	expectArriveTime	timestamp		
Normal	time	timestamp		
Normal	action	string		

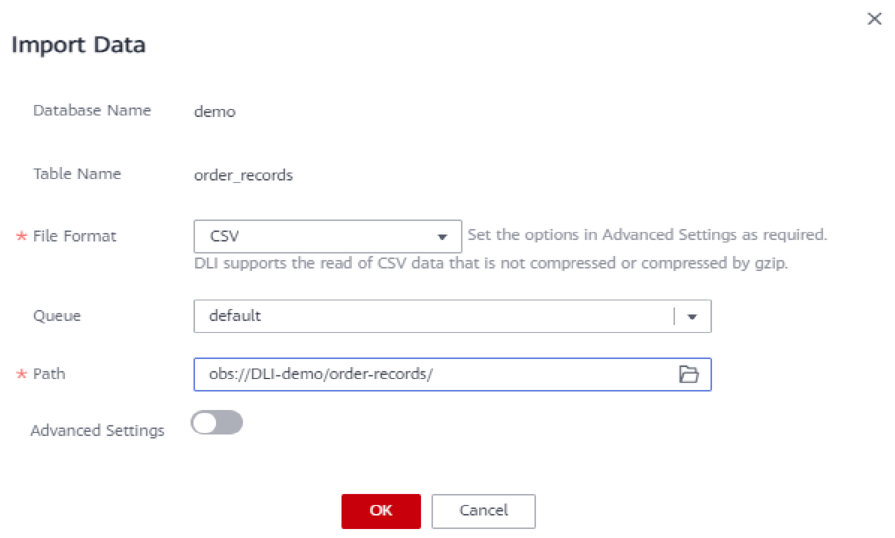
Number of Columns: 9. If there are a large number of columns, you can use SQL statements to create a table or import column details from an Excel file.

i. Import the OBS data to the **order_records** table. Choose **Data Management** > **Databases and Tables**. Click the demo database to go to the table management page. In the **Operation** column of the **order_records** table, choose **More** > **Import**. Set **File Format** to **CSV**, the data storage path to **obs://DLI-demo/order-records/**, and retain default values for the rest parameters. Click **OK**.

NOTE

The default timestamp format is **yyyy-MM-dd HH:mm:ss**. To use other formats, select **Advanced Settings** and enter the desired timestamp format (not modified in this example).

Figure 3-5 Importing table data



Import Data

Database Name demo

Table Name order_records

* File Format CSV Set the options in Advanced Settings as required.
DLI supports the read of CSV data that is not compressed or compressed by gzip.

Queue default

* Path obs://DLI-demo/order-records/

Advanced Settings

2. Querying Data
   a. Run the following SQL statements to query the alarm events of all drivers in a certain time period.

      NOTE

      You can save the frequently-used query statements as a template by clicking **More** > **Save as Template** in the upper right corner of the editing window. The template is available for future use or can be modified in the SQL editor again.

      Choose **Job Templates** > **SQL Templates** and click the **Custom Templates** tab. In the **Operation** column of the target template, click **Execute** to switch to the SQL editor. You can modify it as needed.

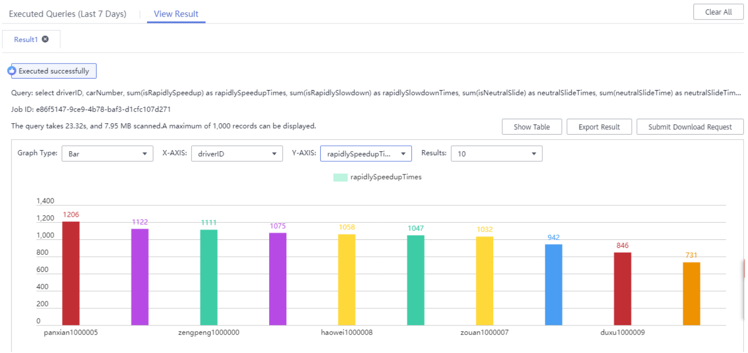
```
select
  driverID,
  carNumber,
  sum(isRapidlySpeedup) as rapidlySpeedupTimes,
  sum(isRapidlySlowdown) as rapidlySlowdownTimes,
  sum(isNeutralSlide) as neutralSlideTimes,
  sum(neutralSlideTime) as neutralSlideTimeTotal,
  sum(isOverspeed) as overspeedTimes,
  sum(overspeedTime) as overspeedTimeTotal,
  sum(isFatigueDriving) as fatigueDrivingTimes,
  sum(isHthrottleStop) as hthrottleStopTimes,
  sum(isOilLeak) as oilLeakTimes
from
  event_records
where
  time >= "2017-01-01 00:00:00"
  and time <= "2017-02-01 00:00:00"
group by
  driverID,
  carNumber
order by
  rapidlySpeedupTimes desc,
  rapidlySlowdownTimes desc,
  neutralSlideTimes desc,
  neutralSlideTimeTotal desc,
  overspeedTimes desc,
  overspeedTimeTotal desc,
  fatigueDrivingTimes desc,
  hthrottleStopTimes desc,
  oilLeakTimes desc
```

In the query result, click 📊 to view graphical results.

- Set **Graph Type** to the bar chart.
- Set **X-AXIS** to **driverID**.
- Set **Y-AXIS** to **rapidlySpeedupTimes**.
- Set **Results** to **10**.

The command output is as follows:

Figure 3-6 Rapid acceleration



Executed successfully

Query: select driverID, carNumber, sum(isRapidlySpeedup) as rapidlySpeedupTimes, sum(isRapidlySlowdown) as rapidlySlowdownTimes, sum(isNeutralSlide) as neutralSlideTimes, sum(neutralSlideTime) as neutralSlideTim...

Job ID: e86f5147-9ce9-4b78-baf3-d1cfc107d271

The query takes 23.32s, and 7.95 MB scanned.A maximum of 1,000 records can be displayed.

Graph Type: Bar X-AXIS: driverID Y-AXIS: rapidlySpeedupTi... Results: 10

b. Run the following SQL statement to query the detailed record of a driver in a certain time period.

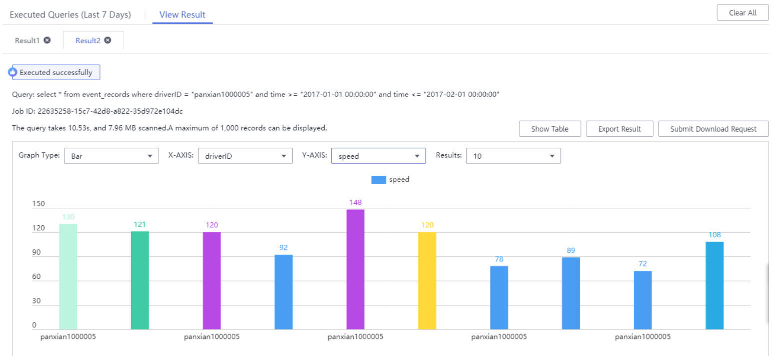
```
select
  *
from
  event_records
where
  driverID = "panxian1000005"
  and time >= "2017-01-01 00:00:00"
  and time <= "2017-02-01 00:00:00"
```

In the query result, click 📊 to view graphical results.

- Set **Graph Type** to the bar chart.
- Set **X-AXIS** to **driverID**.
- Set **Y-AXIS** to **speed**.
- Set **Results** to **10**.

The command output is as follows:

Figure 3-7 Speeding record



Query: select * from event_records where driverID = "panxian1000005" and time >= "2017-01-01 00:00:00" and time <= "2017-02-01 00:00:00"

Job ID: 22635258-15c7-42d8-a822-35d972e104dc

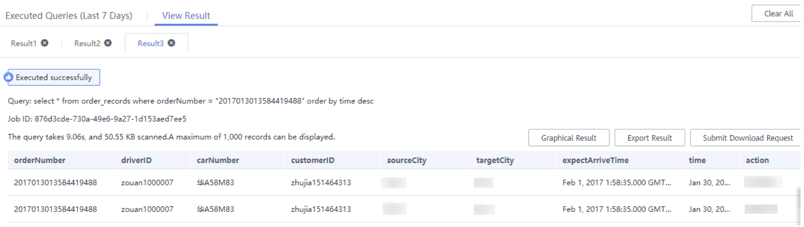
The query takes 10.53s, and 7.96 MB scanned.A maximum of 1,000 records can be displayed.

Graph Type: Bar
X-AXIS: driverID
Y-AXIS: speed
Results: 10

c. Run the following SQL statement to query the order information.

```
select
  *
from
  order_records
where
  orderNumber = "2017013013584419488"
order by
  time desc
```

Figure 3-8 Order information



Query: select * from order_records where orderNumber = "2017013013584419488" order by time desc

Job ID: 876d3cde-730a-49e6-9a27-1d153aed7ee5

The query takes 9.06s, and 50.55 KB scanned.A maximum of 1,000 records can be displayed.

orderNumber	driverID	carNumber	customerID	sourceCity	targetCity	expectArriveTime	time	action
2017013013584419488	zouan1000007	华A58M83	zhujia151464313	[redacted]	[redacted]	Feb 1, 2017 1:58:35.000 GMT...	Jan 30, 20...	[redacted]
2017013013584419488	zouan1000007	华A58M83	zhujia151464313	[redacted]	[redacted]	Feb 1, 2017 1:58:35.000 GMT...	Jan 30, 20...	[redacted]

d. Run the following SQL statement to query a vehicle's driving feature according to the driver ID and time of departure.

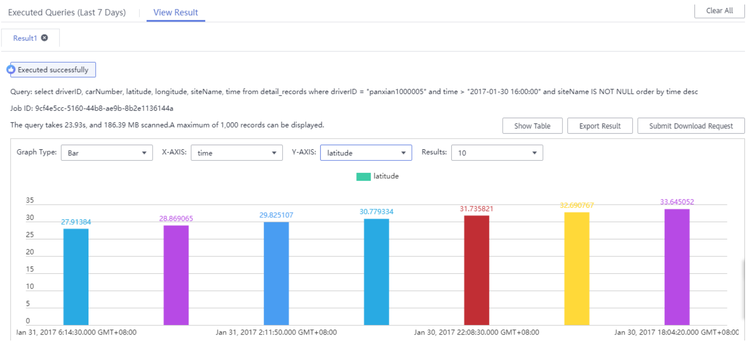
```
select
  driverID,
  carNumber,
  latitude,
  longitude,
  siteName,
  time
from
  detail_records
where
  driverID = "panxian1000005"
  and time > "2017-01-30 16:00:00"
  and siteName IS NOT NULL
order by
  time desc
```

In the query result, click 📊 to view graphical results.

- Set **Graph Type** to the bar chart.
- Set **X-AXIS** to **time**.
- Set **Y-AXIS** to **latitude**.
- Set **Results** to **10**.

The command output is as follows:

Figure 3-9 Driving information



Query: select driverID, carNumber, latitude, longitude, siteName, time from detail_records where driverID = "panxian1000005" and time > "2017-01-30 16:00:00" and siteName IS NOT NULL order by time desc

Job ID: 9cf4e5cc-5160-44b8-ae9b-8b2e1136144a

The query takes 23.93s, and 186.39 MB scanned.A maximum of 1,000 records can be displayed.

3.2 Converting Data Format from CSV to Parquet

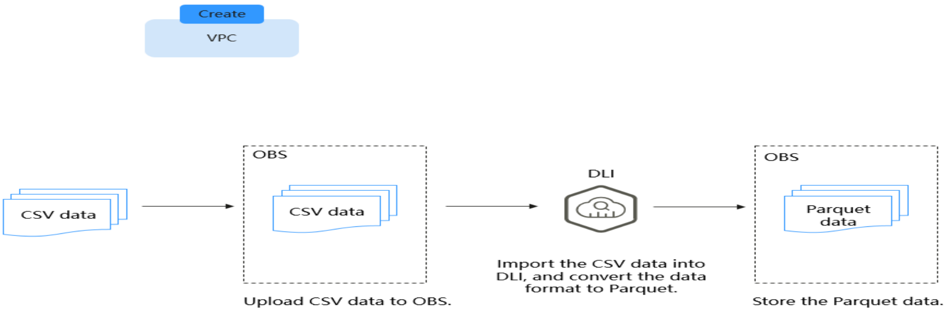
Application Scenarios

Parquet is a columnar storage substrate created for simpler data analysis. This format can speed up queries by allowing only the required columns to be read and calculated. In addition, Parquet is built to support efficient compression schemes, which maximizes the storage efficiency on disks. Using DLI, you can easily convert data format form CSV to Parquet.

Solution Overview

Upload CSV data to an OBS bucket, convert CSV data into Parquet data with DLI, and store the converted Parquet data to OBS.

Figure 3-10 Solution overview



Process

To use DLI to convert CSV data into Parquet data, perform the following steps:

Step 1: Creating and Uploading Data. Upload data to your OBS bucket.

Step 2: Using DLI to Convert CSV Data into Parquet Data. Import CSV data to DLI and convert it into Parquet data.

Solution Advantages

- **The query performance is improved.**

If you have text-based data files or tables in an HDFS and are using Spark SQL to query data, converting data format to Parquet can improve the query performance by about 30 times (or more in some cases), despite of the time consumed during the conversion.

- **Storage is saved.**

  Parquet is built to support efficient compression schemes, which maximizes the storage efficiency on disks. With Parquet, the storage cost can be reduced by about 75%.

Resource Planning and Costs

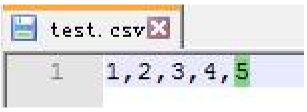
Table 3-4 Resource planning and costs

Resource	Description	Cost
OBS	You need to create an OBS bucket and upload data to OBS for data analysis using DLI.	You will be charged for using the following OBS resources: ● **Storage Fee** for storing static website files in OBS. ● **Request Fee** for accessing static website files stored in OBS. ● **Traffic Fee** for using a custom domain name to access OBS over the public network. The actual fee depends on the size of the stored file, the number of user access requests, and the traffic volume. Estimate the fee based on your service requirements.
DLI	Before creating a SQL job, you need to purchase a queue. When using queue resources, you are billed based on the CUH of the queue.	For example, if you purchase a pay-per-use queue, you will be billed based on the number of CUHs used by the queue. Usage is billed by the hour. For example, 58 minutes of usage will be rounded to the hour. CUH pay-per-use billing = Unit price x Number of CUs x Number of hours.

Step 1: Creating and Uploading Data

1. Create a CSV file. See **test.csv** in **Figure 3-11**.

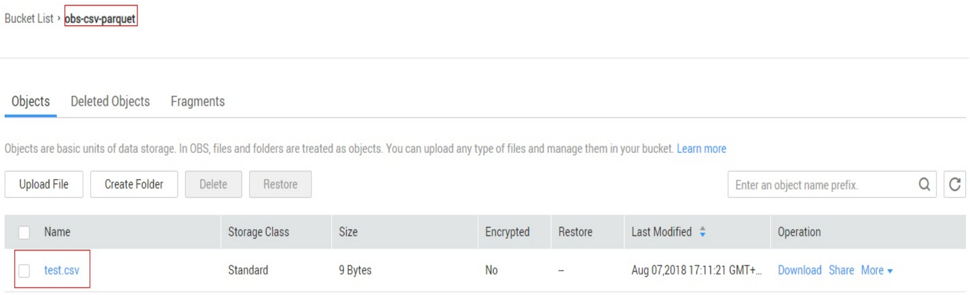
   Figure 3-11 Creating a **test.csv** file

   ```
   1,2,3,4,5
   ```

2. In the OBS management console, create a bucket, name it **obs-csv-parquet**, and upload the **test.csv** file to the bucket.

Figure 3-12 Uploading CSV data to OBS



Objects are basic units of data storage. In OBS, files and folders are treated as objects. You can upload any type of files and manage them in your bucket. Learn more

Name	Storage Class	Size	Encrypted	Restore	Last Modified	Operation
test.csv	Standard	9 Bytes	No	--	Aug 07,2018 17:11:21 GMT+...	Download Share More

3. Create a bucket and name it **obs-parquet-data** to store the converted parquet data.

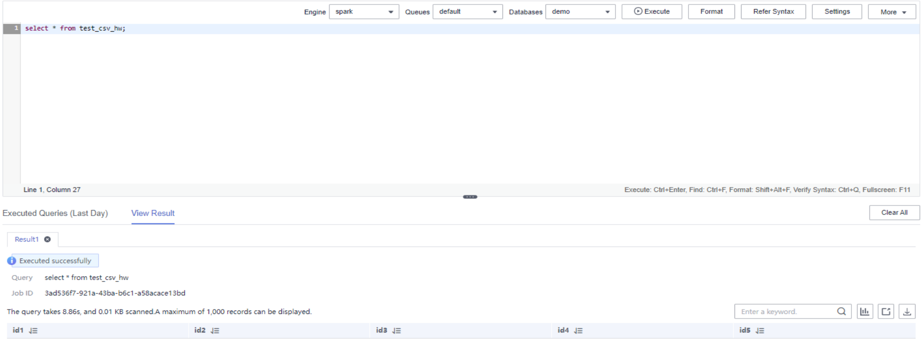
Step 2: Using DLI to Convert CSV Data into Parquet Data

1. Go to the DLI console, click **SQL Editor** in the navigation pane.
2. In the left pane of the SQL editor, click the **Databases** tab. Click ⊕, create a database, and name it **demo**.
3. In the SQL editing window, set **Engine** to **spark**, **Queue** to **default**, and **Database** to **demo**. Execute the following statement to create table **test_csv_hw** to import the data in the **test.csv** file from OBS.

```
create table test_csv_hw(id1 int, id2 int, id3 int, id4 int, id5 int)
 using csv
 options(
 path 'obs://obs-csv-parquet/test.csv'
 )
```

4. In the SQL editing window, query data in the **test_csv_hw** table.

Figure 3-13 Querying data



```
select * from test_csv_hw;
```

Executed successfully

Query select * from test_csv_hw

Job ID 3ad536f7-921a-43ba-b6c1-a58acace13bd

The query takes 8.86s, and 0.01 KB scanned.A maximum of 1,000 records can be displayed.

id1	id2	id3	id4	id5
1	2	3	4	5

5. In the SQL job editing window, create a table to store the OBS data in Parquet format and name the table **test_parquet_hw**.

```
create table `test_parquet_hw` (`id1` INT, `id2` INT, `id3` INT, `id4` INT, `id5` INT)
using parquet
options (
path 'obs://obs-parquet-data/'
)
```

NOTE

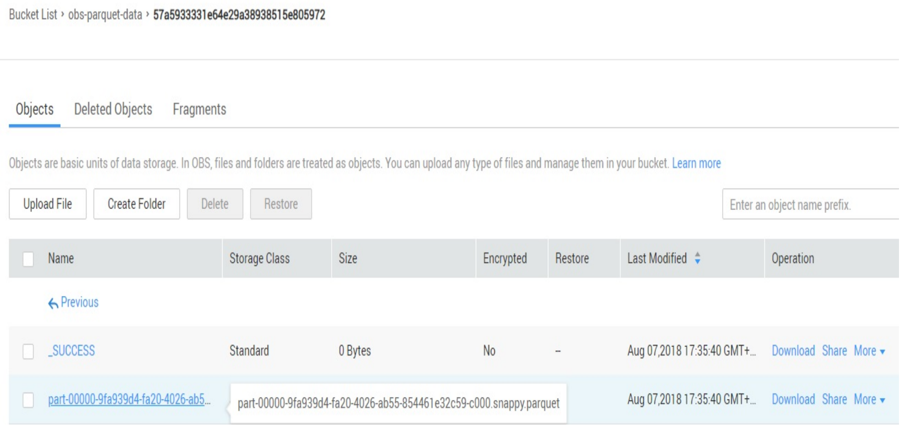
You do not need to specify a file because no Parquet file exists in this OBS bucket before the data is converted.

6. In the SQL editing window, execute the following statement to convert the CSV data to Parquet format and store the data in the specified OBS folder:

```
insert into test_parquet_hw select * from test_csv_hw
```

7. Check the result. OBS automatically created a file for saving the result.

Figure 3-14 Parquet data saved in a file in OBS



Bucket List › obs-parquet-data › **57a5933331e64e29a38938515e805972**

Objects are basic units of data storage. In OBS, files and folders are treated as objects. You can upload any type of files and manage them in your bucket. Learn more

□ Name	Storage Class	Size	Encrypted	Restore	Last Modified	Operation
□ _SUCCESS	Standard	0 Bytes	No	--	Aug 07,2018 17:35:40 GMT+...	Download Share More ▾
□ part-00000-9fa939d4-fa20-4026-ab5...					Aug 07,2018 17:35:40 GMT+...	Download Share More ▾

part-00000-9fa939d4-fa20-4026-ab55-854461e32c59-c000.snappy.parquet

3.3 Analyzing E-commerce BI Reports

Application Scenarios

As a self-operated e-commerce company in China, the *X* mall has developed hundreds of millions of loyal users and accumulated massive amounts of authentic data while maintaining high-speed development. How to use the BI tool to find business opportunities from historical data is a key issue in the precision marketing of big data applications. It is also the core technology required for intelligent upgrade of all e-commerce platforms.

This case uses HUAWEI CLOUD DLI, GaussDB(DWS), and Yonghong BI to analyze data features of users and offerings based on the real user, product, and comment data (anonymized) of the mall, providing high-quality information for marketing decision-making, advertising recommendation, credit rating, brand monitoring, and user behavior prediction.

Process

To use DLI to analyze e-commerce data, perform the following steps:

Step 1: Uploading Data. Upload the data to OBS for data analysis using DLI.

Step 2: Analyzing Data. Use DLI to query the data for analysis.

Data Types

To protect user privacy and data security, all sampled data is anonymized.

- User data

Table 3-5 User data

Field	Data Type	Description	Value
user_id	int	User ID	Anonymized
age	int	Age group	**-1** indicates that the user age is unknown.
gender	int	Gender	● 0: Male ● 1: Female ● 2: Confidential
rank	Int	User level	Sequenced list of user level. The higher the user level, the larger the number.
register_time	string	User registration date	Unit: day

- Product data

Table 3-6 Product data

Field	Data Type	Description	Value
product_id	int	Product No.	Anonymized
a1	int	Attribute 1	Enumerated value. The value **-1** indicates unknown.
a2	int	Attribute 2	Enumerated value. The value **-1** indicates unknown.
a3	int	Attribute 3	Enumerated value. The value **-1** indicates unknown.
category	int	Category ID	Anonymized
brand	int	Brand ID	Anonymized

- Comment data

Table 3-7 Comment data

Field	Data Type	Description	Value
deadline	string	End time	Unit: day

Field	Data Type	Description	Value
product_id	int	Product No.	Anonymized
comment_num	int	Segments of accumulated comment count	● **0**: No comment ● **1**: One comment ● **2**: 2 to 10 comments ● **3**: 11-50 comments ● **4**: More than 50 comments
has_bad_comment	int	Whether there is negative feedback.	0: No; 1: Yes.
bad_comment_rate	float	Dissatisfaction rate	Proportion of the negative feedback.

- Action data

Table 3-8 Action data

Field	Data Type	Description	Value
user_id	int	User ID	Anonymized
product_id	int	Product No.	Anonymized
time	string	Time of action	-
model_id	string	Module ID	Anonymized
type	string	● Browse (refers to the offering details page) ● Add to cart ● Remove from cart ● Place an order ● Follow ● Click	-

Step 1: Uploading Data

Upload the data to OBS for data analysis using DLI.

1. Download OBS Browser+. For details about the download address, see **Object Storage Service Tool Guide**.
2. Install OBS Browser+. For details about the installation procedure, see **Object Storage Service Tool Guide**.

3. Log in to OBS Browser+. OBS Browser+ supports two login modes: AK login (using access keys) or authorization code login. For details about the login procedure, see **Object Storage Service Tool Guide**.
4. Upload data using the OBS Browser+.

   On the OBS Browser+ page, click **Create Bucket**. Select a region and enter a bucket name (for example, **DLI-demo**). After the bucket is created, return to the bucket list and click **DLI-demo**. OBS Browser+ supports upload by dragging. You can drag one or more files or folders from a local path to the object list of a bucket or a parallel file system on OBS Browser+. You can even drag a file or folder directly to a specified folder on OBS Browser+.

   Obtain the test data by downloading the **Best_Practice_04.zip** file, decompressing it, and uploading the **Data** folder to the root directory of the OBS bucket. The test data directory is as follows:

   – **data/JData_User**: Data in the **user** table
   – **data/JData_Product**:Data in the **product** table
   – **data/JData_Product/JData_Comment**: Data in the **comment** table
   – **data/JData_Action**: Data the **action** table

Step 2: Analyzing Data

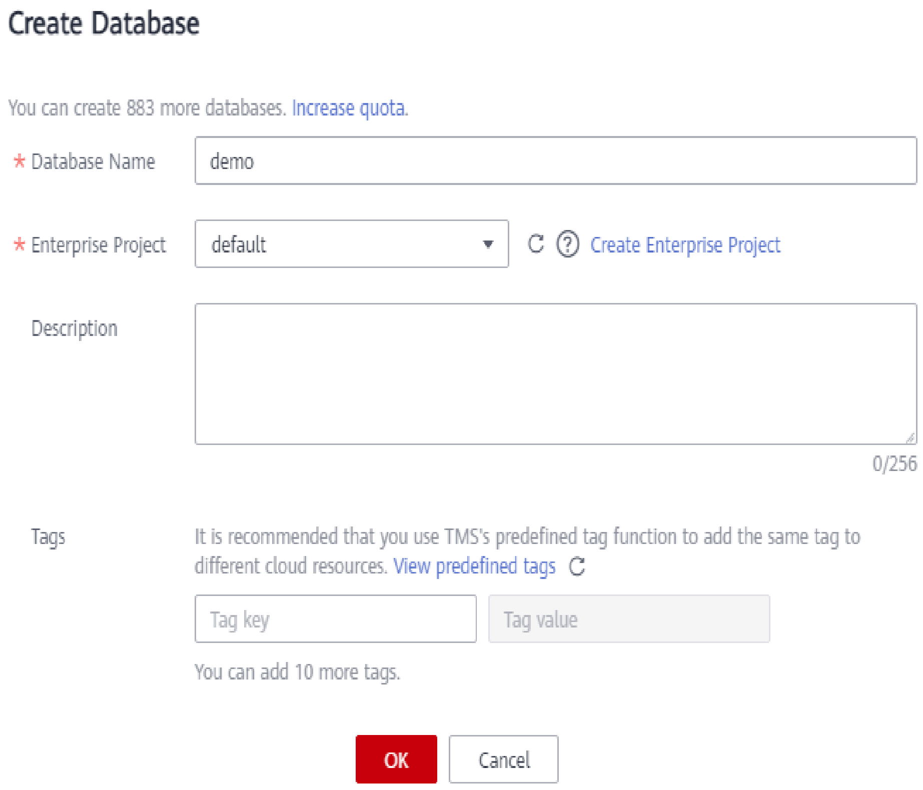
1. Creating a Database and a Table

   a. On the top menu bar of the portal page, choose **Products** > **Analytics** > **Data Lake Insight (DLI)**.

   b. Create a demo database. On the DLI console, choose **Job Management** >**SQL Jobs**. Click the created job on the displayed page to go to the **SQL Editor** page.

   c. In the left pane of the SQL Editor, select the **Databases** tab and click ⊕ to create the **demo** database. For details, see **Figure 3-15**.

Figure 3-15 Creating a database



Create Database

You can create 883 more databases. Increase quota.

* Database Name: demo

* Enterprise Project: default

Description:

0/256

Tags: It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

You can add 10 more tags.

NOTE

The **default** database is a built-in database. You cannot create a database named **default**.

d. Choose the **demo** database, and enter the following SQL statement in the editing box:

```
create table user(
  user_id int,
  age int,
  gender int,
  rank int,
  register_time string
) USING csv OPTIONS (path "obs://DLI-demo/data/JData_User")
```

NOTE

The file path in the preceding SQL statement is the actual OBS path for storing data.

e. Click **Execute** to create the user information table user.

f. Create the **product**, **comment**, and **action** tables in the same way.

- Product data

```
create table product(
  product_id int,
  a1 int,
  a2 int,
  a3 int,
  category int,
  brand int
) USING csv OPTIONS (path "obs://DLI-demo/data/JData_Product")
```

- Comment table

```
create table comment(
  deadline string,
  product_id int,
  comment_num int,
  has_bad_comment int,
  bad_comment_rate float
) USING csv OPTIONS (path "obs://DLI-demo/data/JData_Comment")
```

- Action table

```
create table action(
  user_id int,
  product_id int,
  time string,
  model_id string,
  type string
) USING csv OPTIONS (path "obs://DLI-demo/data/JData_Action");
```

2. Querying Data

You can save common query statements as templates on the **Template Management** page for later use. For details, see **SQL Template Management** in *Data Lake Insight User Guide*.

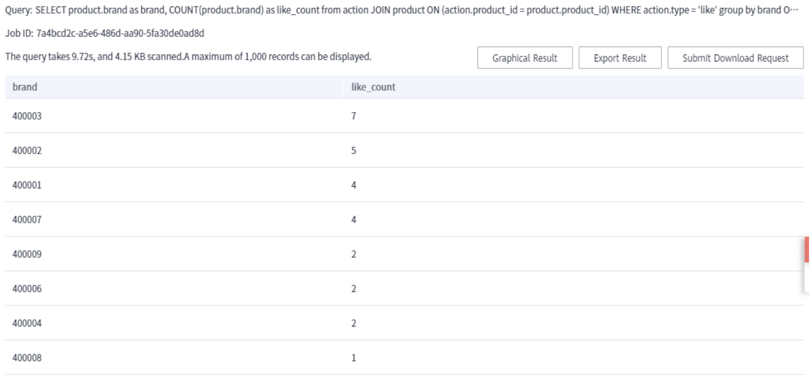
– Top 10 products with the most likes

i. Run the following SQL statement to analyze the top 10 products with the most likes.

```
SELECT
  product.brand as brand,
  COUNT(product.brand) as like_count
from
  action
  JOIN product ON (action.product_id = product.product_id)
WHERE
  action.type = 'like'
group by
  brand
ORDER BY like_count desc
limit
  10
```

ii. Click **Execute**. The execution results are displayed, as shown in **Figure 3-16**.

Figure 3-16 Querying results



Query: SELECT product.brand as brand, COUNT(product.brand) as like_count from action JOIN product ON (action.product_id = product.product_id) WHERE action.type = 'like' group by brand O···

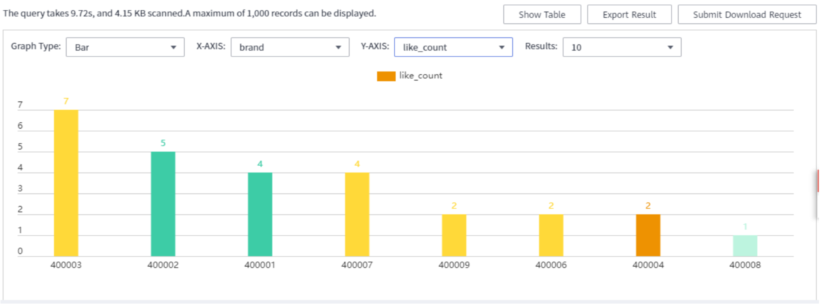
Job ID: 7a4bcd2c-a5e6-486d-aa90-5fa30de0ad8d

The query takes 9.72s, and 4.15 KB scanned.A maximum of 1,000 records can be displayed.

brand	like_count
400003	7
400002	5
400001	4
400007	4
400009	2
400006	2
400004	2
400008	1

iii. Click 📊 to view the result in a chart.

Figure 3-17 Graphical results



The query takes 9.72s, and 4.15 KB scanned.A maximum of 1,000 records can be displayed.

Graph Type: Bar

X-AXIS: brand

Y-AXIS: like_count

Results: 10

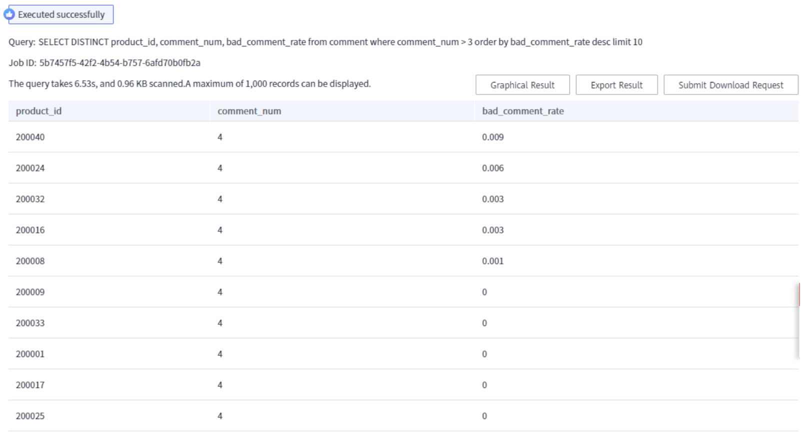
– Top 10 worst-rated products

i. Run the following SQL statement to analyze the top 10 worst-rated products:

```
SELECT
  DISTINCT product_id,
  comment_num,
  bad_comment_rate
from
  comment
where
  comment_num > 3
order by
  bad_comment_rate desc
limit
  10
```

ii. Click **Execute**. The execution results are displayed, as shown in Figure 3-18.

Figure 3-18 Querying results



Executed successfully

Query: SELECT DISTINCT product_id, comment_num, bad_comment_rate from comment where comment_num > 3 order by bad_comment_rate desc limit 10

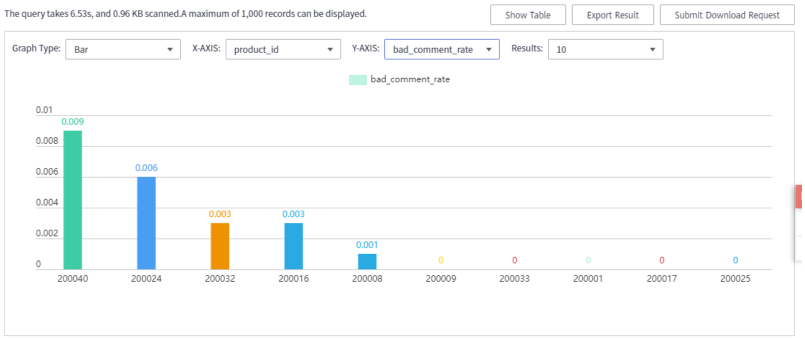
Job ID: 5b7457f5-42f2-4b54-b757-6afd70b0fb2a

The query takes 6.53s, and 0.96 KB scanned.A maximum of 1,000 records can be displayed.

product_id	comment_num	bad_comment_rate
200040	4	0.009
200024	4	0.006
200032	4	0.003
200016	4	0.003
200008	4	0.001
200009	4	0
200033	4	0
200001	4	0
200017	4	0
200025	4	0

iii. Click 📊 to view the result in a chart.

Figure 3-19 Graphical result



The query takes 6.53s, and 0.96 KB scanned.A maximum of 1,000 records can be displayed.

Graph Type: Bar

X-AXIS: product_id

Y-AXIS: bad_comment_rate

Results: 10

You can also analyze data for age distribution, gender ratio, offering evaluation, purchase number, and browsing statistics of users.

3.4 Analyzing DLI Billing Data

Application Scenarios

You can analyze DLI billing data (account information has been masked) on the big data analysis platform of DLI, find possible optimization, and figure out some measures to reduce costs for using DLI.

Analysis Process

Perform the following steps to analyze billing data and reduce costs:

Step 1: Obtaining Consumption Data. Obtain billing data of an account.

Step 2: Analyzing Billing Data and Reducing Costs. Analyze the consumption data, find the resources or users with high expenditure, and provide optimization measures to reduce cost.

Resources and Costs

Table 3-9 Resource planning and costs

Resource	Description	Cost
DLI	DLI is a big data analytics platform on Huawei Cloud. You are billed for using storage and compute resources. DLI supports three billing modes: yearly/ monthly, package, and pay-per-use.	You can run SQL jobs, Flink jobs, and Spark jobs on DLI. For SQL jobs, you are billed for both storage and compute resources. Compute resources can be billed based on a yearly/monthly basis or pay-per-use. ● If you choose the yearly/monthly billing mode, fees are deducted based on the subscription period. This billing mode is recommended for its preferential price and exclusive compute resources within the subscription period. ● In pay-per-use mode, fees are deducted by hour. You can choose either billing by CUH or by the amount of data scanned. Billing by CUH is recommended, for you can have exclusive resources and clear costing. In addition, you can purchase and use packages. – Billing for CUH used = Number of CUs x Usage duration x Unit price. The unit of the usage duration is hour. If the duration is less than one hour, it is rounded to one hour. – Billing for the amount of data scanned = Amount of data scanned during SQL statement execution x Unit price. If a computing task times out or fails, no fee is charged for the task. ● For Flink and Spark jobs, you will be billed for compute resources only. The billing rules are same to those of SQL jobs. For details, see **Price Calculator**. .

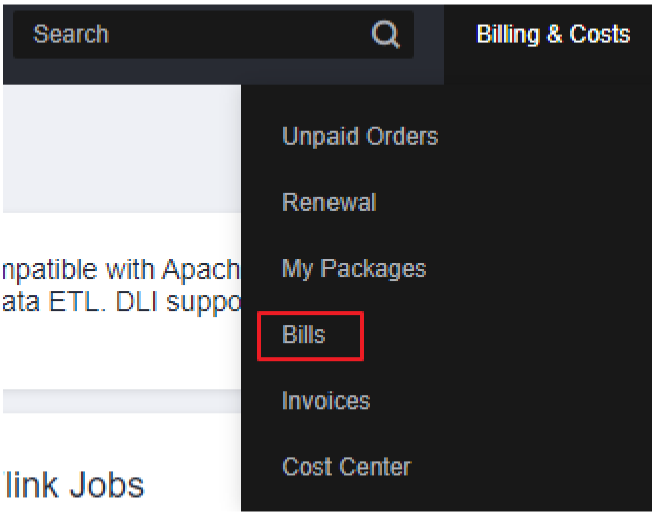
Step 1: Obtaining Consumption Data

1. Obtain billing details.

   a. Log in to the DLI console.

   b. Click **Billing & Costs** on the upper right corner of the page. Choose **Bills**.

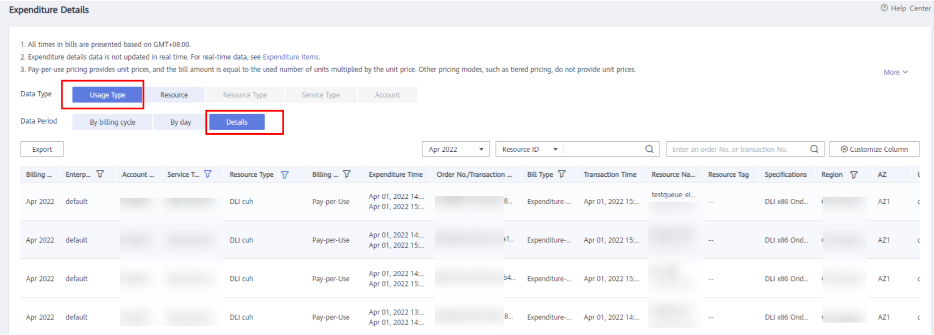
   Figure 3-20 Bills

   c. On the **Dashboard** page of the **Billing Center**, click **Expenditure Details**. On the displayed page, set **Data Type** to **Usage Type** and **Data Period** to **Details**. Set time to the billing cycle you want.

   In the title row of the displayed table, set **Service Type** to **Data Lake Insight (DLI)** and **Resource Type** to **DLI cuh**. Click **Export**. On the **Export** page, configure **Export Content** and **Period** as you need, and click **Export**. The **Export History** page is displayed.

   Figure 3-21 DLI Bills

   1. All times in bills are presented based on GMT+08:00.
   2. Expenditure details data is not updated in real time. For real-time data, see Expenditure Items.
   3. Pay-per-use pricing provides unit prices, and the bill amount is equal to the used number of units multiplied by the unit price. Other pricing modes, such as tiered pricing, do not provide unit prices.

Billing ...	Enterp...	Account ...	Service T...	Resource Type	Billing ...	Expenditure Time	Order No./Transaction ...	Bill Type	Transaction Time	Resource Na...	Resource Tag	Specifications	Region	AZ	U
Apr 2022	default	[redacted]	[redacted]	DLI cuh	Pay-per-Use	Apr 01, 2022 14:... Apr 01, 2022 15:...	[redacted]8...	Expenditure-...	Apr 01, 2022 15:...	testqueue_ei... [redacted]	--	DLI x86 Ond...	[redacted]	AZ1	c
Apr 2022	default	[redacted]	[redacted]	DLI cuh	Pay-per-Use	Apr 01, 2022 14:... Apr 01, 2022 15:...	[redacted]a1...	Expenditure-...	Apr 01, 2022 15:...	[redacted]	--	DLI x86 Ond...	[redacted]	AZ1	c
Apr 2022	default	[redacted]	[redacted]	DLI cuh	Pay-per-Use	Apr 01, 2022 14:... Apr 01, 2022 15:...	[redacted]b4...	Expenditure-...	Apr 01, 2022 15:...	[redacted]	--	DLI x86 Ond...	[redacted]	AZ1	c
Apr 2022	default	[redacted]	[redacted]	DLI cuh	Pay-per-Use	Apr 01, 2022 13:... Apr 01, 2022 14:...	[redacted]8...	Expenditure-...	Apr 01, 2022 14:...	[redacted]	--	DLI x86 Ond...	[redacted]	AZ1	c

   d. On the **Export History** page, wait until the file status changes to **Successful**. Click **Download**.

Step 2: Analyzing Billing Data and Reducing Costs

1. Analyze billing details.

   a. Upload the billing details downloaded in **Step 1: Obtaining Consumption Data** to the created OBS bucket.

b. Create a table on DLI.

   i. Log in to the DLI console. In the navigation pane, choose **SQL Editor**. Select **spark** for **Engine**, and select the queue and database. In this example, the default queue and database are used.

   ii. The downloaded file contains information such as time and usage. Create a table on DLI based on these table headers. For details, see the following example.

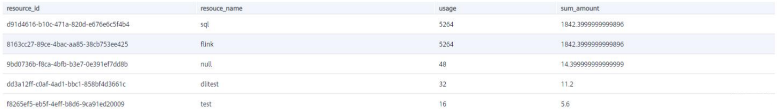
```
CREATE TABLE `spending` (
  account_period string,
  EnterpriseProject string,
  EnterpriseProjectID string,
  accountID string,
  product_type_code string,
  product_type string,
  product_code string,
  product_name string,
  product_id string,
  mode string,
  time1 string,
  use_start string,
  use_end string,
  orderid string,
  ordertime string,
  resource_type string,
  resource_id string,
  resouce_name string,
  tag string,
  skuid string,
  `c22name` STRING,
  `c23name` STRING,
  `c24name` STRING,
  `c25name` STRING,
  `c26name` STRING,
  `c27name` STRING,
  `c28name` STRING,
  `c29name` STRING,
  size STRING,
  `c31name` STRING,
  `c32name` STRING,
  `c33name` STRING,
  `c34name` STRING,
  `c35name` STRING,
  `amount` STRING,
  `c37name` STRING,
  `c38name` STRING,
  `c39name` STRING,
  `c40name` STRING,
  `c41name` STRING,
  `c42name` STRING,
  `c43name` STRING,
  `c44name` STRING,
  `c45name` STRING,
  `c46name` STRING,
  `c47name` STRING,
  `c48name` STRING,
  `c49name` STRING,
  `c50name` STRING,
  `c51name` STRING,
  `c52name` STRING,
  `c53name` STRING,
  `c54name` STRING
) USING csv options (
  path 'obs://xxx/Spendings(ByTransaction)_20200501_20200531.csv',
  header true)
```

c. Query **resource_id** and **resource_name** with the highest amount within the period.

The following statement shows the amount charged for using the SQL and Flink queues.

```
select resource_id, resouce_name, sum(size)
   as usage, sum(amount)
   as sum_amount
   from spending
   group by resource_id, resouce_name
   order by sum_amount desc
```

Figure 3-22 Query results



resource_id	resouce_name	usage	sum_amount
d91d4616-b10c-471a-820d-e676e6c5f4b4	sql	5264	1842.3999999999896
8163cc27-89ce-4bac-aa85-38cb753ee425	flink	5264	1842.3999999999896
9bd0736b-f8ca-4bfb-b3e7-0e391ef7dd8b	null	48	14.399999999999999
dd3a12ff-c0af-4ad1-bbc1-858bf4d3661c	dlitest	32	11.2
f8265ef5-eb5f-4eff-b8d6-9ca91ed20009	test	16	5.6

d. Run the following statements to analyze the usage periods of SQL and Flink resources:

```
select * from spending where resource_id = 'd91d4616-b10c-471a-820d-e676e6c5f4b4' order by
ordertime
```

The SQL queue was billed each hour from May 14 2020 17:00:00 GMT +08:00 to May 28, 2020 10:00:00 GMT+08:00.

Similarly, the Flink queue was continuously used from May 14, 2020 17:00:00 GMT+08:00 to May 28 2020 10:00:00 GMT+08:00.

2. Suggestion for reducing the cost

   You can change the SQL and Flink queues to yearly/monthly queues for lower costs. If you are sure about the number of CUHs required for a job, you can purchase a package to reduce the cost.

DLI helps you to analyze billing details of your enterprise to quickly find the unreasonable expenses and control costs. You can also use DLI to reduce your cost on Huawei Cloud.

3.5 Interconnecting Yonghong BI with DLI to Submit Spark Jobs

3.5.1 Preparing for Yonghong BI Interconnection

Scenario

Prepare for the interconnection between Yonghong BI system and DLI.

Procedure

Step 1 (Optional) In the upper left corner of the Huawei Cloud management console, click **Service List** and choose **Analytics** > **Data Lake Insight**. On the **Overview** page displayed, find the **Common Links** area on the right, and click **SDK Download**. On the **DLI SDK DOWNLOAD** page displayed, download a DLI JDBC driver, for example, **dli-jdbc-1.1.0-jar-with-dependencies-jdk1.7.jar**. For details, see .

Step 2 The AK/SK and token authentication modes can be used for JDBC authentication. The AK/SK authentication mode is recommended.

Step 3 Contact Yonghong customer service personnel to obtain the username and password of the Yonghong SaaS production environment.

Step 4 Log in to the Yonghong SaaS production environment and enter the username and password.

----End

3.5.2 Adding Yonghong BI Data Source

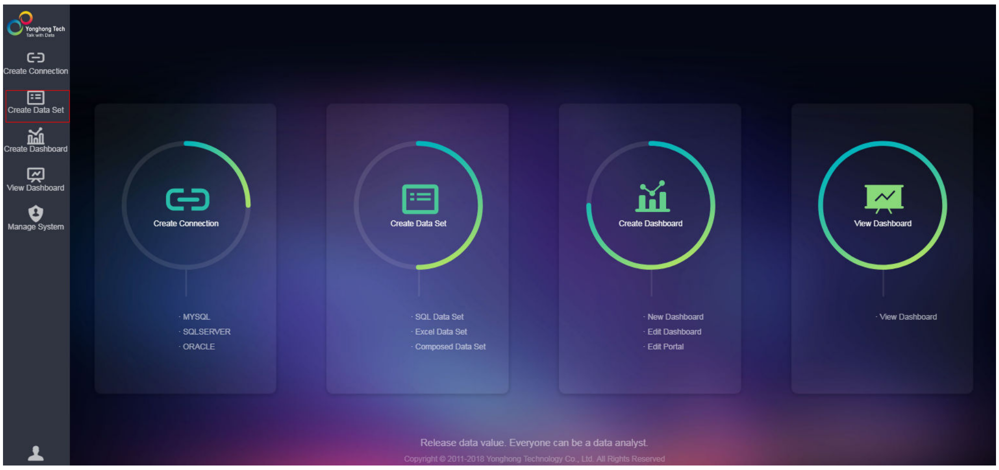
Scenario

Add the DLI data source to the Yonghong SaaS production environment.

Procedure

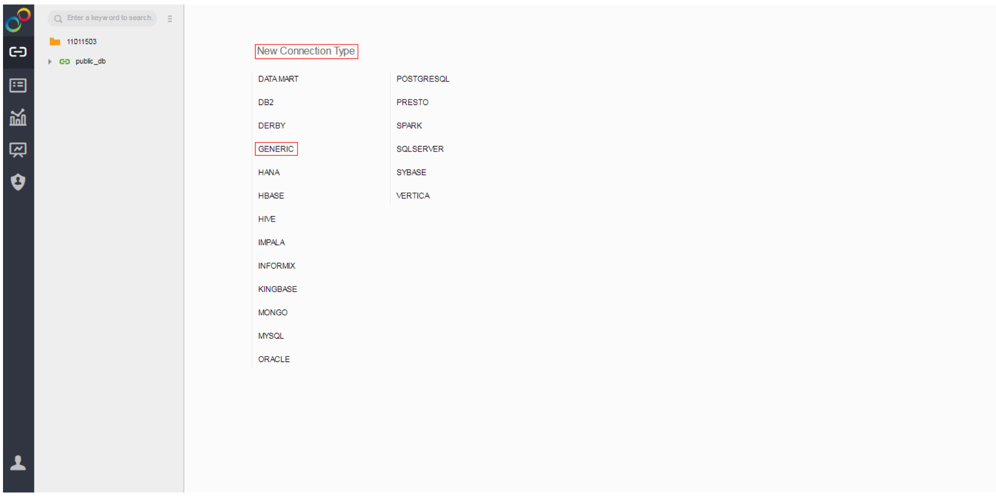
Step 1 On the homepage of the Yonghong SaaS production environment, click **Create Connection** from the left navigation tree. See **Figure 3-23**.

Figure 3-23 Adding a connection



Step 2 On the **New Connection Type** page, choose **GENERIC** for the type of the new connection. See **Figure 3-24**.

Figure 3-24 Choosing a new connection type



Step 3 Configure the new connection. See **Figure 3-25**.

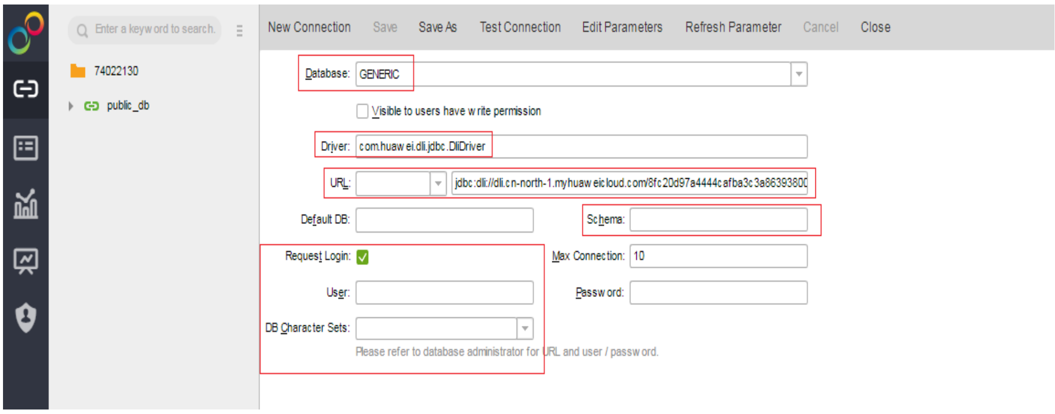
In **Driver**, enter **com.huawei.dli.jdbc.DliDriver**.

In **URL**, select **Self-defined Protocol**. Enter the URL of the DLI JDBC driver. For details about the URL format and the attributes, see **Table 3-10** and **Table 3-11**, respectively.

NOTE

- In **Schema**, you can optionally enter the name of the database to be accessed. If you enter the name, only tables in the database are displayed during data set creation. If you do not enter the name, tables in all databases are displayed during data set creation. For details about how to create a data set, see **Creating Yonghong BI Data Set**.
- Retain default values of other parameters. You do not need to select **Request Login**.

Figure 3-25 Configuring the new connection



Database: GENERIC

□ Visible to users have write permission

Driver: com.huawei.dli.jdbc.DliDriver

URL: jdbc:dli://dli.cn-north-1.myhuaweicloud.com/8fc20d97a4444cafba3c3a86393800

Default DB:

Schema:

Request Login: ☑

Max Connection: 10

User:

Password:

DB Character Sets:

Please refer to database administrator for URL and user / password.

Table 3-10 Database connection parameters

Parameter	Description
URL	The URL format is as follows: ***jdbc:dli://<endPoint>/<projectId>?<key1>=<val1>;<key2>=<val2>...*** **NOTE** ● **endpoint** indicates the domain name of DLI. For details, . ● **projectId** indicates the project ID, which can be obtained from the **My Credentials** page of the public cloud platform. ● The question mark (?) is followed by other configuration items. Each configuration item is listed in the "key=value" format. Semicolons (;) are used to separate configuration items. For details, see **Table 3-11**.

Table 3-11 Attribute-related configuration items

Attribute (key)	Mandatory	Default Value (value)	Description
queuename	Yes	-	Queue name of DLI.
databasename	No	-	Default database to be accessed. If this parameter is not specified in the URL, you need to use **db.table** (for example, **select * from dbother.tabletest**) to access tables in the database.
authentication mode	Yes	token	Authentication method, which can be **token** or **aksk**. Value **aksk** is recommended during the interconnection with Yonghong BI system.
accesskey	This parameter must be configured if **authentication mode** is set to **aksk**.	-	For details, see **Preparing for Yonghong BI Interconnection**.
secretkey	This parameter must be configured if **authentication mode** is set to **aksk**.	-	For details, see **Preparing for Yonghong BI Interconnection**.
regionname	This parameter must be configured if **authentication mode** is set to **aksk**.	-	For details, .
servicename	This parameter must be configured if **authentication mode** is set to **aksk**.	-	**servicename**=**dli**

Attribute (key)	Mandatory	Default Value (value)	Description
dli.sql.checkNoResultQuery	No	false	Whether to allow invoking the executeQuery API to execute statements (for example, DDL) that do not return results. ● Value **false** indicates that invoking of the executeQuery API is allowed. ● Value **true** indicates that invoking of the executeQuery API is not allowed. **NOTE** If **dli.sql.checkNoResultQuery** is set to **false**, non-query statements will be executed twice.

Step 4 On the tool bar of the displayed page, click **Test Connection**. After the test is complete, click **Save**. Enter the data source name, and save the data source.

NOTE

Currently, you are not allowed to save the data source to the root directory. Therefore, you can only save the data source to an existing folder.

----End

3.5.3 Creating Yonghong BI Data Set

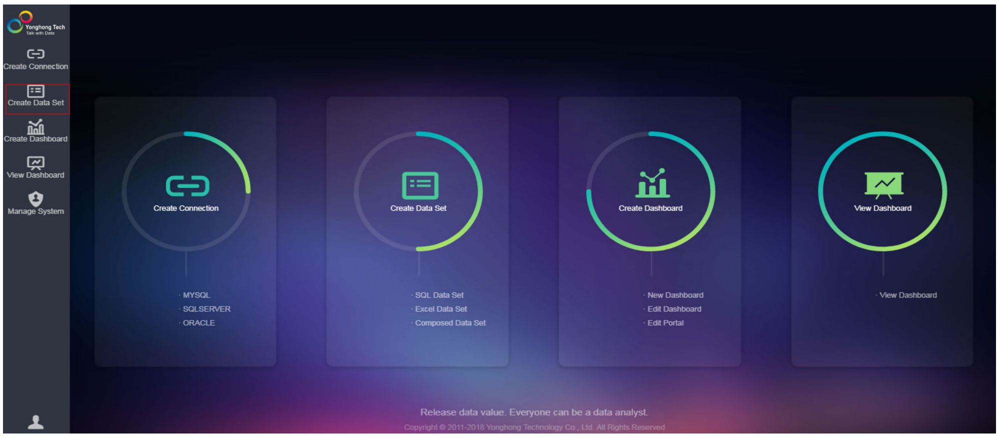
Scenario

Create a DLI data set in the Yonghong SaaS production environment.

Procedure

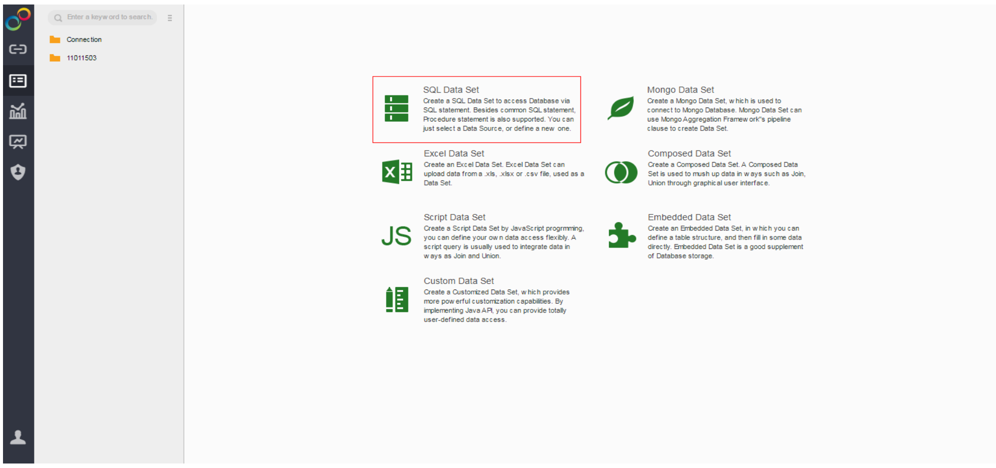
Step 1 On the home page of the Yonghong SaaS production environment, click **Create Data Set** in the left navigation tree. See **Figure 3-26**.

Figure 3-26 Creating a data set



Step 2 On the displayed page, click **SQL Data Set**. See **Figure 3-27**.

Figure 3-27 Creating a SQL data set



SQL Data Set
Create a SQL Data Set to access Database via SQL statement. Besides common SQL statement, Procedure statement is also supported. You can just select a Data Source, or define a new one.

Mongo Data Set
Create a Mongo Data Set, which is used to connect to Mongo Database. Mongo Data Set can use Mongo Aggregation Framework"s pipeline clause to create Data Set.

Excel Data Set
Create an Excel Data Set. Excel Data Set can upload data from a .xls, .xlsx or .csv file, used as a Data Set.

Composed Data Set
Create a Composed Data Set. A Composed Data Set is used to mush up data in ways such as Join, Union through graphical user interface.

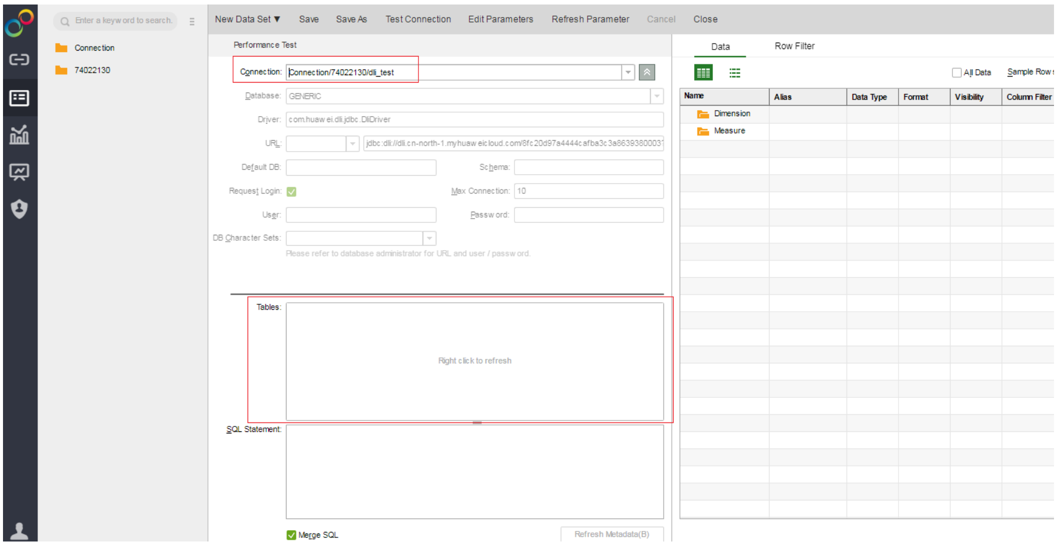
Script Data Set
Create a Script Data Set by JavaScript progrmming, you can define your own data access flexibly. A script query is usually used to integrate data in ways as Join and Union.

Embedded Data Set
Create an Embedded Data Set, in which you can define a table structure, and then fill in some data directly. Embedded Data Set is a good supplement of Database storage.

Custom Data Set
Create a Customized Data Set, which provides more powerful customization capabilities. By implementing Java API, you can provide totally user-defined data access.

Step 3 On the displayed page, select the added DLI data source from the **Connection** drop-down list box. See **Figure 3-28**.

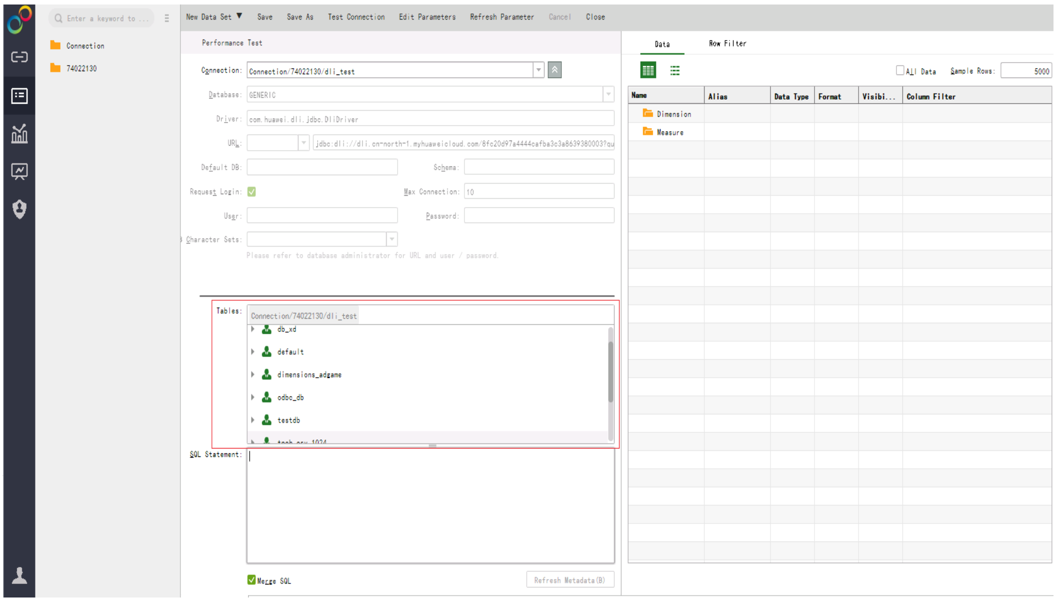
Figure 3-28 Selecting a data source



Connection: Connection/74022130/dli_test
Database: GENERIC
Driver: com.huawei.dli.jdbc.DliDriver
URL: jdbc:dli://dli.cn-north-1.myhuaweicloud.com/8fc20d97a4444cafba3c3a8639380003
Default DB:
Schema:
Request Login: ☑
Max Connection: 10
User:
Password:
DB Character Sets:
Please refer to database administrator for URL and user / password.

Step 4 In the **Table** area on the left pane, right-click and choose **Update** to update tables. All databases and their tables are listed in the area. **Figure 3-29** shows the page displayed when **Table Structure** is not configured during connection creation.

Figure 3-29 Updating tables



Connection: Connection/74022130/dli_test

Database: GENERIC

Driver: com.huawei.dli.jdbc.DliDriver

URL: jdbc:dli://dli.cn-north-1.myhuaweicloud.com/8fc20d97a4444cafba3c3a8639380003?qu

Default DB:

Schema:

Request Login: ☑

Max Connection: 10

User:

Password:

Character Sets:

Please refer to database administrator for URL and user / password.

Tables: Connection/74022130/dli_test
- db_xd
- default
- dimensions_adgame
- odbc_db
- testdb
- tech_env_1024

SQL Statement:

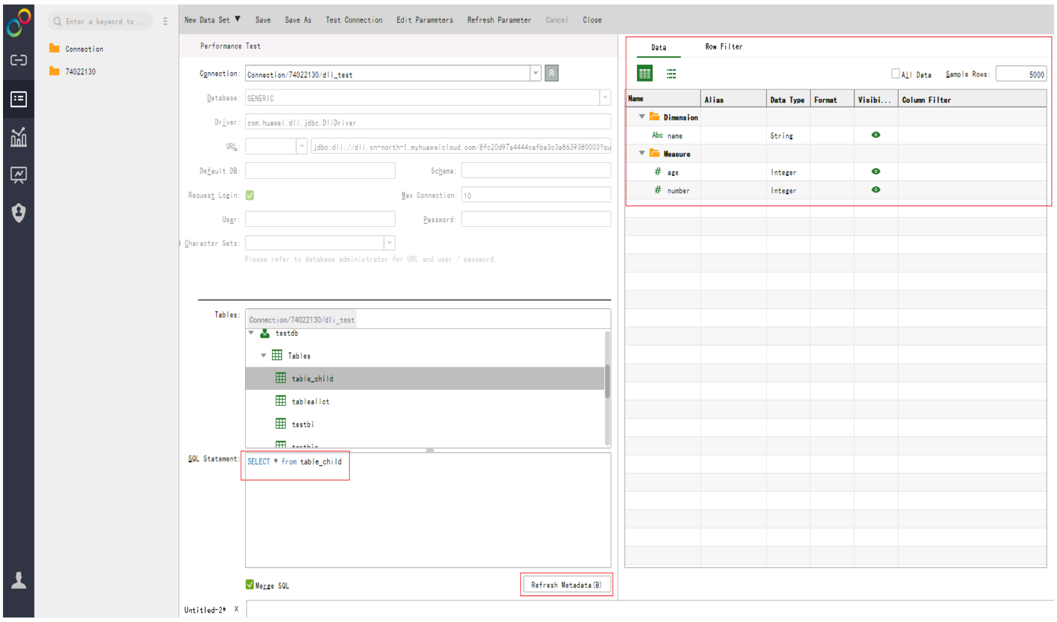
☑ Merge SQL

□ All Data Sample Rows: 5000

Name	Alias	Data Type	Format	Visibi...	Column Filter
Dimension					
Measure					

Step 5 In the **SQL Statement** area on the left pane, enter the **select * from table_name** command to query tables. On the **Preview Data** page on the right pane, click ▦. Metadata of the table, including fields and field types, is displayed. See **Figure 3-30**.

Figure 3-30 Querying the table



Connection: Connection/74022130/dli_test

Database: GENERIC

Driver: com.huawei.dli.jdbc.DliDriver

URL: jdbc:dli://dli.cn-north-1.myhuaweicloud.com/8fc20d97a4444cafba3c3a8639380003?qu

Default DB:

Schema:

Request Login: ☑

Max Connection: 10

User:

Password:

Character Sets:

Please refer to database administrator for URL and user / password.

Tables: Connection/74022130/dli_test
- testdb
  - Tables
    - table_child
    - tableallct
    - testbi
    - testbic

SQL Statement:

```
SELECT * from table_child
```

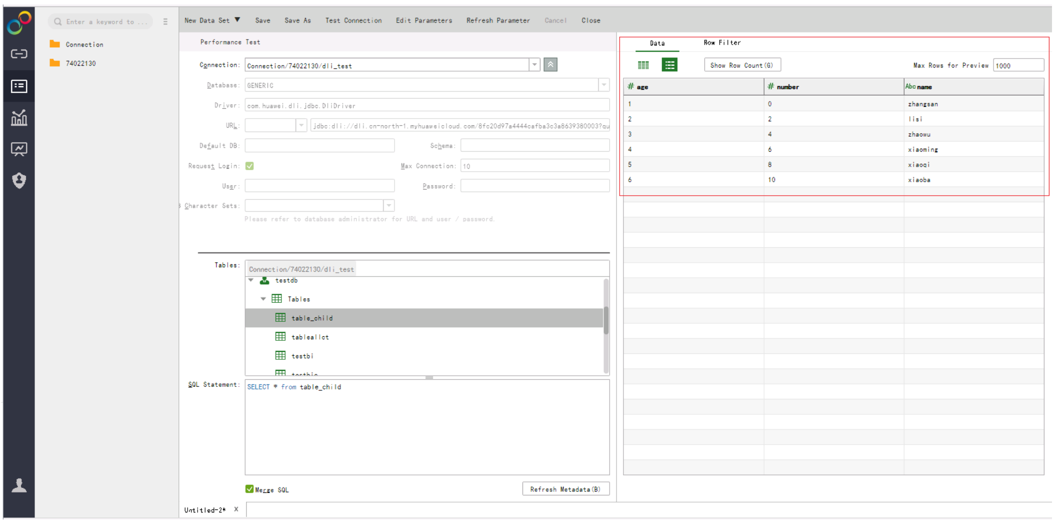
☑ Merge SQL

□ All Data Sample Rows: 5000

Name	Alias	Data Type	Format	Visibi...	Column Filter
Dimension					
Abc name		String		●	
Measure					
# age		Integer		●	
# number		Integer		●	

Step 6 Click ☰ on the right pane to query data details. See **Figure 3-31**.

Figure 3-31 Querying data of the table



Connection: Connection/74022130/dli_test

Database: GENERIC

Driver: com.huawei.dli.jdbc.DliDriver

URL: jdbc:dli://dli.cn-north-1.myhuaweicloud.com/8fc20d97a4444cafba3c3a8639380003?qu

Default DB:

Schema:

Request Login: ☑

Max Connection: 10

User:

Password:

Character Sets:

Please refer to database administrator for URL and user / password.

Tables: Connection/74022130/dli_test

- testdb
  - Tables
    - table_child
    - tableallct
    - testbi

SQL Statement:

```
SELECT * from table_child
```

☑ Merge SQL

Max Rows for Preview 1000

age	number	name
1	0	zhangsan
2	2	lisi
3	4	zhaowu
4	6	xiaoming
5	8	xiaoqi
6	10	xiaoba

Step 7 On the tool bar of the displayed page, click **Save**.

----End

3.5.4 Creating a Chart in Yonghong BI

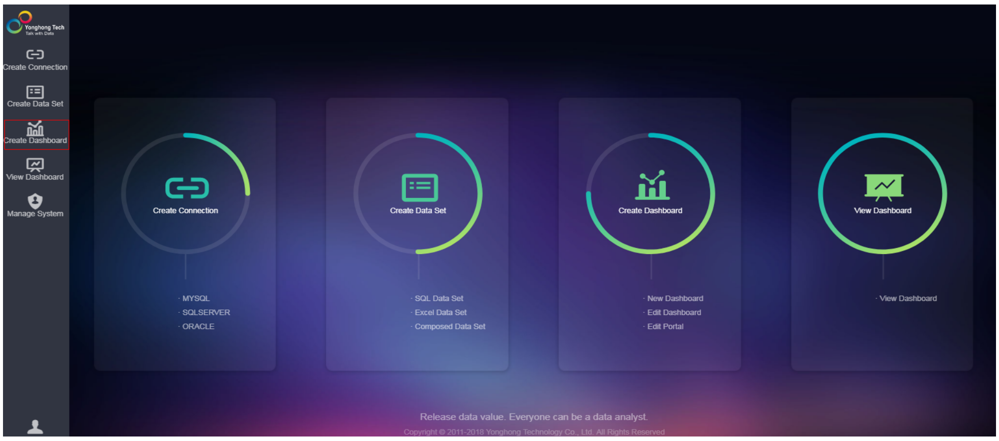
Scenario

Create a chart in the Yonghong SaaS production environment.

Procedure

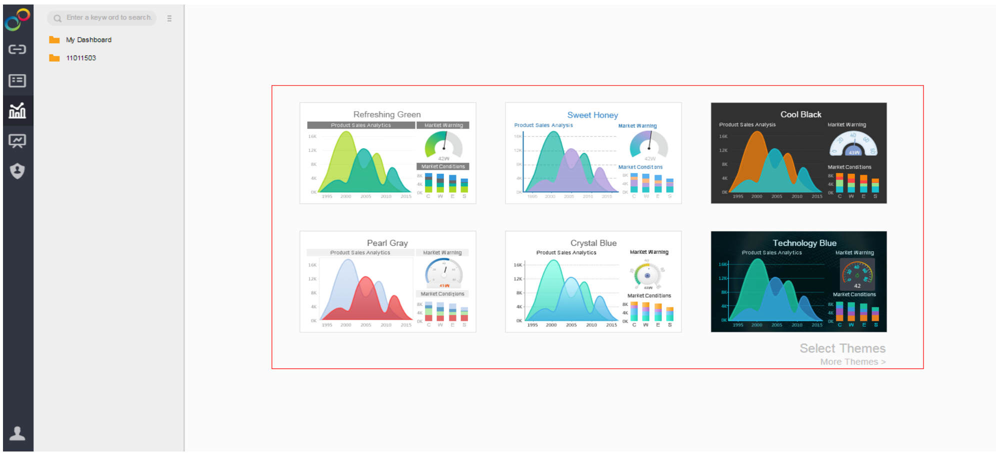
Step 1 On the home page of the Yonghong SaaS production environment, click **Create Dashboard** in the left navigation tree. See **Figure 3-32**.

Figure 3-32 Creating a dashboard



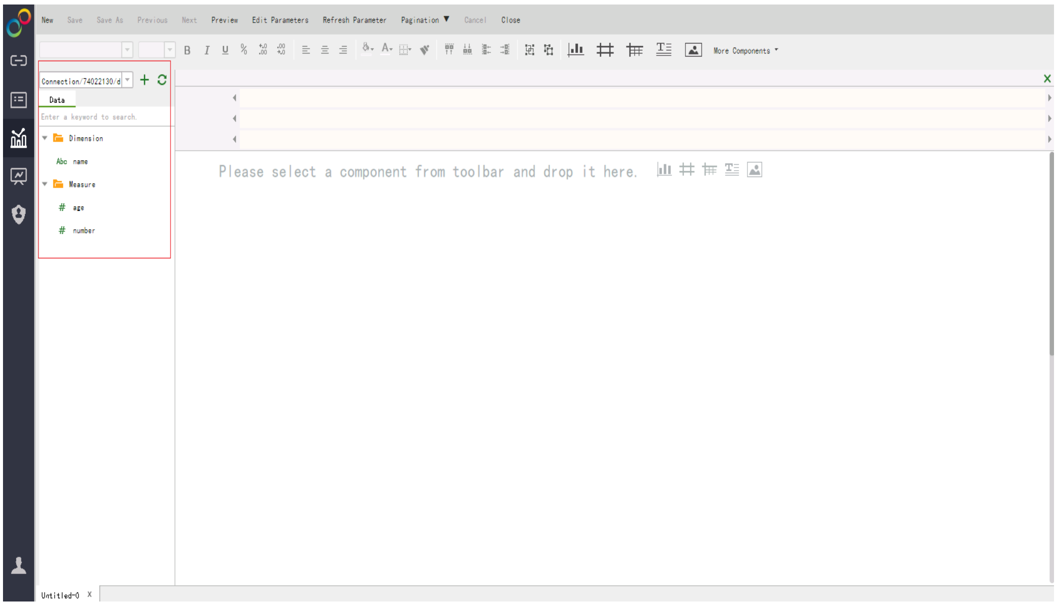
Step 2 Select a theme. See **Figure 3-33**.

Figure 3-33 Selecting a theme



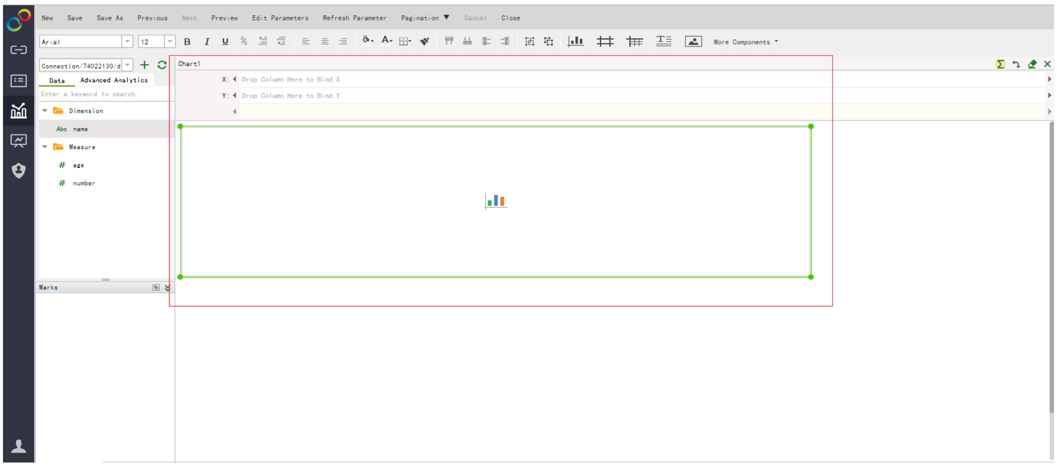
Step 3 In this example, the Refreshing Green theme is selected. On the left pane, select the created data set from the drop-down list box and choose a table as the data source (for example, **table_child**). Metadata (including fields and field types) of the table is displayed in the lower part of the **Data** column. See **Figure 3-34**.

Figure 3-34 Selecting a connection for the table



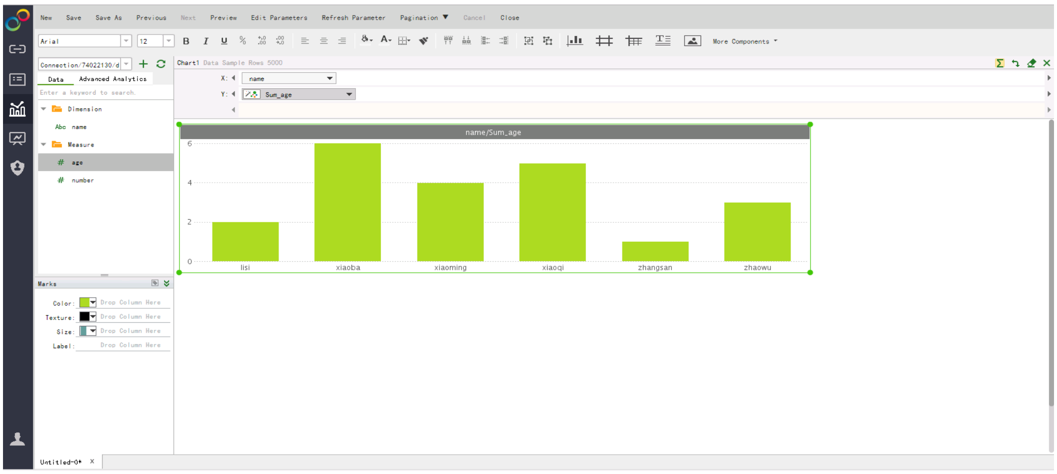
Step 4 On the report creation page, chart, table, matrix, and list filtering components are available. For example, if you want to create a chart, click 📊 and drag it to the editing area. See **Figure 3-35**.

Figure 3-35 Creating a chart



Step 5 In **X**, choose **name**. In **Y**, choose **age**. Drag them to the corresponding area, and the system automatically generates a bar chart. See **Figure 3-36**.

Figure 3-36 Generating a chart



Step 6 On the tool bar of the displayed page, click **Save**.

----End

4 Connections

4.1 Configuring the Connection Between a DLI Queue and a Data Source in a Private Network

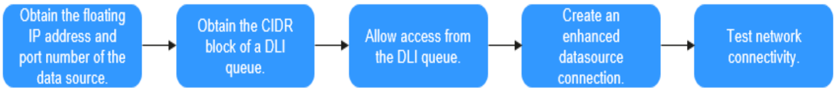
Background

If your DLI jobs need to connect to external data sources, for example, MRS, RDS, CSS, Kafka, or GaussDB(DWS), you need to enable the network between DLI and the external data sources. DLI enhanced datasource connection uses VPC peering to directly connect the VPC networks of the destination data sources for point-to-point data exchanges.

This section provides a guide to help you connect to data sources. You can also refer to this section to rectify connection faults.

Development Process

Figure 4-1 Configuration process of an enhanced datasource connection


Obtain the floating IP address and port number of the data source.
Obtain the CIDR block of a DLI queue.
Allow access from the DLI queue.
Create an enhanced datasource connection.
Test network connectivity.

Prerequisites

- You have created a queue. For details about how to create a queue, see **Creating a Queue**.

⚠ CAUTION

The queue billing mode must be **Pay-per-use**, and **Dedicated Resource Mode** must be selected after you select a queue type.

Enhanced datasource connections can be created only for pay-per-use resources in dedicated resource mode.

- A cluster of the external data source has been created. You can select a data source as needed.

Table 4-1 Reference for creating clusters of other data sources

Service Name	Reference Documents
RDS	**Getting Started with RDS for MySQL**
GaussDB(DWS)	**Creating a GaussDB(DWS) Cluster**
DMS Kafka	**Creating a Kafka Instance** CAUTION When you create the instance, do not enable **Kafka SASL_SSL**.
CSS	**Creating a CSS Cluster**
MRS	**Creating an MRS Cluster**

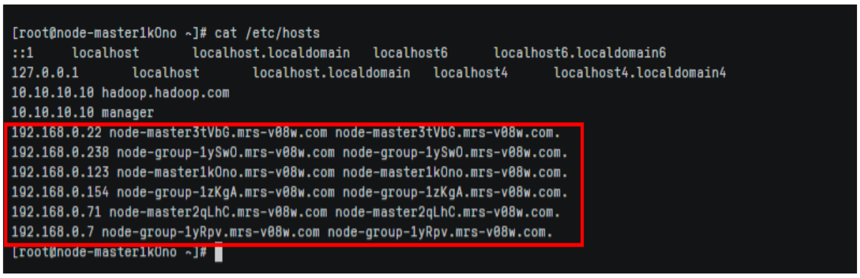
⚠ CAUTION

- The CIDR block of the DLI queue bound with a datasource connection cannot overlap with the CIDR block of other data sources.
- Datasource connections cannot be bound with the **default** queue.

Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source

Table 4-2 Data source information to be obtained

Data Source	Obtain Method
DMS Kafka	1. On the Kafka management console, click an instance name on the **DMS for Kafka** page. Basic information of the Kafka instance is displayed. 2. In the **Connection** pane, obtain the **Instance Address (Private Network)** value. In the **Network** pane, obtain the VPC and subnet of the instance. 3. In the **Network** pane, obtain the security group of the instance.
RDS	On the **Instances** page of the RDS console, click the target DB instance name. In the displayed page, locate the **Connection Information** pane and obtain the **Floating IP Address**, **VPC**, **Subnet**, **Database Port**, and **Security Group**.

<table>
<tr><th>Data Source</th><th>Obtain Method</th></tr>
<tr><td>CSS</td><td>1. On the CSS management console, choose <b>Clusters</b> &gt; <b>Elasticsearch</b>. On the displayed page, click the name of the created CSS cluster to view basic information.
2. On the <b>Cluster Information</b> page, obtain the <b>Private Network Address</b>, <b>VPC</b>, <b>Subnet</b>, and <b>Security Group</b>.</td></tr>
<tr><td>GaussDB(DWS)</td><td>1. On the GaussDB(DWS) management console, choose <b>Clusters</b>. On the displayed page, click the name of the created GaussDB(DWS) cluster to view basic information.
2. On the <b>Basic Information</b> tab, locate the <b>Database Attributes</b> pane and obtain the private IP address and port number of the DB instance. In the <b>Network</b> pane, obtain the VPC, subnet, and security group information.</td></tr>
<tr><td>MRS HBase</td><td>An MRS 3.x cluster is used as an example.
1. Log in to the MRS management console, click a cluster name on the <b>Clusters</b> &gt; <b>Active Clusters</b> page to view basic information.
2. On the dashboard, obtain VPC, subnet, and security group from the <b>Basic Information</b> pane.
3. The ZooKeeper instance and its port of the MRS cluster are required for creating a job that connects DLI to MRS HBase. You need to obtain the host information of the MRS cluster.
a. On MRS Manager, choose <b>Cluster</b> &gt; <i>Name of the desired cluster</i> &gt; <b>Services</b> &gt; <b>ZooKeeper</b>. Click the <b>Instance</b> tab and obtain the ZooKeeper host information such as the host name and service IP address.
b. On MRS Manager, choose <b>Cluster</b> and click the name of the desired cluster. Choose <b>Services</b> &gt; <b>ZooKeeper</b>. Click the <b>Configurations</b> tab and select <b>All Configurations</b>, search for the <b>clientPort</b> parameter, and obtain its value, that is, the ZooKeeper port number.
c. Log in to any MRS node as user <b>root</b> in SSH mode.
d. Run the following command to obtain MRS hosts information. Copy and save the information.
<b>cat /etc/hosts</b>
An example query result is as follows:

<pre>
[root@node-master1kOno ~]# cat /etc/hosts
::1   localhost   localhost.localdomain   localhost6   localhost6.localdomain6
127.0.0.1   localhost   localhost.localdomain   localhost4   localhost4.localdomain4
10.10.10.10 hadoop.hadoop.com
10.10.10.10 manager
192.168.0.22 node-master3tVbG.mrs-v08w.com node-master3tVbG.mrs-v08w.com.
192.168.0.238 node-group-1ySwO.mrs-v08w.com node-group-1ySwO.mrs-v08w.com.
192.168.0.123 node-master1kOno.mrs-v08w.com node-master1kOno.mrs-v08w.com.
192.168.0.154 node-group-1zKgA.mrs-v08w.com node-group-1zKgA.mrs-v08w.com.
192.168.0.71 node-master2qLhC.mrs-v08w.com node-master2qLhC.mrs-v08w.com.
192.168.0.7 node-group-1yRpv.mrs-v08w.com node-group-1yRpv.mrs-v08w.com.
[root@node-master1kOno ~]#
</pre></td></tr>
</table>

Step 2: Obtain the CIDR Block of the DLI Queue

On the DLI management console, choose **Resources** > **Queue Management** from the navigation pane. Locate the queue you have created, and click ⌄ next to the queue name to view the CIDR block of the queue.

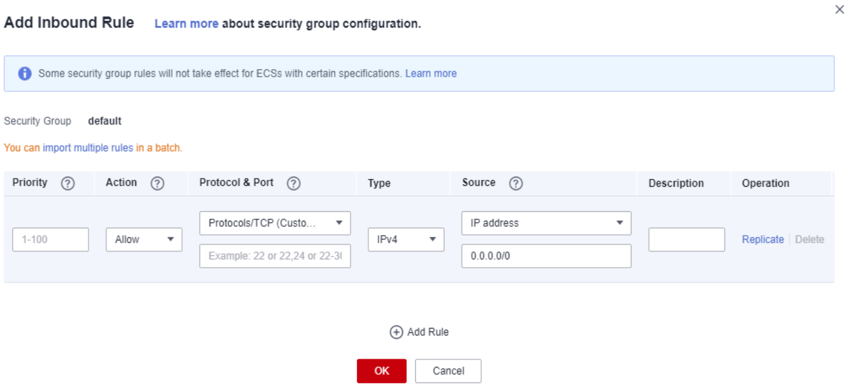
Step 3: Add a Rule to the Security Group of the External Data Source to Allow Access from the DLI Queue

1. Log in to the VPC console.
2. In the navigation pane on the left, choose **Access Control** > **Security Groups**.
3. Click the name of the security group to which the external data source belongs.

   To obtain the security group information, go to the management console of the data source service and follow the steps provided in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**.
4. In the **Inbound Rules** tab, add a rule to allow access from the queue network segment.

   For details about how to set the inbound rule parameters, see **Table 4-3**.

Figure 4-2 Adding an inbound rule



Add Inbound Rule Learn more about security group configuration.

Some security group rules will not take effect for ECSs with certain specifications. Learn more

Security Group **default**

You can import multiple rules in a batch.

Priority	Action	Protocol & Port	Type	Source	Description	Operation
1-100	Allow	Protocols/TCP (Custo... Example: 22 or 22,24 or 22-3(	IPv4	IP address 0.0.0.0/0		Replicate | Delete

Table 4-3 Inbound rule parameters

Parameter	Description	Example
Priority	The security group rule priority. The priority value ranges from 1 to 100. The default value is **1**, indicating the highest priority. A smaller value indicates a higher priority of a security group rule.	1
Action	Action of the security group rule.	Select **Allow**.
Protocol &Port	● Network protocol: The value can be **All**, **TCP**, **UDP**, **ICMP**, or **GRE**. ● Port: Port or port range over which the traffic can reach your instance. The port ranges from 1 to 65535.	In this example, select TCP. Leave the port blank or set it to the data source port obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**.
Type	Type of IP addresses.	IPv4
Source	Allow access from IP addresses or instances in another security group.	In this example, enter the queue network segment obtained in **Step 2: Obtain the CIDR Block of the DLI Queue**.
Description	Supplementary information about the security group rule. This parameter is optional.	–

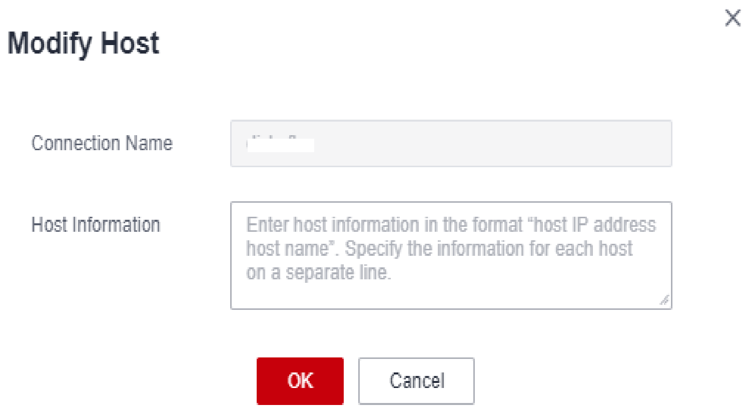
Step 4: Create an Enhanced Datasource Connection

1. Log in to the DLI management console. In the navigation pane on the left, choose **Datasource Connections**. On the displayed page, click **Create** in the **Enhanced** tab.
2. In the displayed dialog box, set the following parameters:
   - **Connection Name**: Name of the enhanced datasource connection
   - **Resource Pool**: Select the target DLI queue. (Queues that are not added to a resource pool are displayed in this list.)
   - VPC: VPC of the data source obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**

- Subnet: Subnet of the data source obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source**
- Set other parameters as you need.

3. Click **OK**. Click the name of the created datasource connection to view its status. You can perform subsequent steps only after the connection status changes to **Active**.
4. To connect to MRS HBase, you need to add MRS host information. The procedure is as follows:
   a. On the **Datasource Connections** page, click the **Enhanced** tab and locate the row that contains the created enhanced datasource connection. Click **More** > **Modify Host** in the **Operation** column.
   b. In displayed dialog box, enter the MRS HBase host information obtained in **Step 1: Obtain the Floating IP Address, Port Number, and Security Group of an External Data Source** to the **Host Information** box.

Figure 4-3 Modifying host information



Modify Host

Connection Name

Host Information: Enter host information in the format "host IP address host name". Specify the information for each host on a separate line.

   c. Click **OK**.

Step 5: Test Network Connectivity

1. Choose **Resources** > **Queue Management** from the left navigation pane, locate the target queue. In the **Operation** column, click **More** > **Test Address Connectivity**.
2. In the displayed dialog box, enter the obtained IP address and port number of the data source in the address box, and click **Test**. If the queue passes the test, it can access the data source.

NOTE

For MRS HBase, use **ZooKeeper IP address**:**ZooKeeper port** or **ZooKeeper host information**:**ZooKeeper port** for the test.

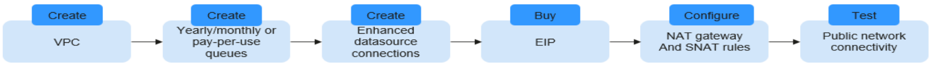
4.2 Configuring the Connection Between a DLI Queue and a Data Source in the Internet

Scenario

This section provides instructions to enable network connectivity for DLI queues to be accessed from the Internet. You can configure SNAT rules and add routes to the public network to enable communications between a queue and the Internet.

Procedure

Figure 4-4 Configuration process

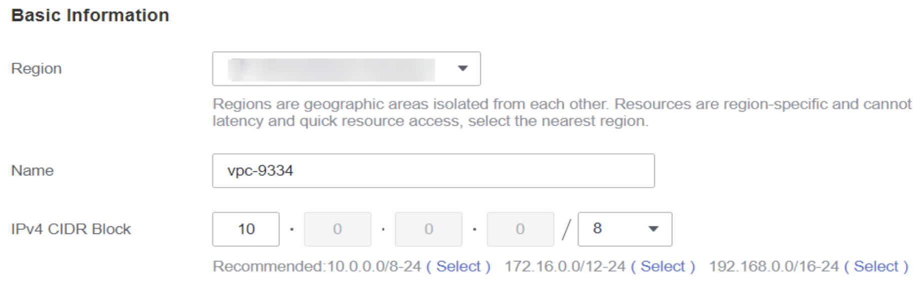

Create
VPC
Create
Yearly/monthly or pay-per-use queues
Create
Enhanced datasource connections
Buy
EIP
Configure
NAT gateway And SNAT rules
Test
Public network connectivity

Step 1: Create a VPC

Log in to the VPC console and create a VPC. The created VPC is used for NAT to access the public network.

For details about how to create a VPC, see **Creating a VPC**.

Figure 4-5 Creating a VPC



Basic Information

Region

Regions are geographic areas isolated from each other. Resources are region-specific and cannot latency and quick resource access, select the nearest region.

Name: vpc-9334

IPv4 CIDR Block: 10 · 0 · 0 · 0 / 8

Recommended:10.0.0.0/8-24 (Select) 172.16.0.0/12-24 (Select) 192.168.0.0/16-24 (Select)

Step 2: Create a Dedicated Queue

In this example, you will create a pay-per-use queue that uses dedicated resources.

⚠ CAUTION

The billing mode of the queue must be **Yearly/Monthly** or **Pay-per-use**. (If you select **Pay-per-use**, select **Dedicated Resource Mode** after you select a queue type.)

Enhanced datasource connections can be created only for yearly/monthly resources or pay-per-use resources in dedicated resource mode.

1. Log in to the DLI management console.
2. Click **Buy Queue** in the upper left corner on the homepage page. On the displayed page, specify specifications and other required parameters.

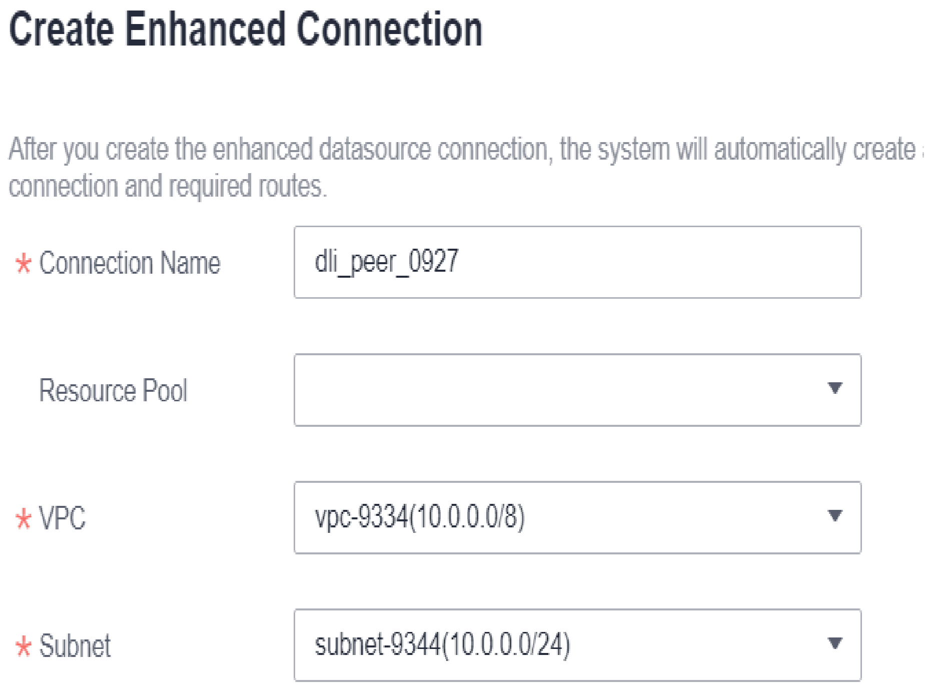
   For details about the parameters for purchasing a queue, see **Creating a Queue**.

Step 3: Create an Enhanced Datasource Connection Between the Queue and a VPC

1. In the navigation pane of the DLI management console, choose **Datasource Connections**.
2. In the **Enhanced** tab, click **Create**.

   Enter the connection name, select the created queue, VPC, and subnet, and enter the host information (optional).

Figure 4-6 Creating an enhanced datasource connection



Create Enhanced Connection

After you create the enhanced datasource connection, the system will automatically create connection and required routes.

* Connection Name	dli_peer_0927
Resource Pool	
* VPC	vpc-9334(10.0.0.0/8)
* Subnet	subnet-9344(10.0.0.0/24)

Step 4: Buy an EIP

1. Log in to the **EIPs** page of the network console, click **Buy EIP**.
2. In the displayed page, configure the parameters as required.

   For details about how to set the parameters, see **Buy EIP**.

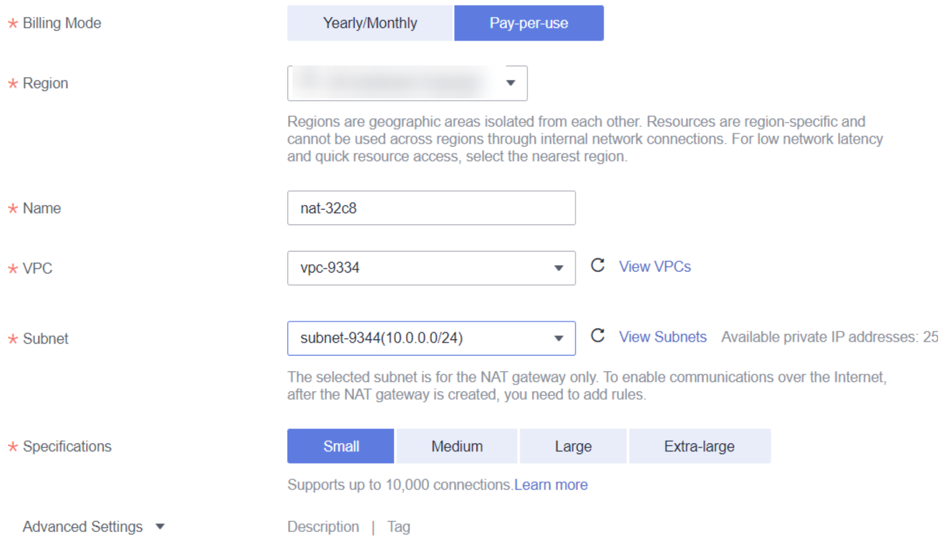
Step 5: Configure a NAT Gateway

Step 1 Create a NAT gateway.

1. Log in to the console and search for **NAT Gateway** in the Service List. The **Public NAT Gateways** page of the network console is displayed.

2. Click **Buy Public NAT Gateway** and configure the required parameters.

Figure 4-7 Buying a NAT gateway



* Billing Mode: Yearly/Monthly | **Pay-per-use**

* Region: [redacted]

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

* Name: nat-32c8

* VPC: vpc-9334 View VPCs

* Subnet: subnet-9344(10.0.0.0/24) View Subnets Available private IP addresses: 25

The selected subnet is for the NAT gateway only. To enable communications over the Internet, after the NAT gateway is created, you need to add rules.

* Specifications: **Small** | Medium | Large | Extra-large

Supports up to 10,000 connections.Learn more

Advanced Settings: Description | Tag

3. Click **Next**, confirm the configurations, and click **Submit**.

NOTE

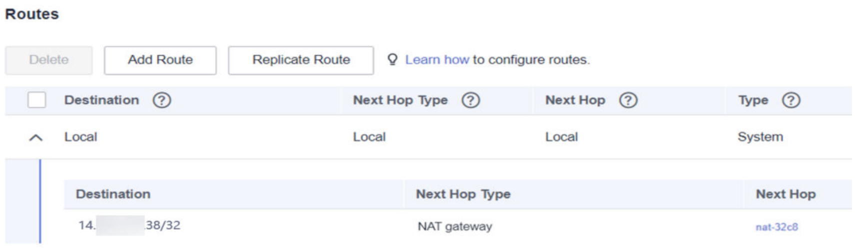
During the configuration, you need to set **VPC** to the one created in **Step 1: Create a VPC**.

Step 2 Add a route.

In the navigation pane on the left of the network console, choose Virtual **Private Cloud** > **Route Tables**. After a NAT gateway instance is created, a route to that gateway is automatically created. Click the route table name to view the automatically created route.

The destination address is the public IP address you want to access, and the next hop is the NAT gateway.

Figure 4-8 Viewing the route



Routes

Delete | Add Route | Replicate Route | Learn how to configure routes.

Destination	Next Hop Type	Next Hop	Type
Local	Local	Local	System

Destination	Next Hop Type	Next Hop
14.[redacted]38/32	NAT gateway	nat-32c8

Step 3 Add an SNAT rule.

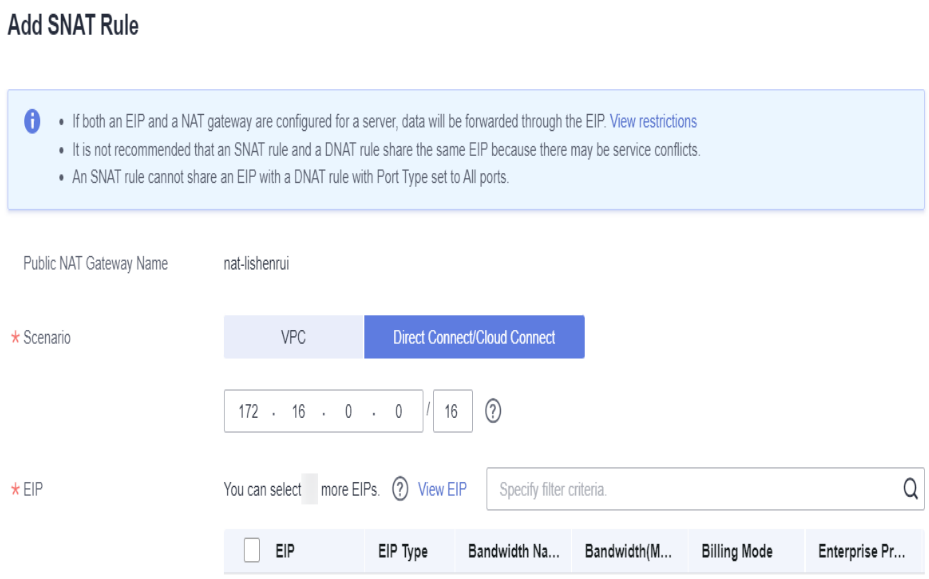
You need to add SNAT rules for the new NAT gateway to allow the hosts in the subnet to communicate with the Internet.

1. Click the name of the created NAT gateway on the **Public NAT Gateways** page of the network console.
2. On the **SNAT Rules** tab, click **Add SNAT Rule**.

For details, see **Adding an SNAT Rule**.

3. **Scenario**: Select **Direct Connect/Cloud Connect**.
4. **Subnet**: Select the subnet where the queue you want to connect locates.
5. **EIP**: Select the target EIP.

Figure 4-9 Adding an SNAT rule



Add SNAT Rule

- If both an EIP and a NAT gateway are configured for a server, data will be forwarded through the EIP. View restrictions
- It is not recommended that an SNAT rule and a DNAT rule share the same EIP because there may be service conflicts.
- An SNAT rule cannot share an EIP with a DNAT rule with Port Type set to All ports.

Public NAT Gateway Name nat-lishenrui

* Scenario VPC | Direct Connect/Cloud Connect

172 . 16 . 0 . 0 / 16

* EIP You can select more EIPs. View EIP Specify filter criteria.

□ EIP	EIP Type	Bandwidth Na...	Bandwidth(M...	Billing Mode	Enterprise Pr...

6. Click **OK**.

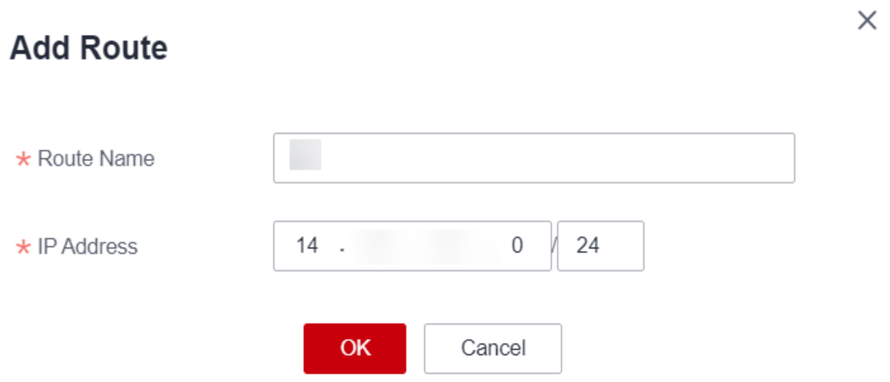
----End

Step 6: Adding a Custom Route

Add a custom route for the enhanced datasource connection you have created. Specify the route information of the IP address you want to access.

For details, see **Custom Route Information**.

Figure 4-10 Adding route information for test



Add Route

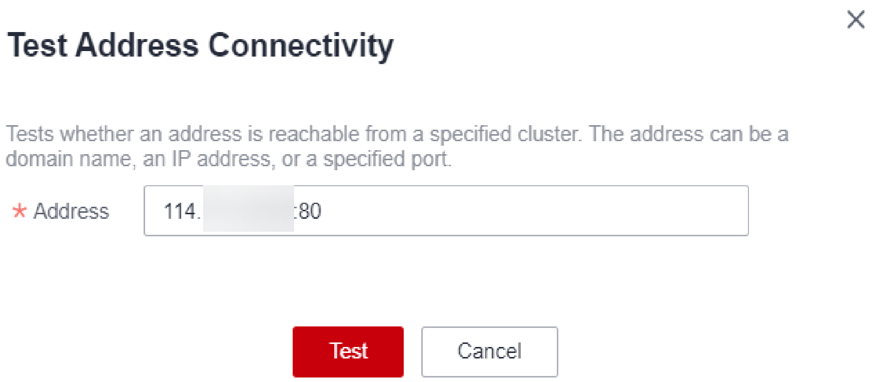
* Route Name

* IP Address 14 . 0 / 24

Step 7: Testing the Connectivity to the Public Network

Test the connectivity between the queue and the public network. Click **More** > **Test Address Connectivity** in the **Operation** column of the target queue and enter the public IP address you want to access.

Figure 4-11 Testing address connectivity



Test Address Connectivity

Tests whether an address is reachable from a specified cluster. The address can be a domain name, an IP address, or a specified port.

* Address 114.[redacted]:80

A Change History

Released On	What's New
2023-03-31	This issue is the third official release. Adjusted the document structure and moved the content related to DLI data development to *Data Lake Insight Development Guide*.
2022-10-31	This issue is the second official release. Optimized the following sections and added information about solution advantages, process guidance, and resource planning and costs. ● **Analyzing Driving Behavior Data** ● **Converting Data Format from CSV to Parquet**
2022-09-15	This issue is the first official release.